

Developement of a power grid ontology and a sample application

Felipe Oehrwald, Clemens von Schwerin

March 15, 2017

1 ONTOLOGY DETAILS

1.1 BUILDING BLOCKS

The power grid ontology is aimed to give a basis for representing power grids in terms of power production, consumption and transmission. For that purpose the following classes, as listed in figure 1.1a were designed.

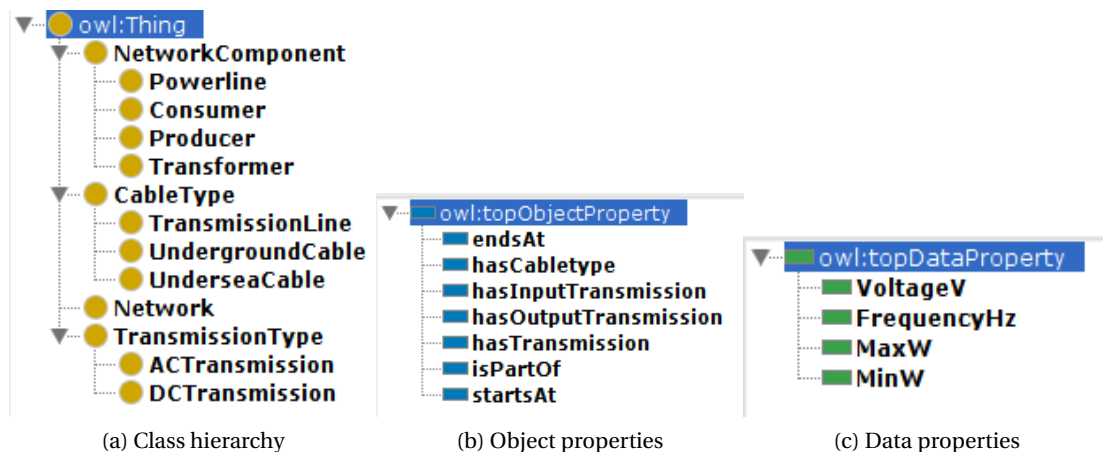


Figure 1.1: Building blocks of the Power Grid ontology.

The most basic building blocks are the *NetworkComponents* which model all physical entities a network may consist of and are disjoint. There are *Producers* adding power to the network, *Consumers* drawing energy from the network, *Powerlines* transporting the energy between exactly two other network entities and *Transformers* managing splits, merges and changes in the transmission type of *Powerlines*.

Those transmission types are modelled in the *TransmissionType* class. This class is divided into two disjoint subclasses *ACTransmission* for alternating current and *DCTransmission* for direct current.

Each *Powerline* furthermore has a *CableType* indicating the layout of a power line. There are 3 disjoint alternatives: regular *TransmissionLines*, *UndergroundCables* and *UnderseaCables*.

Finally every *NetworkComponent* is part of a specific *Network*. The relations between the different classes written in prose above are represented in the ontology as object properties. The domain and the range of every object property in this ontology are left blank on purpose, because they do not serve as constraint but as axioms for the reasoner. Constraints for the object properties however are represented as subclass rules in the power grid ontology. Every class relating to another one via an object property has a subclass rule of the type

(object property name) exactly 1 (other classes name)

e.g. in *NetworkComponent* the following subclass rule exists:

isPartOf exactly 1 *Network*

Apart from the *isPartOf* property for *NetworkComponents* other properties exist to model the relations described above. *hasOutputTransmission* maps a *TransmissionType* to every *Producer*, *hasInputTransmission* does the mapping between *TransmissionType* and *Consumer* and finally *hasTransmission* relates a *Powerline* with its *TransmissionType*. Furthermore for expressing the *CableType* of a *Powerline* the *hasCableType* entity exists. Furthermore *Powerlines* need a starting- end an endpoint so *startsAt* allows for *Producers* or *Transformers* as starting points while *endsAt* respectively allows for *Consumers* and *Transformers* as endpoints.

Finally the ontology defines some data properties which quantify important measures in a power grid. For starters every *Producer* and *Consumer* has a maximum power consumption expressed by the *MaxW* property and a minimum power consumption expressed by the *MinW* property, which are both quantified by integers. Furthermore every *TransmissionType* uses a certain voltage, which is represented in the ontology by the integer property *VoltageV*. On top of that every *ACTransmission* also has a certain frequency of phase shift, which in the ontology is described by the integer property *FrequencyHz*.

Apart from the building blocks describing a power grid we also created entities forming a sample power grid.

1.2 A SAMPLE POWER GRID

All entities involved in the sample power grid can be seen in 1.2. The power grid uses three different types of transmission. The energy of all producers is gathered in the *HighToMediumTransformer*. While the current received from the *OffshoreWindFarm* via long range transmission line using the *DCTransmission 800kVDC* is transformed to a medium energy level

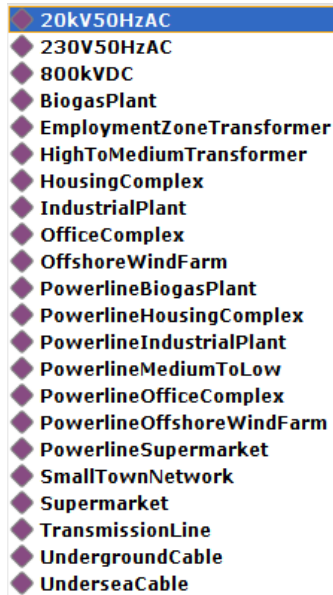


Figure 1.2: Entities of the sample power grid.

ACTransmission *20kV50HzAC*, the current from the *BiogasPlant* already is transmitted as *20kV50HzAC*. The *HighToMediumTransformer* distributes energy to an *IndustrialPlant*, a *HousingComplex* and an employment zone having its own *EmploymentZoneTransformer*. That transformer transforms the incoming current to a *230V50HzAC* ATransmission. Connected to that transformer are a *Supermarket* and an *OfficeComplex*.

All connections between the mentioned Producers, Consumers and Transformers are Entities of the *Powerline* type and named *Powerline....* Each of those power line have one of the three defined CableTypes: *TransmissionLine*, *UnderseaCable* or *UndergroundCable*.

Each NetworkComponent involved is part of the *SmallTownNetwork*.

Filling in every possible data and object property involved in the construction of this sample network would have been a lot of redundant work. Therefore a set of SWRL Rules was created that allows a reasoner to induce some network properties.

1.3 SWRL RULES

The power grid ontology uses four rules for inducing a *TransimssionType* for a *NetworkCom-
ponent*, as can be seen in figure 1.3.

	Na...	Body	Comment
✓	S1	<code>startsAt(?pl, ?s) ^ hasTransmission(?pl, ?tt) ^ Powerline(?pl) -> hasOutputTransmission(?s, ?tt)</code>	
✓	S2	<code>hasTransmission(?pl, ?tt) ^ endsAt(?pl, ?e) ^ Powerline(?pl) -> hasInputTransmission(?e, ?tt)</code>	
✓	S3	<code>Producer(?power_gridp) ^ startsAt(?power_gridline, ?power_gridp) ^ hasOutputTransmission(?power_gridp, ?power_gridtt) -> hasTransmission(?power_gridline, ?power_gridtt)</code>	
✓	S5	<code>Consumer(?power_gridc) ^ endsAt(?power_gridline, ?power_gridc) ^ hasInputTransmission(?power_gridc, ?power_gridtt) -> hasTransmission(?power_gridline, ?power_gridtt)</code>	

Figure 1.3: SWRL Rules used in the power grid ontology.

The first rule says that if a *Powerline* starts at s and has *TransmissionType* tt , s also has to have transmission type tt .

The second rule is the same for the endpoint: If a *Powerline* ends at e and has *TransmissionType* tt , e also has to have transmission type tt .

The third rule says if a *Powerline* starts at a *Producer* having *TransmissionType* tt it also has to have transmission type tt .

Finally the fourth rule again reflects the third one only for endpoints: If a *Powerline* ends at a *Consumer* having *TransmissionType* tt it also has to have transmission type tt .