The Transactive Energy Challenge Model

1.1 TE Challenge

1.2 Use Cases for using TE ---

1.3 A Common Testbed for Transactive Energy Schemes

This section describes an abstract specification for a common model for simulating and emulating transactive energy schemes. It has a component model, a canonical simulation profile, examples of extended and collapsed components for various composition models, and a baseline scenario to use to explore the model.

It is envisioned that this model will be realized on one or more simulation platforms enabling participants in TE Challenge comparisons to evaluate their concepts on these testbeds.

Note: this version is a draft that will mature with the rest of this document. It is not complete at the present time and presented only to show the progress and path of thinking.

1.3.1.1 Model

Set of Federate Models for the TE Challenge. Each model represents a set of roles or interfaces that an actual device or computing platform might play in a transactive energy simulation.

1.3.1.1.1 TEComponents

These model components expose the key interfaces of the TE Common Model platform. Shown are the roles, their interfaces, and persistent data required.

Note that in an actual implementation, several of these "roles" may be combined into a single instance exposing multiple interfaces. See examples in the "Composite Classes" section.

1.3.1.1.1.1 Model diagram

This diagram illustrates the classes or roles of components of the TE Challenge Common Platform Model. The diagram shows three groupings of model components:

Core Components -- these represent the granularity of components that can be used in simulations. Additionally, they can be combined to represent less-granular components by aggregating their function and interfaces into composite components.

Specializations -- these represent specializations of the core components for specific roles in transactive engergy simulations.

Experiment Orchestration and Analysis -- these represent the orchestrator for the simulation experiment (Experiment Manager), and, the core Analytics component that evaluate the data produced during the simulation.

Note that these models provide for a minimum of interoperability and the ability to define an experiment that can exercise the models based on these designs. Any particular realization of these models might extend their capabilities and information exchanged. The base interoperable characteristics herein provide for the consistency of simulation execution and minimal availability of data for the analytics.

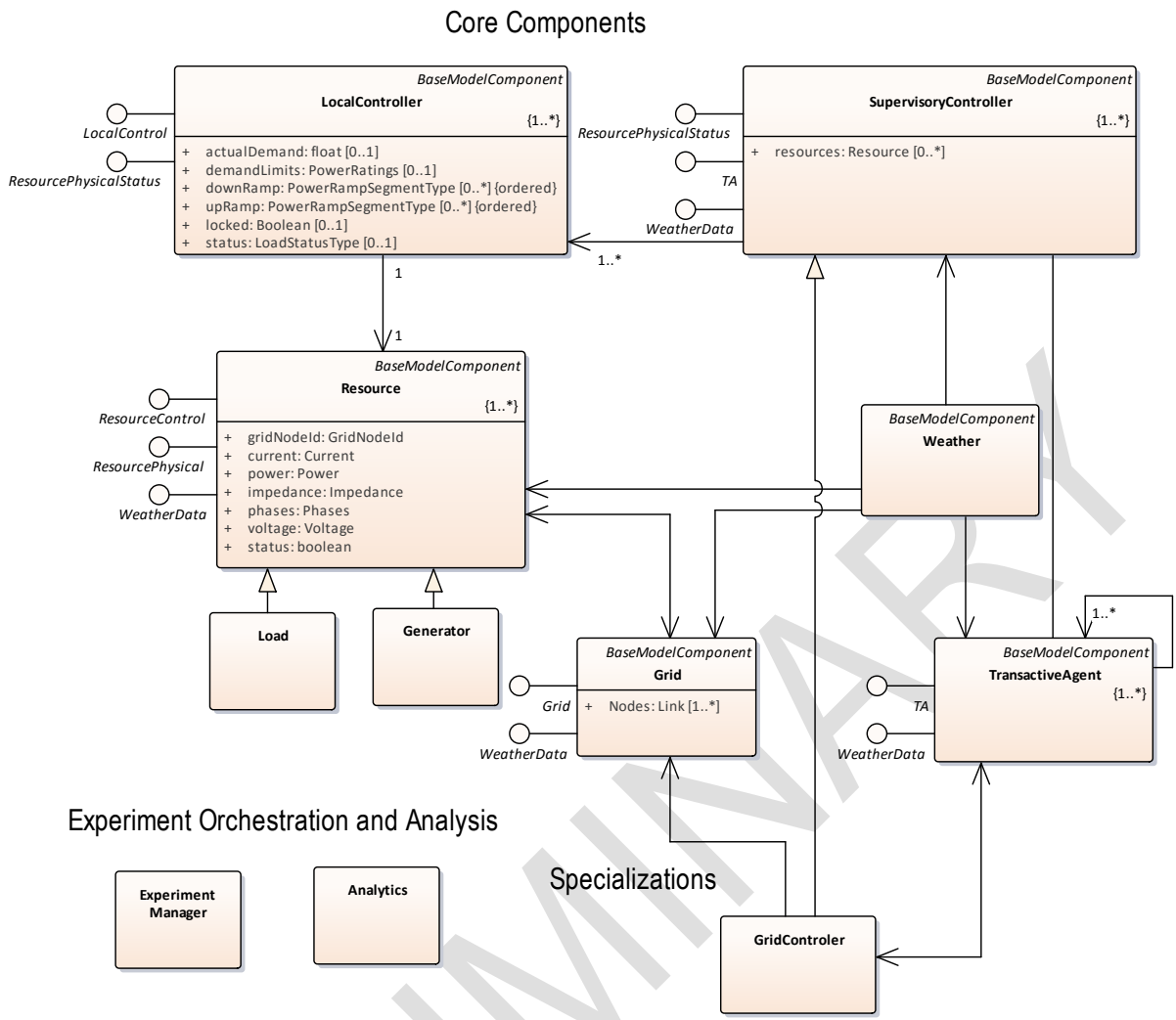


Figure 1: Model

1.3.1.1.1.2 Grid

A simulation of a power grid or grid segment.

A Grid consists of a set of Link structures that represent a network of interconnected nodes that comprise and energy system. Resources such as loads and generators are "attached" to the nodes of the Grid.

Name	Type	Notes
Nodes	Link	List of nodes in Grid

1.3.1.1.1.3 LocalController

A simple base controller that only knows how to control a resource based on a dimensionless modulation setting. It does not have awareness of any other component of the system.

Name	Type	Notes
actualDemand	float	This attribute defines the power being consumed by the device (as measured by the present subinterval demand) at the present time.
demandLimits	PowerRatings	The operational demand characteristics and their associated end points for the load.
downRamp	PowerRampSegmentType	This attribute defines the reduction in power over time when a load being partially or fully de-energized has a complex load reduction profile. For each element of the Load.downRamp array, the downRamp[n].rate defines the amount of power decrease and the downRamp[n].duration defines the length of time in seconds upon which the decrease is in effect. If the downRamp[n].beginRamp attribute is defined for a ramp segment, this is the initial value of the ramp segment; if it is not present the initial value of the ramp equals the ending value of the previous ramp segment.

Name	Type	Notes
		Although the downRamp attribute name implies that the rise is monotonically decreasing, individual array elements may have slopes less than, greater than, or equal to 0. However, the overall trend of the function shall be decreasing. The downRamp function shall measure the time from the load being fully energized until the power is completely depleted. If downRamp is not present, the power decrease to 0 shall be instantaneous. When a curtailable load is partially curtailed (less than the maximumCurtailableDemand) and curtailment is increased, the power will decrease starting at n-th downRamp of the sum from 0 to n of the downRamp[n].rate that is closest to the actualCurtailedDemand.
upRamp	PowerRampSegmentType	This attribute defines the increase in power over time when a load being partially or fully energized has a complex demand restoration profile. For each element of the Load.upRamp array, the upRamp[n].rate defines the amount of power increase and the upRamp[n].duration defines the length of time in seconds upon which the increase is in effect. If the upRamp[n].beginRamp attribute is defined for a ramp segment, this is the initial value of the ramp segment; if it is not present the initial value of the ramp equals the ending value of the previous ramp segment. Although the upRamp attribute name implies that the rise is monotonically increasing, individual array elements may have slopes less than, greater than, or equal to 0. However, the overall trend of the function shall be increasing. The upRamp function shall measure the time from the load being fully de-energized until the power is completely restored as defined by Load.maximumDemand. If this attribute is not present, the power increase upon restoration shall be instantaneous. When a curtailable load is partially curtailed (less than the maximumCurtailableDemand) and curtailment is reduced, the power will increase starting at n-th recoveryRamp of the sum from 0 to n of the upRamp[n].rate that is closest to the actualCurtailedDemand.
locked	Boolean	This attribute defines whether the load is locked and therefore ineligible for curtailment; or unlocked and available for curtailment. Load locking behavior changes depending on the load's curtailmentStatus attribute value at the time the load was locked. If the Load.locked attribute is set to TRUE and the: 1) load is not curtailable, the load is immediately locked. The behavior of this operation is a local matter; 2) curtailmentStatus is curtailmentInactive, the load will immediately be locked out from curtailment eligibility; 3) curtailmentStatus is curtailmentNoncompliant, the load will cycle to its curtailmentInactive state then immediately be locked out of curtailment eligibility; 4) If the CurtailableLoad supports multi-stage curtailment, the load will cycle to its curtailmentInactive state for the present curtailment stage and then be locked out of any further curtailment eligibility. 5) curtailmentStatus is curtailmentNoncompliant, the load will cycle to its curtailmentInactive state then immediately be locked out of curtailment eligibility; 6) If the CurtailableLoad supports multi-stage curtailment, the load will cycle to its curtailmentInactive state for the present curtailment stage and then be locked out of any further curtailment

Name	Type	Notes
		eligibility. Loads that are locked will remain in the locked state indefinitely until the Load.locked attribute is reset to FALSE. The mechanism used to unlock the load is a local matter.
status	LoadStatusType	This attribute defines the current status of the load. For non-curtailable loads, it provides the present communication status and reliability of the data. For curtailable loads, it also defines if the load is eligible for curtailment or why it is ineligible for curtailment.

1.3.1.1.1.4 Resource

Grid connected Resources can be loads, generators, storage devices. They consume or generate energy.

Resources are considered to be intelligent in that they can provide information about themselves and have a defined interface for local control over their operation.

In practice, there may be many details of a resource that a modeler may expose. Those shown in this model are minimum interfaces required to perform standardized simulations for transactive energy. Actual models will inherit from these more general interfaces to include the specialized behaviors and information exchanges.

Name	Type	Notes
gridNodeId	GridNodeId	Identifies grid node load is connected to.
current	Current	Power
power	Power	rate of change of power
impedance	Impedance	Energy over time step
phases	Phases	
voltage	Voltage	
status	boolean	

1.3.1.1.1.5 SupervisoryController

Individual agents that control devices; this does not need to be a one-to-one mapping of controller to device (e.g., this may include a non-transactive aggregator or volt-var control system). These include logic for such things as voltage regulators or protection devices and determine what state the device should be in and how it might progress to the next state. This explicitly does not contain transactive elements, but does contain non-transactive optimizations and operator and consumer actions. NOTE: this element will need to be extremely broad, since there are hundreds of different control variables that one may need access to.

Name	Type	Notes
resources	Resource	

1.3.1.1.1.6 BaseModelComponent

General Transactive Energy Model Component. This abstract component provides for the initialization of simulation model components.

Name	Type	Notes
name	char	Name of the component
description	char	

1.3.1.1.1.7 TransactiveAgent

Describes a transaction to occur at a given place, including an expression of value, logic for estimating that value (e.g., forecasts) and quantity (including limits), and rules for how “bids” are formed and how often they are presented. This would include all “market” functions including device level bidding (replacing a traditional controller/thermostat), large-scale optimization (e.g., it could be used to describe an ISO or double-auction), etc.

1.3.1.1.1.8 Weather

Basic weather model.

Name	Type	Notes
stateName	char	
cityName	char	
latitudeDegrees	double	
latitudeMinutes	int	
longitudeDegrees	double	
longitudeMinutes	int	

Name	Type	Notes
timeZoneOffset	int	
temperature	double	
windSpeed	double	
humidity	double	
pressure	double	
month	int	
day	int	
hour	int	
minute	int	
second	int	

1.3.1.1.1.9 GridControler

A specialization of the SupervisoryController that provides for supervisory control of the grid segment and represents the distribution system operator (DSO) or similar entity that can represent the grid management in a transactive energy scheme.

1.3.1.1.1.10 Load

A specialization of a resource that represents a customer premise based load.

1.3.1.1.1.11 Generator

A specialization of a resource that represents a customer premise based grid connected generation source.

1.3.1.1.1.12 Experiment Manager

The Federate Manager runs the experiment.

1.3.1.1.1.13 Analytics

Analyzes data and produces metrics of the scenario.

1.3.1.1.2 Interfaces

The interfaces for the model represent those messages that are received (subscribed). It is assumed that in order to invoke these interfaces, the source can publish the data.

In implementing this model, pub-sub or request-response can produce equivalent results and the arrows and data flowed interpreted appropriately to the message mechanism.

1.3.1.1.2.1 Interfaces diagram

This diagram illustrates the interfaces defined for the TE Challenge Common Platform Model. Shown are the core components of the model and those interfaces that they realize.

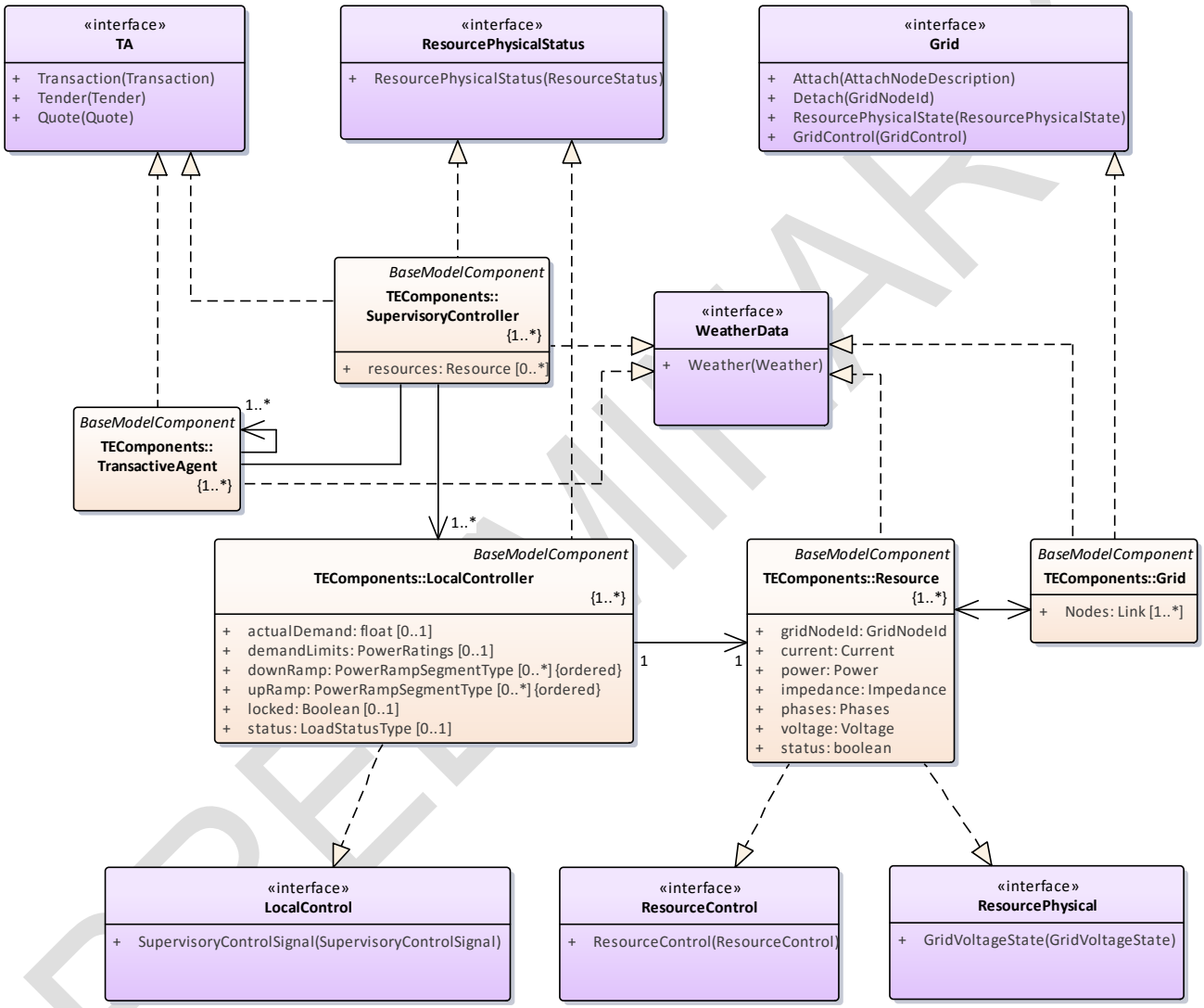


Figure 2: Interfaces

1.3.1.1.3 DataTypes

Data Types used in message exchanges.

1.3.1.1.3.1 DataTypes diagram

This diagram presents data types in three categories:

Component Model Data Structures -- defines the main data structures that represent grid state.

Interface Parameter Classes -- defines the content of interface messages.

Data Descriptions Imported from ASHRAE 201 -- FSGIM. These data definitions were modified from the standard to fit the data primitives of this modeling effort. They can be losslessly converted from FSGIM data types.

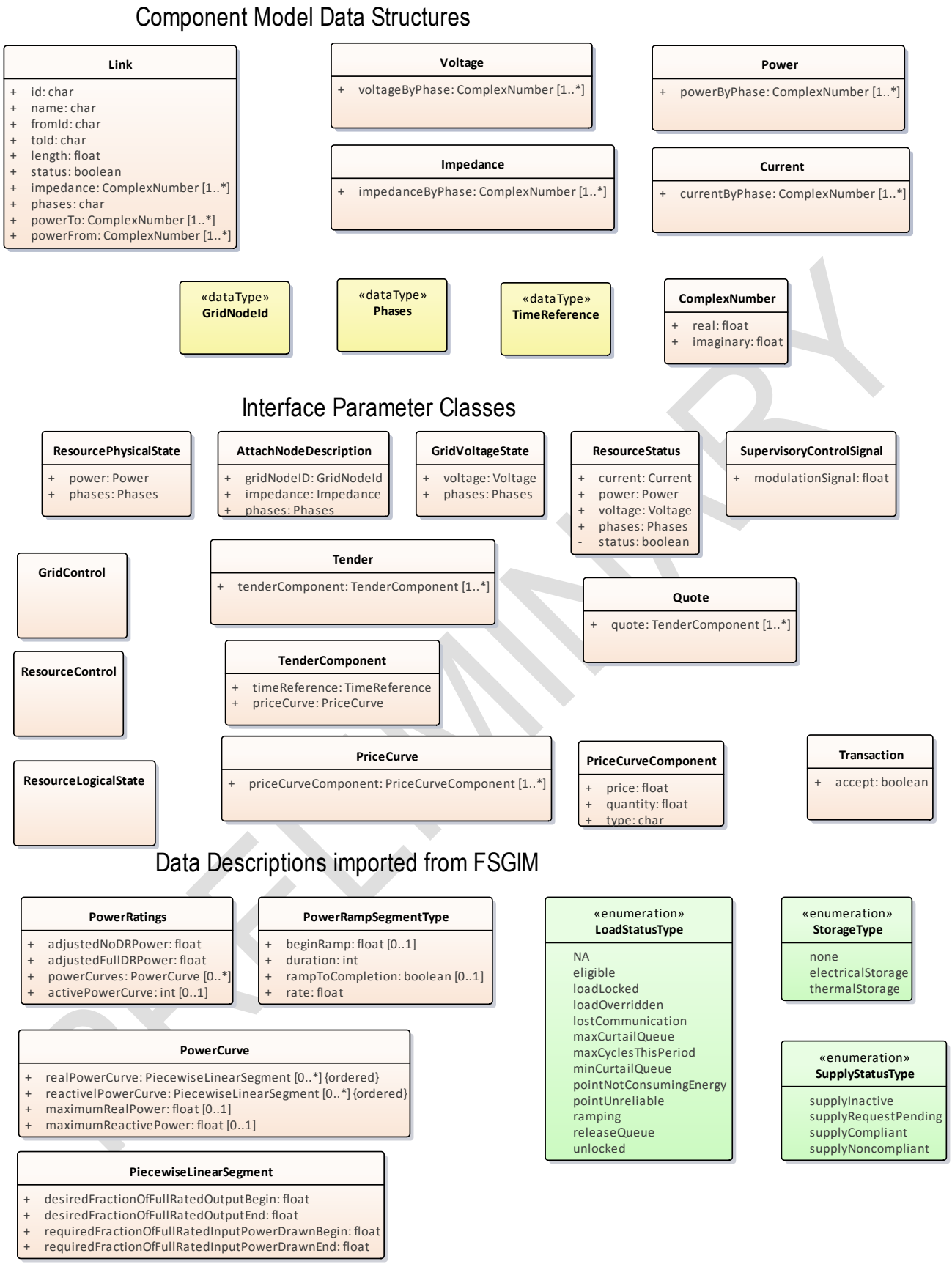


Figure 3: DataTypes

1.3.1.1.3.2 AttachNodeDescription

Parameters to attach a node to the Grid.

Name	Type	Notes
gridNodeID	GridNodeId	Node identifier to attach to in the Grid model.
impedance	Impedance	Impedance matrix for phase connections.
phases	Phases	Phases for attachment.

1.3.1.1.3.3 GridControl

GridControl service data structure.

1.3.1.1.3.4 GridVoltageState

Name	Type	Notes
voltage	Voltage	
phases	Phases	

1.3.1.1.3.5 PriceCurve

A price curve. Depending on sign of the PriceCurveComponent.quantity it can be price of supply or demand.

Name	Type	Notes
priceCurveComponent	PriceCurveComponent	A component of a price curve.

1.3.1.1.3.6 PriceCurveComponent

A component of a pricing curve.

Name	Type	Notes
price	float	Price of commodity.
quantity	float	Quantity of commodity (signed number) can be supply or demand.
type	char	Type of commodity -- W, Var, V, Frequency, Wh, ...

1.3.1.1.3.7 Quote

Name	Type	Notes
quote	TenderComponent	Array of tender components.

1.3.1.1.3.8 ResourceControl

Parameters used by local controller to control the resource.

1.3.1.1.3.9 ResourceLogicalState

1.3.1.1.3.10 ResourceStatus

The status of the resource provided describing current load and other conditions.

Name	Type	Notes
current	Current	
power	Power	
voltage	Voltage	
phases	Phases	
status	boolean	is resource active.

1.3.1.1.3.11 SupervisoryControlSignal

Name	Type	Notes
modulationSignal	float	Modulation control signal 0..1.0 for off (0.0) to full load or supply (1.0).

1.3.1.1.3.12 Tender

Name	Type	Notes
tenderComponent	TenderComponent	Array of time ordered tender components that provides a load or generation profile of price curves.

1.3.1.1.3.13 TenderComponent

A component of a tender component.

Name	Type	Notes
timeReference	TimeReference	Time reference for this tender component.
priceCurve	PriceCurve	Price curve for this time reference.

1.3.1.1.3.14 Transaction

Name	Type	Notes
accept	boolean	Accept the last quote.

1.3.1.1.3.15 ComplexNumber

A comlpex number

Name	Type	Notes
real	float	The real part.
imaginary	float	The imaginary part.

1.3.1.1.3.16 Current

Name	Type	Notes
currentByPhase	ComplexNumber	

1.3.1.1.3.17 Impedance

Name	Type	Notes
impedanceByPhase	ComplexNumber	

1.3.1.1.3.18 Link

Describes the physical components of the system and their internal state properties, such as power (or current) flow, current tap position, etc. It also includes topological information, such as “to” and “from”, which makes the nodal information implicit. Note that "to" and "from" simply identify two ends of the link and do not make a statement about the direction of flow of power/energy.

Name	Type	Notes
id	char	link segment id.
name	char	
fromId	char	node 0 id
toId	char	node 1 id
length	float	length of link segment in meters
status	boolean	true if connectivity established false if connectivity denied
impedance	ComplexNumber	matrix of impedances for each phase
phases	char	sequence of phases from A, B, C, N, L1, L2. e.g. three phase 4 wire -- ABCN Note phase order indicates the index in to the vector array of impedances or power or voltage. Voltage relative to N.
powerTo	ComplexNumber	Power into the "to" node.
powerFrom	ComplexNumber	Power into the "from" node.

1.3.1.1.3.19 PiecewiseLinearSegment

The PiecewiseLinearSegment class defines the attributes needed to specify a single straight line segment for a piecewise linear curve. Each straight line segment is specified by two X-axis coordinates, percentOFFullRatedOutputBegin and percentOFFullRatedOutputEnd; and by two Y-axis coordinates, percentOFFullRatedInputPowerDrawnBegin and percentOFFullRatedInputPowerDrawnEnd.

The entire piecewise linear curve is defined by the runningProfile attribute; where the 'percent of full rated input power' is a function of the 'percent of full rated output'. That is, as the output varies between 0 and 100 percent; the function maps to the percentage of input power (0..100) required to achieve the specified output.

Name	Type	Notes
desiredFractionOfFull	float	This attribute defines the starting x-coordinate of the straight line

Name	Type	Notes
RatedOutputBegin		segment.
desiredFractionOfFullRatedOutputEnd	float	This attribute defines the ending x-coordinate of the straight line segment.
requiredFractionOfFullRatedInputPowerDrawnBegin	float	This attribute defines the starting y-coordinate of the straight line segment.
requiredFractionOfFullRatedInputPowerDrawnEnd	float	This attribute defines the ending y-coordinate of the straight line segment.

1.3.1.1.3.20 Power

Name	Type	Notes
powerByPhase	ComplexNumber	

1.3.1.1.3.21 PowerCurve

This class describes the characteristics of a mathematical function used to estimate the power consuming characteristics of a load or the power generating characteristics of a generator.

Name	Type	Notes
realPowerCurve	PiecewiseLinearSegment	This attribute defines the real component of a single piecewise linear curve mapping the percentage of power consumed by the device as a function of the present level of operation of the device.
reactivelPowerCurve	PiecewiseLinearSegment	This attribute defines the reactive component of a single piecewise linear curve mapping the percentage of power consumed by the device as a function of the present level of operation of the device.
maximumRealPower	float	This attribute defines the maximum real power consumed (or supplied) by the device in units specified in PowerRealType.
maximumReactivePower	float	This attribute defines the maximum reactive power consumed (or supplied) by the device in units specified in PowerReactiveType.

1.3.1.1.3.22 PowerRatings

This class describes the power characteristics of a Load (or Generator) component. The attributes defined allow specifying the minimum and maximum expected power draw from the load (supply from a generator). It also allows a series of predefined operation power curves to be defined with one designated as presently being operational.

Name	Type	Notes
adjustedNoDRPower	float	This attribute defines the maximum expected power draw of a load (or the minimum power supplied by a generator) during operation. This value differs from the rated power since it may take into account operational considerations such as environmental, equipment safety or regulatory conditions.
adjustedFullDRPower	float	This attribute defines the minimum expected power draw of a load (or the maximum power supplied by a generator) during operation. This value differs from the rated power since it may take into account operational considerations such as environmental, equipment safety or regulatory conditions.
powerCurves	PowerCurve	This attribute defines one or more piecewise linear curves mapping the percentage of power consumed by the device as a function of the present level of operation of the device. Many loads draw power (or generators supply power) based on the present loading characteristics of the device. For example, a motor driving a fan will draw more power as the fan blade pitch is increased. The axes of the curve are defined in percent to allow loads of any type to utilize the attribute. For example a simple 60W incandescent light bulb would likely be modeled by a linear segment (0,0), (100,100). Here as the bulb is energized (dimmed) from off (0%) to fully on (100%), the power needed to energize the device also travels from 0% to 100%. A more complicated device such as a room air-conditioner may have a non-linear power curve. Here, as the coolness setting is adjusted from

Name	Type	Notes
		warmest to coolest, the air conditioner will draw relative more power when set to maximum cooling than when set to minimum cooling. When powerCurve is not present, the load or generator is assumed to be a two state device drawing no power when the device is off and adjustedNoDRPower when the device is on. When adjustedNoDRPower also is not present, the load or generator is assumed to be a two state device drawing no power when the device is off and maximumRealPower when the device is on.
activePowerCurve	int	This attribute defines the index into the zero based array of powerCurves indicating which powerCurve is presently active.

1.3.1.1.3.23 PowerRampSegmentType

The PowerRampSegmentType data structure is used to define a single array element of the recoveryRamp and stagingRamp array of the Load class. Each array element defines the beginning demand for the line segment and the rate of rise or drop. These attributes combined with the duration completely forms a line segment defining a portion of the ramp.

Name	Type	Notes
beginRamp	float	The attribute defines the quantity of power at the start of the ramp segment. If this attribute is not defined in the segment, the start of the ramp is assumed to be the end of the ramp of the previous segment.
duration	int	The attribute defines the time horizon in seconds upon which the associated rise or drop is valid.
rampToCompletion	boolean	The attribute defines whether the ramping up or down of this load may be halted in midstream (false) or once started must complete through all segments of the ramp (true). As an example, a multistate fan may only use a portion of the ramp, as it sequences from low to medium to high speed levels (false); whereas, a production line, once started, may need to run through its complete set of ramp segments (true). If the attribute is not defined it is assumed to be false.
rate	float	The attribute defines rate of rise (positive value) in demand or the rate of drop (negative value) in demand when a load either powers up or shuts down respectively. Its sister attribute, duration, defines the time frame upon which the rate is defined.

1.3.1.1.3.24 ResourcePhysicalState

ResourcePhysicalState describes physical state parameters for the resource.

Name	Type	Notes
power	Power	
phases	Phases	

1.3.1.1.3.25 Voltage

A complex vector of Voltage and a string enumeration of the phases

Name	Type	Notes
voltageByPhase	ComplexNumber	

1.3.1.1.3.26 TimeReference

A time reference, UTC.

1.3.1.1.3.27 GridNodeId

An integer representing a grid node.

1.3.1.1.3.28 Phases

A string indicating the list of phases involved -- an order subset of <A,B,C,N,L1,L2>

1.3.1.1.3.29 StorageType

An enumeration that defines the energy storage characteristics of an instance of the Generator Class.

Name	Type	Notes
none		This value indicates that this instance of the Generator Class models a device that does not produce energy from storage.
electricalStorage		This value indicates that this instance of the Generator Class models a device that produces electricity from storage.
thermalStorage		This value indicates that this instance of the Generator Class models a device that produces thermal energy from storage.

1.3.1.1.3.30 SupplyStatusType

This enumeration indicates if the load is presently curtailed and if curtailed is in compliance with the curtailment request received.

Name	Type	Notes
supplyInactive		This generator is not presently operating
supplyRequestPending		A request has been received and is pending.
supplyCompliant		The generator operation is compliant with the last request.
supplyNoncompliant		The generator is not compliant with the last request.

1.3.1.1.3.31 LoadStatusType

This enumeration provides the present overall state of the load.

Name	Type	Notes
NA		Not applicable
eligible		The load is presently communicating properly and its data values are correct. In addition, for curtailable loads this load is presently eligible to be curtailed.
loadLocked		The load is ineligible for curtailment since it has been locked.
loadOverridden		An external process has set this override attribute prohibiting curtailment.
lostCommunication		The load presently cannot be accessed.
maxCurtailQueue		The load is in curtailment and presently being timed for the maximum curtailment time.
maxCyclesThisPeriod		The load has been cycled the maximum number of times this period.
minCurtailQueue		The load is in curtailment and presently being timed for the minimum curtailment time.
pointNotConsumingEnergy		The load is ineligible for curtailment since the point associated with the load is already shut off and not consuming any energy.
pointUnreliable		The load is ineligible for curtailment since the point associated with the load is unreliable. 'Unreliable' is an error condition when the present value of a point is questioned due to some hardware or software failure. When a point is unreliable, it still may present a value (e.g., Space Temp Present Value = 67 DegF) but carries along a second attribute that indicates this value is suspect. When a point is in the unreliable state, it shall not be curtailed.
ramping		The load is ramping. That is, it is a transitional state and is either starting up or shutting down. While in this temporary state, it is not eligible for curtailment.
releaseQueue		The load is ineligible for curtailment. The load has completed its curtailment and is presently timing down the restore time before it is again eligible for curtailment.
unlocked		The load has recently been unlocked. It will analyze all conditions and set its present eligibility state after analysis completes.

1.3.1.2 Scenario

Presents experimental scenarios for Transactive Energy simulations.

1.3.1.2.1 Base TE Experiment Scenario diagram

Base scenario for TE experiments. This sequence diagram contains the following:

1. Initialization of all components by the Experiment Manager
2. Three parallel sequences that continue until experiment ends:

- Physical: represents the timing needed to perform multiphysics power and energy simulation.
- Logical Controller: represents the timing needed to perform supervisory and local control of resources.
- Transactive: represents the timing needed to perform a periodic transactive sequence resulting in a pricing model for the duration of the transactive step. This includes a "settle" loop for performing market/participant convergence on price.

This diagram illustrates the data flows and the target destinations of data for those components to use. In implementing this model, pub-sub or request response can produce equivalent results and the arrows and data flowed interpreted appropriately to the message mechanism.

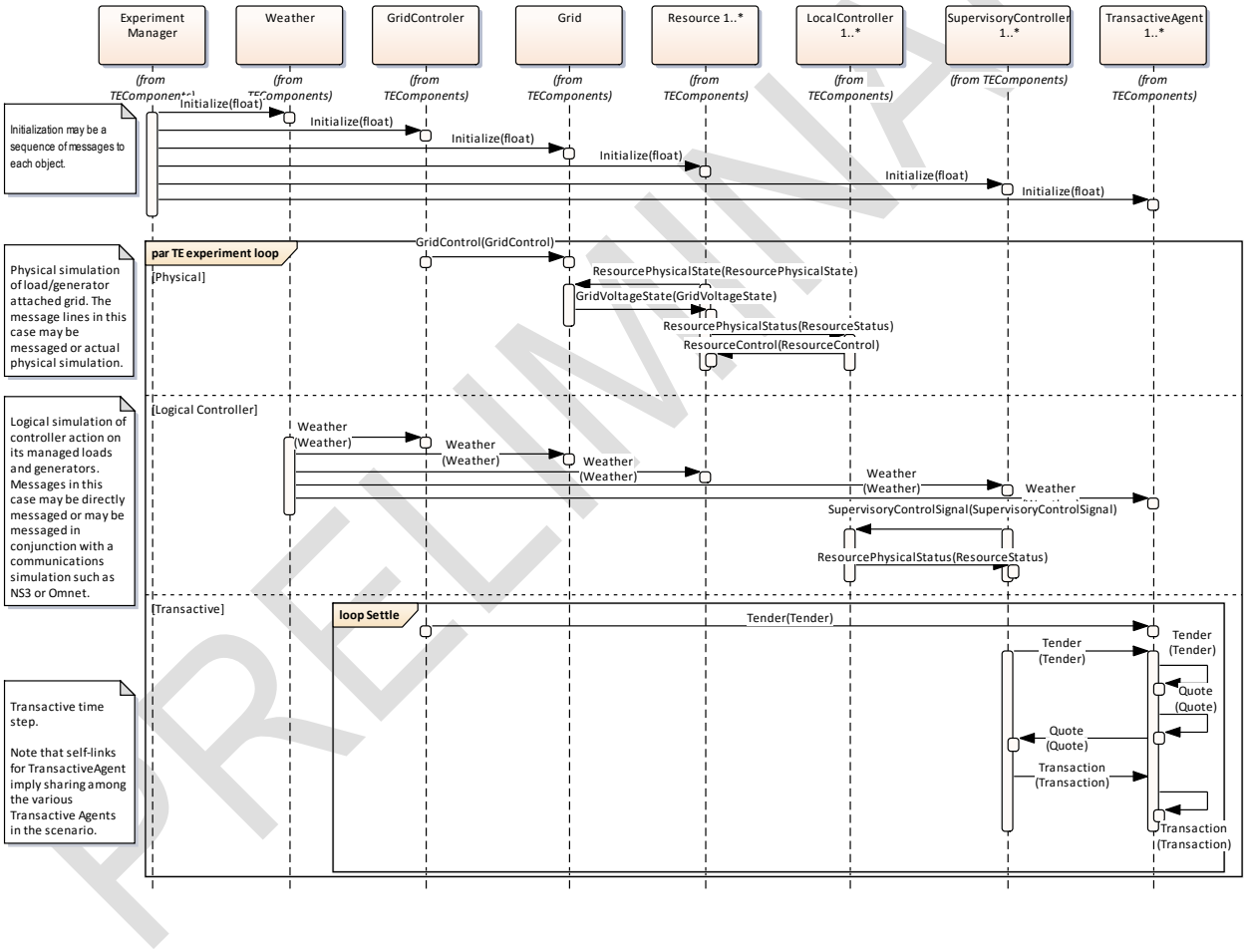


Figure 4: Base TE Experiment Scenario

1.3.1.3 Composite and Extended Classes

Composite classes allows for the composition and testing of classes that realize selected sets of interfaces. In any given instance of a TE Component, one or more of the roles or interfaces may be realized. Exchanges shown in the Base TE Experiment Scenario between components that have been combined do not occur during the experiment.

1.3.1.3.1 Composite Classes diagram

This diagram illustrates how components can be combined and/or extended for use in experiments of the common platform model.

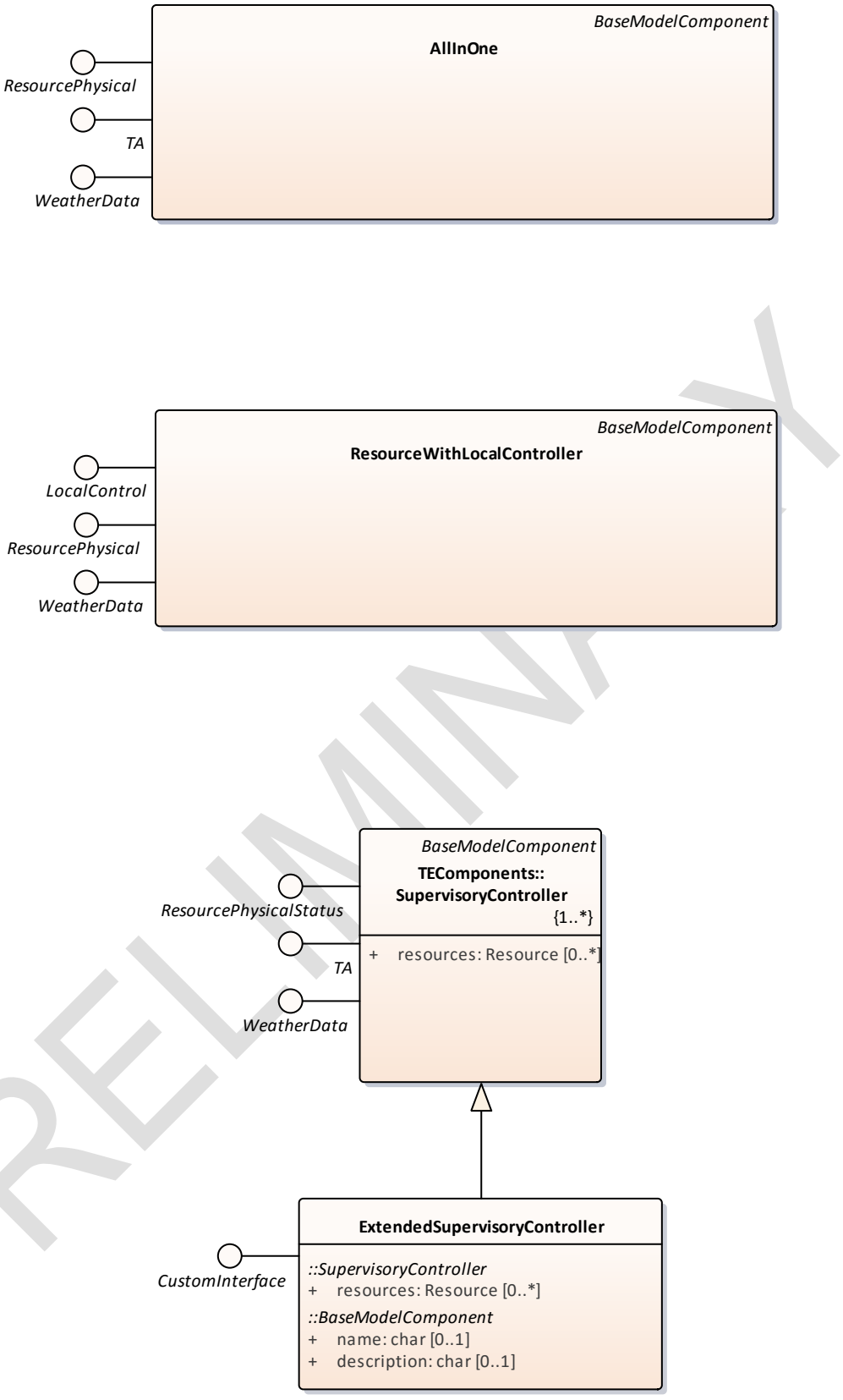


Figure 5: Composite Classes

1.3.1.3.2 AllInOne

Contains Resource, Local Controller, Supervisory Controller with Transactive interface.

1.3.1.3.3 ExtendedSupervisoryController

This extended SupervisoryController exposes an additional CustomInterface. By substituting this version of the SupervisoryController additional capabilities can be exposed when substituting from the base.

1.3.1.3.4 ResourceWithLocalController

Contains Resource with Local Controller.

1.3.1.4 BetaUseCase

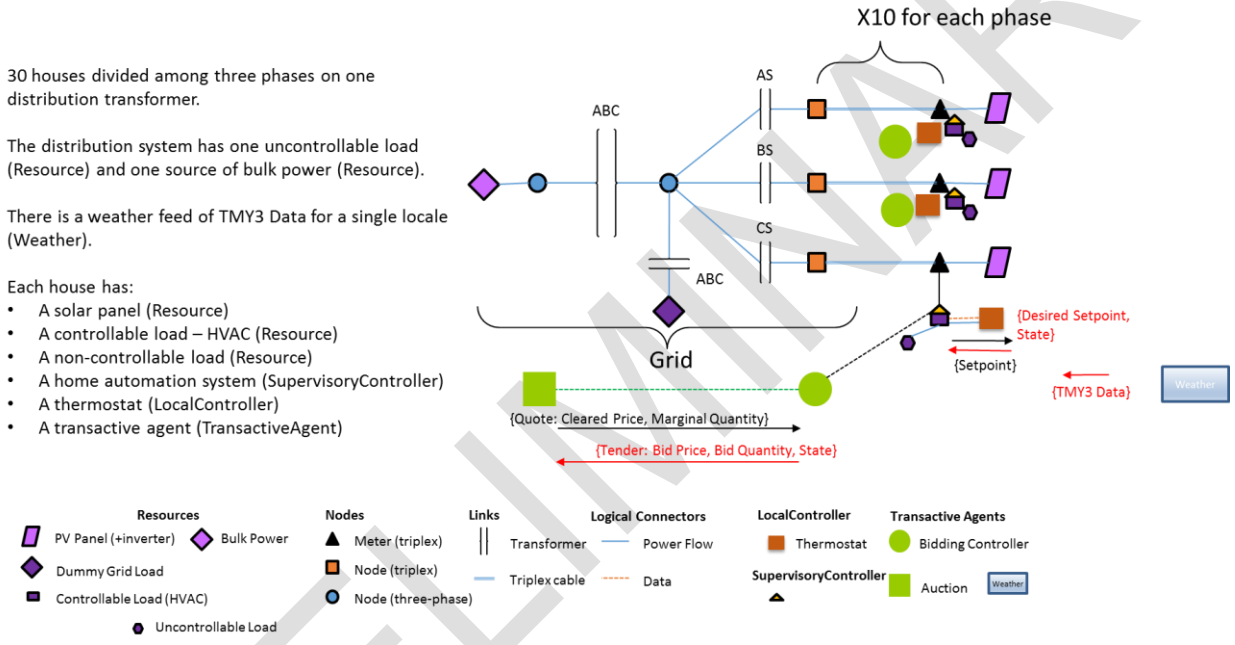
1.3.1.4.1 BetaUseCase diagram

The Beta Use Case is a scenario chosen to illustrate the interactions of the components of the TE Common Platform Model. Since its main goal is to illustrate the workings of the TE components and provide a reference simulation to use in developing more realistic and useful use cases, it should be considered in that context.

Once realized on several simulation framework platforms, it provides the basis for comparison and baselining of these platforms.

This use case is based on a gridlab-d model contributed by PNNL. The content herein abstracts that use case as a set of instance objects which inherit from the common platform components.

This is a work in progress.



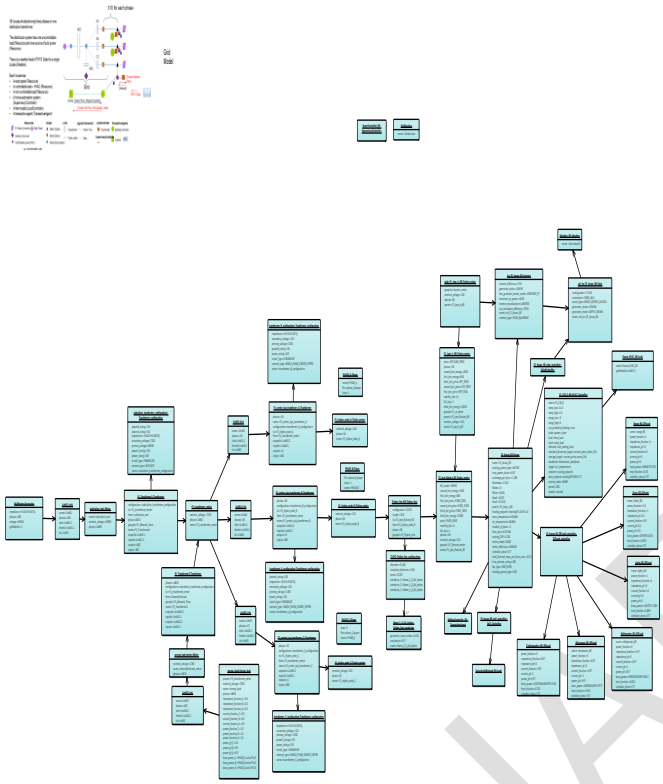


Figure 6: BetaUseCase

1.3.1.4.2 AdditionalClasses

1.3.1.4.2.1 AdditionalClasses diagram

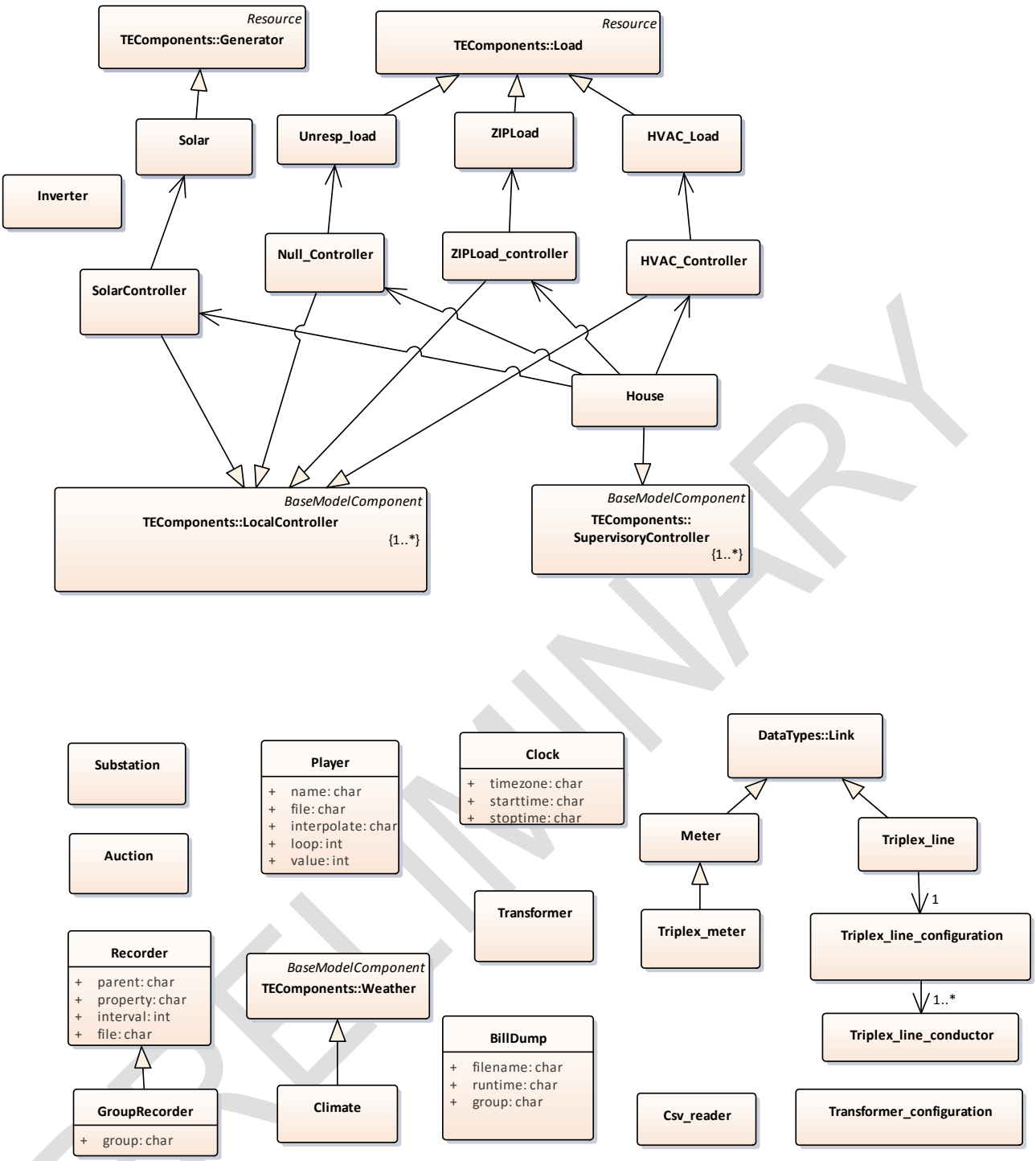


Figure 7: AdditionalClasses

1.3.1.4.2.2 Auction

Name	Type	Notes
current_price_stdev_24h	double	
current_price_mean_24h	double	

1.3.1.4.2.3 BillDump

Name	Type	Notes
filename	char	

Name	Type	Notes
runtime	char	
group	char	

1.3.1.4.2.4 Climate

Name	Type	Notes
tmyfile	char	
interpolate	char	
name	char	

1.3.1.4.2.5 Clock

Name	Type	Notes
timezone	char	
starttime	char	
stoptime	char	

1.3.1.4.2.6 Csv_reader

Name	Type	Notes
name	char	
filename	Voltage	

1.3.1.4.2.7 GroupRecorder

Name	Type	Notes
group	char	

1.3.1.4.2.8 HVAC_Controller

gridlab-d "controller"

Name	Type	Notes
name	char	
market	char	
period	int	
control_mode	char	
base_setpoint	char	
setpoint	char	
target	char	
deadband	char	
average_target	char	
standard_deviation_target	char	
demand	char	
total	char	
load	char	
state	char	
use_predictive_bidding	boolean	
range_high	int	
range_low	int	
ramp_high	int	
ramp_low	int	

1.3.1.4.2.9 HVAC_Load

1.3.1.4.2.10 House

Name	Type	Notes
name	char	
parent	char	
schedule_skew	int	
Rroof	boolean	
Rwall	double	
Rfloor	double	
Rdoors	int	
Rwindows	double	
airchange_per_hour	double	
hvac_power_factor	char	
cooling_system_type	char	
heating_system_type	char	
fan_type	char	
hvac_breaker_rating	int	
total_thermal_mass_per_floor_area	int	
motor_efficiency	char	
motor_model	char	
cooling_COP	double	
floor_area	double	
number_of_doors	int	
air_temperature	double	
mass_temperature	double	
heating_setpoint	char	

1.3.1.4.2.11 Inverter

Name	Type	Notes
name	char	
inverter_type	char	
use_multipoint_efficiency	char	
inverter_manufacturer	char	
maximum_dc_power	int	
four_guadrant_control_mode	char	
generator_status	GeneratorStatus	
inverter_efficiency	double	

1.3.1.4.2.12 Meter

Name	Type	Notes
nominal_voltage	int	

1.3.1.4.2.13 Null_Controller

1.3.1.4.2.14 Player

Name	Type	Notes
name	char	
file	char	
interpolate	char	
loop	int	
value	int	

1.3.1.4.2.15 Recorder

Name	Type	Notes
parent	char	
property	char	
interval	int	
file	char	

1.3.1.4.2.16 Solar

Name	Type	Notes
panel_type	char	
orientation	char	
rated_power	double	
generator_mode	char	
generator_status	char	

1.3.1.4.2.17 SolarController

1.3.1.4.2.18 Substation

Name	Type	Notes
name	char	
phases	Phases	
nominal_voltage	Voltage	
voltage_A	Voltage	
voltage_B	Voltage	
voltage_C	Voltage	
busType	Voltage	
reference_phase	char	

1.3.1.4.2.19 Transformer

Name	Type	Notes
inputs	Phases	
outputs	Phases	
inputId	GridNodeId	
outputId	GridNodeId	
name	char	
groupid	char	
phases	Phases	
from	char	
to	char	
configuration	char	

1.3.1.4.2.20 Transformer_configuration

Name	Type	Notes
name	char	
connect_type	char	
install_type	char	
power_rating	int	
powerC_rating	int	
primary_voltage	int	
secondary_voltage	int	
impedance	Impedance	
powerA_rating	int	
powerB_rating	int	

1.3.1.4.2.21 Triplex_line

Name	Type	Notes
groupId	char	
configuration	char	

1.3.1.4.2.22 Triplex_line_conductor

Name	Type	Notes
name	char	
resistance	double	
geometric_mean_radius	double	

1.3.1.4.2.23 Triplex_line_configuration

Name	Type	Notes
name	char	
conductor_1	char	
conductor_2	char	
conductor_3	char	
insulation_thickness	double	
diameter	double	

1.3.1.4.2.24 Triplex_meter

Name	Type	Notes
groupid	char	
parent	char	
price	char	
first_tier_price	char	
second_tier_price	char	
third_tier_price	char	
first_tier_energy	int	
second_tier_energy	int	
third_tier_energy	int	
bill_day	int	
monthly_fee	int	
bill_mode	char	

1.3.1.4.2.25 Unresp_load

Name	Type	Notes
base_power_A	char	
base_power_B	char	
base_power_C	char	
power_pf_A	double	
power_pf_B	double	
power_pf_C	double	
power_fraction_A	double	
power_fraction_B	double	
power_fraction_C	double	
current_fraction_A	double	
current_fraction_B	double	
current_fraction_C	double	
impedance_fraction_A	double	
impedance_fraction_B	double	
impedance_fraction_C	double	
parent	char	
nominal_voltage	int	

1.3.1.4.2.26 ZIPLoad

Name	Type	Notes
schedule_skew	int	
heat_fraction	double	
base_power	char	
power_pf	int	
current_pf	int	
current_fraction	int	
impedance_pf	int	
impedance_fraction	int	
power_fraction	int	

1.3.1.4.2.27 ZIPLoad_controller

1.3.1.4.3 DataTypes

1.3.1.4.3.1 DataTypes diagram

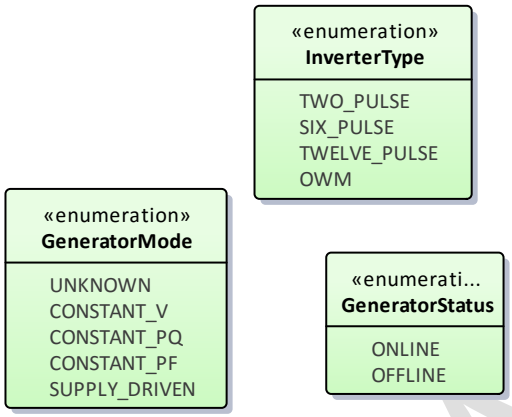


Figure 8: DataTypes

1.3.1.4.3.2 GeneratorMode

Name	Type	Notes
UNKNOWN		
CONSTANT_V		
CONSTANT_PQ		
CONSTANT_PF		
SUPPLY_DRIVEN		

1.3.1.4.3.3 GeneratorStatus

Name	Type	Notes
ONLINE		
OFFLINE		

1.3.1.4.3.4 InverterType

Name	Type	Notes
TWO_PULSE		
SIX_PULSE		
TWELVE_PULSE		
OWM		

1.3.1.4.4 Load_child