

# PNRG – A Library for Modeling Variable Structure Energy Grids in Modelica using Energetic Petri Nets

Christian Gutsche<sup>1,2</sup> Zizhe Wang<sup>1,2</sup> Sebastian Götz<sup>2</sup> Volodymyr Prokopets<sup>2</sup> Uwe Aßmann<sup>2</sup>

<sup>1</sup>Boysen–TU Dresden–Research Training Group, Dresden, Germany

<sup>2</sup>Chair of Software Technology, Technische Universität Dresden, Dresden, Germany, {christian.gutsche, zizhe.wang, sebastian.goetz1, volodymyr.prokopets, uwe.assmann}@tu-dresden.de

## Abstract

Operating energy grids with a high share of renewable energy sources (RES) requires system reconfiguration as a response to environmental condition changes. To understand them better, simulations are needed and Modelica is an excellent choice for that. Energy grids with event-based reconfigurations are an instance of variable structure systems (VSS). However, the full support of VSS in Modelica is challenging and topic of ongoing research. Petri nets (PNs) offer a formalism for modeling VSS. The capability to simulate PNs in Modelica gives an opportunity to model VSS in Modelica. This paper presents an approach to utilize PNs in Modelica for modeling variable structure energy grids. Therefore, we introduce energetic Petri nets, a special type of PNs and an experimental library called PNRG for PN-based energy system modeling is presented. Furthermore, possibilities and limits of modeling VSS energy grids are discussed and an outlook how to develop this technique is provided.

*Keywords:* Energy Grids, Sector Coupling, Variable Structure Systems, Context-oriented Programming, Petri Nets

## 1 Introduction

The climate change forces societies to radically change the way of providing energy to their residents. Energy grids with a high share of renewable energy sources (RES) have to handle high fluctuations between the production and consumption of different energy media. To mitigate the negative effects and optimize the energy efficiency, so called sector coupling is investigated. This principle describes the combination of different energy systems. For example, electrical, gas and heat systems are coupled so that an oversupply of one energy medium can compensate the lack of another one. Sector coupling is nowadays broadly discussed in science, as (Fridgen et al. 2020) describes. However, the complexity of systems grows with an increased sector coupling and therefore modeling of energy grids gets even more important. With Modelica, there is a widely used modeling language which is well suited to simulate such complex systems.

The Modelica library *TransiEnt* (Senkel et al. 2021) is an excellent choice to simulate exactly this type of energy

systems (Heckel et al. 2022). It provides a magnificent level of detail and includes many components which are needed for realistic energy grid simulations. However, *TransiEnt* only works with the *Dymola* environment for now.

Another problem with simulating energy grids with a high share of RES and sector coupling is the missing possibility to fully describe and simulate so called variable structure systems (VSS) in Modelica. While Specific types of VSS can be simulated, mostly basic definitions as in (Utkin 1977), some more advanced VSS types start to cause problems (Tinnerholm 2022). For example, when the number of equations or variables is varying.

While Modelica simulations based on PNs are also limited in the way of simulating VSS with dynamically changing equations or varying numbers of variables and components, they allow to describe the variability of models and their state transitions. To make component based modeling using PNs accessible in Modelica, a special type of PNs, energetic PNs will be defined. They are also necessary to realize instantaneous effects on PN components within the energetic Petri net after local changes. These energetic Petri nets will be used to implement a Modelica package which allows to model VSS energy grids. In the following, two example models using this package are presented and it will be shown, that VSS can be modeled with that.

### 1.1 Variable Structure Energy Grids

The power supply of RES power plants is highly depended on the availability of natural energy sources like sun and wind energy. Therefore ensuring a reliable power supply from renewable energy sources requires a focus not only on sunny and windy days but also on cloudy days with low wind conditions, particularly when operating power grids with a high share of RES. Various storage systems are discussed to be used for mitigating these effects. As (Alhamali et al. 2016) show, there are a lot of possible solutions with different advantages and disadvantages. All storage systems need additional infrastructure which might be expensive. In some cases the location does not allow to install specific plants.

Another way to react to supply or demand changes is to activate and deactivate producers or consumers during

operation time. Additionally, special grid designs can improve the resilience of power grids. (Liu et al. 2022) provides a general overview of so called micro grids (MGs) to stabilize larger power grids. The investigation of such designs is an interesting objective for research. When simulating the variability of such power grids, structural changes must be performed during the simulation's run time.

## 1.2 Variable Structure Systems in Modelica

Systems with a variable structure are quite natural. For example, opening windows when the air quality is bad or covering windows when it is too sunny outside are such structural changes of a room system. Therefore, it is not surprising that VSS are studied for a long time as (Utkin 1977) already recognises in 1977. However, mathematically these VSS are not easy to describe. Especially for modeling there is a lot of ongoing research in this field. Some theoretical background is provided by (Benveniste et al. 2019).

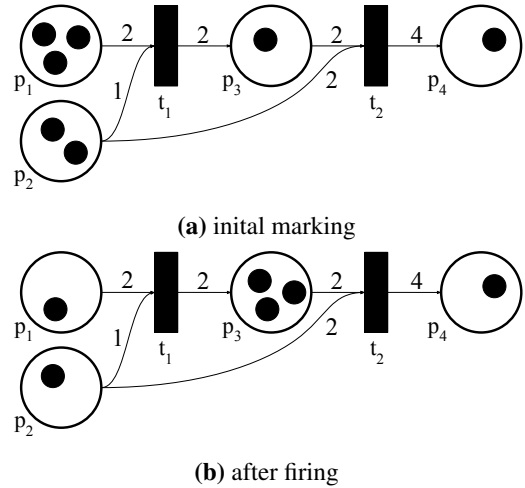
For Modelica there is a significant interest in increasing the possibilities for VSS modeling. Different approaches to simulate VSS in Modelica have been investigated and proposed, e.g. MOSILAB (Nytsch-Geusen et al. 2005), MoVasE (Esperon, Mehlhase, and Karbe 2015) and DySMo (Mehlhase 2015). Also, other solutions are proposed like the modeling language Sol introduced by (Zimmer 2010). However, all these approaches realized VSS in Modelica only partly or implemented VSS only in a subset of Modelica. Recently, (Tinnerholm, Pop, Sjölund, et al. 2020) presented an approach to make the modeling of VSS available in Modelica using a new compiler based on the Julia programming language which is discussed in detail in (Tinnerholm, Pop, and Sjölund 2022).

While investigating VSS, two aspects need to be considered. The studied system must be described clearly. This means not only the components itself have to be defined well but also how these components behave in different states of the system. The other aspect is the controlling of the systems states. In complex systems a lot of conditions can occur under which state transitions shall be performed. But not only the definition of these rules can become very complicated. Also the approval of a correctly working state control is an important but complex topic itself. A secure formalization of these state transitions must therefore be defined. So called PNs are an excellent choice for systems with a large and complex state space.

## 1.3 Petri Nets

PNs were firstly described by Carl Adam Petri in his PhD thesis (Petri 1962) but got extensively expanded afterwards. The following definition of a basic Petri net is given in (Murata 1989).

**Definition 1.1** (Petri Net). A Petri net is a graph defined by the 5-tuple  $PN = (P, T, F, W, M_0)$  with the following definitions:



**Figure 1.** An example of a simple discrete Petri net. (a) shows the initial state of the PN and (b) the state after the first firing which is also the final state.

1.  $P = \{p_1, p_2, \dots, p_m\}, m \in \mathbb{N}$  is a finite set of places
2.  $T = \{t_1, t_2, \dots, t_n\}, n \in \mathbb{N}$  is a finite set of transitions
3.  $F \subseteq (P \times T) \cup (T \times P)$  is a set of arcs
4.  $W : F \rightarrow \mathbb{N}_{>0}$  is a weight function
5.  $M_0 : P \rightarrow \mathbb{N}$  is the initial marking
6.  $P \cap T = \emptyset$
7.  $P \cup T \neq \emptyset$

This definition results in a directed graph composed of places and transitions which are connected with arcs. Every place  $p$  has a number of tokens  $M(p) \in \mathbb{N}$  and every arc has a weight  $w$ , here denoted as  $w_{in}$  when it points to a transition and  $w_{out}$  when it points to a place. How the PN works is explained with the help of Figure 1.

A transition  $t$  fires if  $W(p_i, t) \leq M(p_i)$  for all places  $p_i$  which are connected to the transition via the arcs with the weights  $w_{in,i}$ . In Figure 1 transition  $t_1$  has two input places  $p_1$  and  $p_2$ . The weight of the arc between  $p_1$  and  $t_1$  is  $W(p_1, t_1) = 2$  while the weight for  $p_2$  and  $t_1$  is  $W(p_2, t_1) = 1$ . Since  $M(p_1) = 3 \geq 2 = W(p_1, t_1)$  and  $M(p_2) = 3 \geq 1 = W(p_2, t_1)$  the transition fires. Transition  $t_2$  can not fire since  $M(p_3) = 1 < 2 = W(p_3, t_2)$  which is the weight of the arc between  $p_3$  and  $t_2$ . The arc weights define the number of tokens which are taken from the input places and given to the output places. Therefore, both  $p_1$  and  $p_2$  now have 1 token and since the output weight of  $t_1$  is  $W(t_1, p_3) = 2$ , the place  $p_3$  has now 3 tokens. Since  $M(p_1) = 1 < 2 = W(p_1, t_1)$  and  $M(p_2) = 1 < 2 = W(p_2, t_2)$ , neither the firing condition of  $t_1$  nor  $t_2$  is fulfilled after the first firing so that there will be no further firing. The PN is in its final state.

There are many extensions to these basic discrete Petri nets. Here, only some notable extensions are mentioned.

The introduction of inhibitor arcs, firstly described in (Agerwala 1974) is simple yet powerful. They allow to check if a token number is smaller than a specific threshold, and enable firing in this case without consuming tokens. Originally this threshold was 1 and therefore inhibitor arcs were used to check if places are empty. However inhibitor arcs can also be used with thresholds  $> 1$ . Test arcs (David and Alla 2008) allow to check if a token number is larger than a specific threshold and they enable the firing of transitions without consuming tokens. They can be used to prioritize transitions and therefore avoid conflicts when two transitions have the same input place.

Another important extension are continuous Petri nets (David and Alla 1987). For those the tokens are real numbers. Their transition firing is not described by discrete token transport but continuous flows. Therefore, those are a limit case of discrete PNs with token numbers  $\gg$  arc weights (David and Alla 2008). How this type of PNs works is explained by the following example. A continuous Petri net consisting of a continuous place  $p_1$  with no input arc, an initial token number of  $M(p_1) = 2$  and an output arc which connects  $p_1$  to a transition  $t_1$  with  $W(p_1, t_1) = 4$  would behave the following way. The token number of  $p_1$  would decrease with a constant rate of  $4 \frac{1}{s}$  until the token number is 0. So the number of tokens at the time  $t$  is given as the equation  $M(p_1, t) = -4t + 2$  for  $t < 0.5s$  and afterwards  $M(p_1) = 0$ . These Petri nets are powerful since they can represent systems of ordinary differential equations.

Petri nets containing discrete and continuous places and transitions are called hybrid petri nets. The modeling possibilities with Petri nets are significantly enhanced with these hybrid PNs since continuous systems like fluids flowing in a pipe system can be modeled with such PNs.

In Modelica, the open-source library PNLlib (Proß and Bachmann 2012) can be used for integrating PNs for modeling systems. PNLlib implements the basic discrete PNs as well as the test and inhibitor arcs extension, continuous Petri nets and other special transitions types.

## 1.4 Petri Nets and Variable Structure Systems

Why are those automata interesting for studying VSS? Petri nets can be used as state machines what is well described in (David and Alla 2008) but PNs are much more powerful. They can define many sub-states in parallel and how these affect each other. Additionally there are numerous verification methods to investigate the Petri net properties like reachability or liveness. An overview for some of these model checking approaches is given in (Wolf 2019). Therefore, PNs are interesting for controlling the state transitions of VSS.

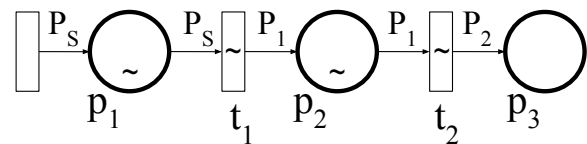
However, this is not the only connection between VSSs and Petri nets. In (Mai et al. 2018), so called adaptive Petri nets were introduced. In these adaptive PNs, sub-nets can be activated and deactivated based on the marking of the PN. Therefore, the Petri net itself is a variable structure system where the state transition controlling is

integrated in the system itself. Furthermore, (Mai et al. 2018) presents an algorithm to flatten adaptive Petri nets to basic PNs with inhibitor arcs.

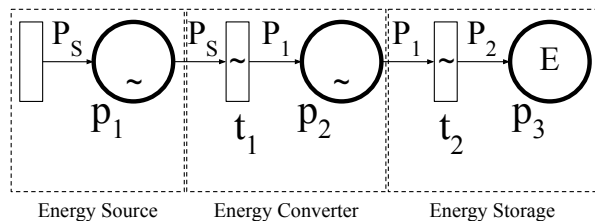
In (David and Alla 2008) several application examples for modeling based on Petri nets are described, such as gas storage systems or a four-stroke engine. Examples for PN-based modeling in Modelica are presented in (Proß and Bachmann 2012).

For the goal of enabling the expression of VSS in Modelica, the application of Petri nets appears to be a promising approach. In the following, a way to use PNs for modeling power grids will be presented.

## 2 Energy Grid modeling and Energetic Petri Nets



(a) Example of a simplified PN representing an energetic flow.



(b) Possible interpretation of the different parts of the energetic Petri net.

**Figure 2.** An example of a simple energetic Petri net. Energetic places and transitions are denoted with a “~”. Transitions and places without a “~” are continuous transitions and places.

The modeling of VSS energy grids is not possible in Modelica when using large physical models. Therefore, a method reducing the modeling to a problem accessible in Modelica is needed. An alternative approach based on Petri nets is capable of doing this and is presented in the following.

Modeling electrical grids with Petri nets was investigated for example in (Lu et al. 2016). However, these modeling techniques lack user friendliness and do not contain sector coupling. The component based modeling in Modelica would help to increase the usability of this modeling method. To study the usage of Petri nets for the component based modeling of VSS power grids, a simplified base model is used which just takes the power generation and consumption into account. This means other properties like phase angles or the frequency that are also crucial for the grid stability are neglected in this approach. However, the main goal of the model discussed here is to

validate the assumption, that it is possible to model VSS using Petri nets.

Modeling the power generation and consumption leads to the idea to model the flow of energy through the power grid. Since in continuous Petri nets arc weights represent the derivative of the place tokens, the natural choice for PN representations for energy and power are the following: The tokens in places represent stored energy and therefore, the transition arcs will represent the electrical power of components.

However, energy can not be stored in every case. So during modeling, it is important to make sure that energy is not stored in places where it should not be stored. To ensure this, two possible solutions can be implemented. The first method is to realize all grid components in a single transition whenever energy is not allowed to be stored in-between, instead of creating one transition for every grid component with places between the transitions. This is very disadvantageous, e.g. for component-based modeling or when a sub-net needs to be activated adaptively. Therefore, it is preferable to define so called energetic Petri nets that cover the needs of energetic flow modeling by definition.

**Definition 2.1** (Energetic Petri Net). An energetic Petri net is a hybrid Petri net with inhibitor arcs and:

1. Energetic places  $P_E$  with  $N \in \mathbb{N}_{>0}$  input and  $M \in \mathbb{N}_{>0}$  output arcs, that behave like continuous places with:

$$\sum_{i=1}^N w_{in,i} = \sum_{j=1}^M w_{out,j} \quad (1)$$

2. Energetic transitions  $T_E$  with  $K \in \mathbb{N}_{>0}$  input and  $L \in \mathbb{N}_{>0}$  output arcs that behave like continuous transitions with:

$$\forall j : w_{out,j} = f(w_{in,1}, \dots, w_{in,K}) \quad (2)$$

and

$$\sum_{i=1}^K w_{in,i} \leq \sum_{j=1}^L w_{out,j} \quad (3)$$

for normal arcs and no restriction for inhibitor or test arcs.

This definition ensures three properties. At first, the mentioned flow in places where no energy should be stored is given. Also, energy can not be produced but only converted with loss due to the efficiencies of the components. For dynamic simulations it is also important to not only have constant weights but weights that can change over time. A definition for such PNs was already introduced in 1978 by (Valk 1978). These self-modifying PNs allow to represent more complex ordinary differential equation systems with PNs using the token numbers of places as arc weights. In combination with equation 2 this leads to a Petri nets where the change of an arc

weight leads to instantaneous changes of weights in following arcs within the energetic PN.

How can these Petri nets be used to model energy flows? As shown in Figure 2, power sources like wind or solar energy can be modeled as continuous transitions without input arcs and varying weights of the output arcs. The output arc weight should be the source power  $P_S$ . Since these natural energy sources cannot be stored directly, the source transition is connected to an energetic place which instantaneously transfers the energy to an energetic transition which can represent simplified power plants converting the energy to electrical energy using a physical equation  $P_1 = f(P_S)$ . The last place is an ordinary continuous one and can be interpreted as an ideal energy storage.

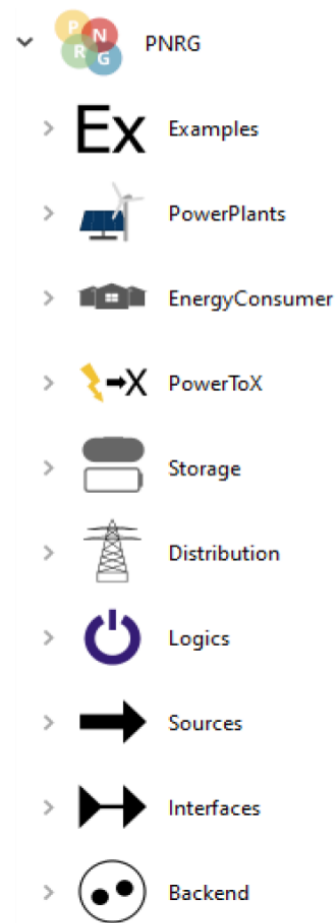
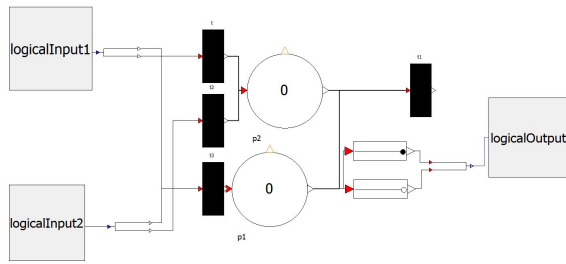


Figure 3. Structure of the PNRG package.

### 3 PNRG Library – Petri Net based Renewable Energy Grids

For a user-friendly and reusable modeling, the different sub-nets representing components of the energy grid should be programmed as Modelica models that can be connected accordingly. Therefore, a package based on the PNlib Petri nets was developed for component-wise mod-

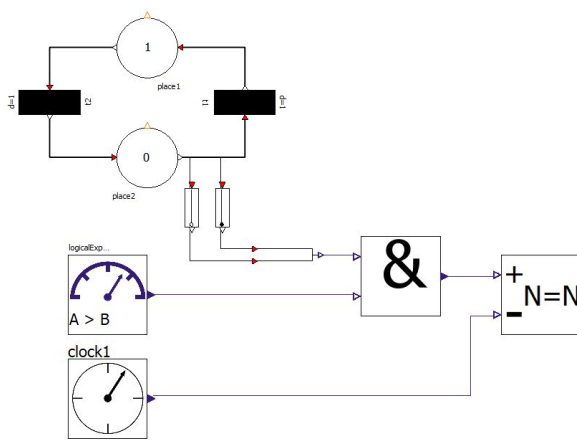


**Figure 4.** Example of a PNlib implementation of an AND gate. A token number of 1 represents true and a token number of 0 represents false.

eling of different energy grids. However, the package is still in an experimental state. In this section we will briefly introduce the main aspects of this package.

Figure 3 shows the structure of the PNRG package. Examples for specific Models are photovoltaic power plants, wind power plants, electricity consumers, electrolyzers, batteries, transformers and logic components like AND gates. To implement the energetic places and transitions, the continuous place and transition models from PNlib were modified.

### 3.1 Control Layer



**Figure 5.** Example for different logical elements based on Petri nets. A simple PNlib Petri net, a module to convert boolean expressions to logical inputs and a clock are connected to an integer controller using an AND gate.

As described in 1.4, the first application of PNs for VSS is the control of state transitions. For switching PNRG components on and off, logical interfaces are defined. These logical interfaces have a combination of test and inhibitor arcs as input and output. Both of these arcs are needed to express logical expressions satisfactorily. This allows to build logical gates. Exemplary, an implementation of an AND gate built using PNlib Petri nets is shown in Figure 4.

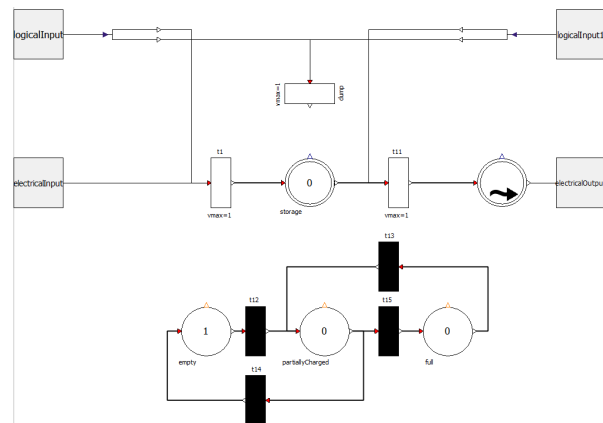
In principle, every PNlib place connected to test and inhibitor arcs can be used for these logical interfaces. Ad-

ditionally, a periodical clock and a module that converts a boolean expression to a logical input are implemented. Another handy component is a module called integer controller. This module can be used for incrementally increasing or reducing an integer number by activation the respective logical input. This can be used, for example, if the number of active wind turbines should be variable during simulation time. The application layer components that can be activated or deactivated have logical inputs as interfaces to realize this functionality. An example for such an activation or deactivation is the charging or discharging of a battery which can be both stopped and reactivated depending of the current power supply and demand.

Figure 5 shows an example for the usage of these control layer components. Here, an integer controller is controlled by a clock for decreasing its number and it is increased when place 2 has a token number of 1 and for two variables  $A$  and  $B$ , the requirement  $A > B$  is fulfilled.

In principle, controlling via the definition of rules can also be done with standard Modelica Boolean Blocks. However, in some cases it might be desirable to only use Petri nets in the model, e.g. for verification. Then it is helpful to use the components provides by PNRG.

### 3.2 Application Layer



**Figure 6.** The implementation of a battery. the upper interfaces are the logical inputs that control the charging and discharging. The energy storage is modelled as a place. The Petri net at the bottom is a inner control net that indicates the state of charging.

To model energy grids, typical components like power plants, storage systems or consumers are implemented with Petri nets. In the following, some basic implementations are presented.

In the most simplified way, components like photovoltaic (PV) power plants can be treated as an energy converter abstracted as a transition. The solar power  $P_m$  can be read in from a file and fed into a Petri net transition output. Using energetic PNs, a model that fulfills constraints of an energetic flow can be ensured and separated into components. The PV power plant can then be modeled as a transition with the input weight:

$$w_{in} = P_{in} \tag{4}$$

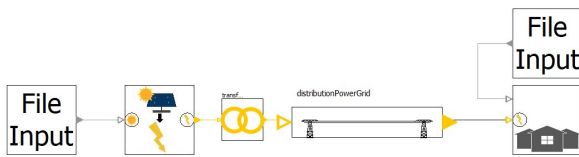
and the output weight:

$$w_{out} = P_{in} \cdot A \cdot \eta \tag{5}$$

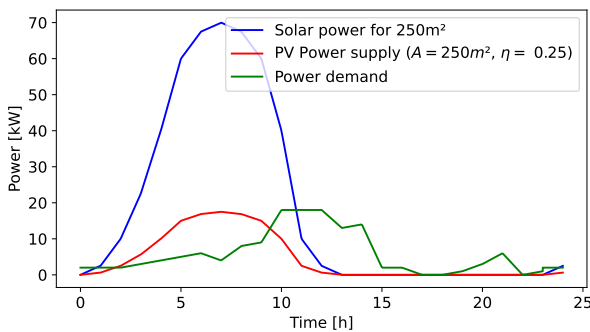
where  $A$  is the area of the PV panels and  $\eta$  is its efficiency. The angle between panel and sun or the temperature of the plant is neglected here but could be added in principle. The transitions output will then be connected to an energetic place. Afterwards consumers or storage systems could be connected to the energetic PN, where energy can be stored in normal continuous places.

With this implementation, the activation and deactivation of components can be achieved by blocking transitions with inhibitor and test arcs so that they are not able to forward energy. An example for a battery with charging and discharging that can be activated is shown in figure 6.

Simplifications that are applied are a constant charging and discharging power of batteries and the ignoring of energy losses during charging, discharging and storing of the energy. Also, power grids are assumed to have no energy losses. Wind turbines are idealized and do not include inertia effects.



**Figure 7.** Simple example of an energy grid modeled with PNRG.



**Figure 8.** The power of the sun for 100 m<sup>2</sup>, a PV power plant with 250 m<sup>2</sup> are and an efficiency of 0.25 as well as a consumer as shown in Figure 7.

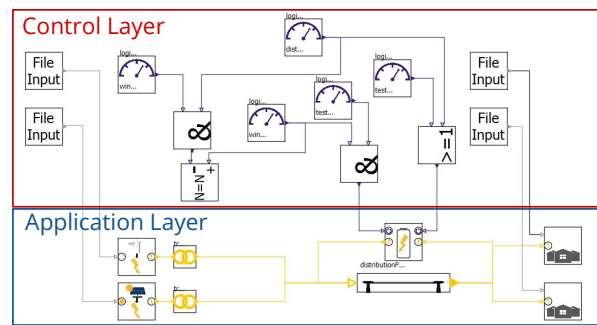
A simple example without VSS is shown in Figure 7. The grid consists of a PV power plant, a transformer, a distribution power grid and a consumer. The solar radiation power per area and the consumed power are read in from files. These values are artificially chosen but shall be used for demonstration purposes. The PV power plant has an area of  $A = 250\text{m}^2$  and an efficiency of  $\eta = 0.25$ . The

results are shown in Figure 8. It can be seen that the power of the PV plant follows the course of the solar power. Also it is clearly visible that the power supply exceeds the demand from  $t = 2\text{h}$  to  $t = 9\text{h}$  and can not fulfill it after  $t = 10\text{h}$ .

## 4 Modeling VSS Energy Grids with Energetic Petri Nets

To study how VSS can be modeled with PNRG, two examples were investigated. Both examples include conditional charging and discharging of a battery and a variable number of active wind turbines. The second example also contains instances of conditional connections.

### 4.1 Example 1 – Adaptive Activation of Wind Turbines and Battery Charging and Discharging



**Figure 9.** Example of an VSS energy grid modeled with PNRG in Modelica. The battery is charged only if enough power can be supplied from the wind and solar power plants. To control the difference between power supply and demand, additionally the number of active wind turbines can be controlled and the discharging of the battery can be activated.

The first example is shown in Figure 9. The grid contains wind power plants and PV power plants, a battery as a storage system and two consumers. The control layer is shown in Figure 9 marked with a red rectangle. It contains the file inputs and the rules that control the variability of the system. These rules are defined as following:

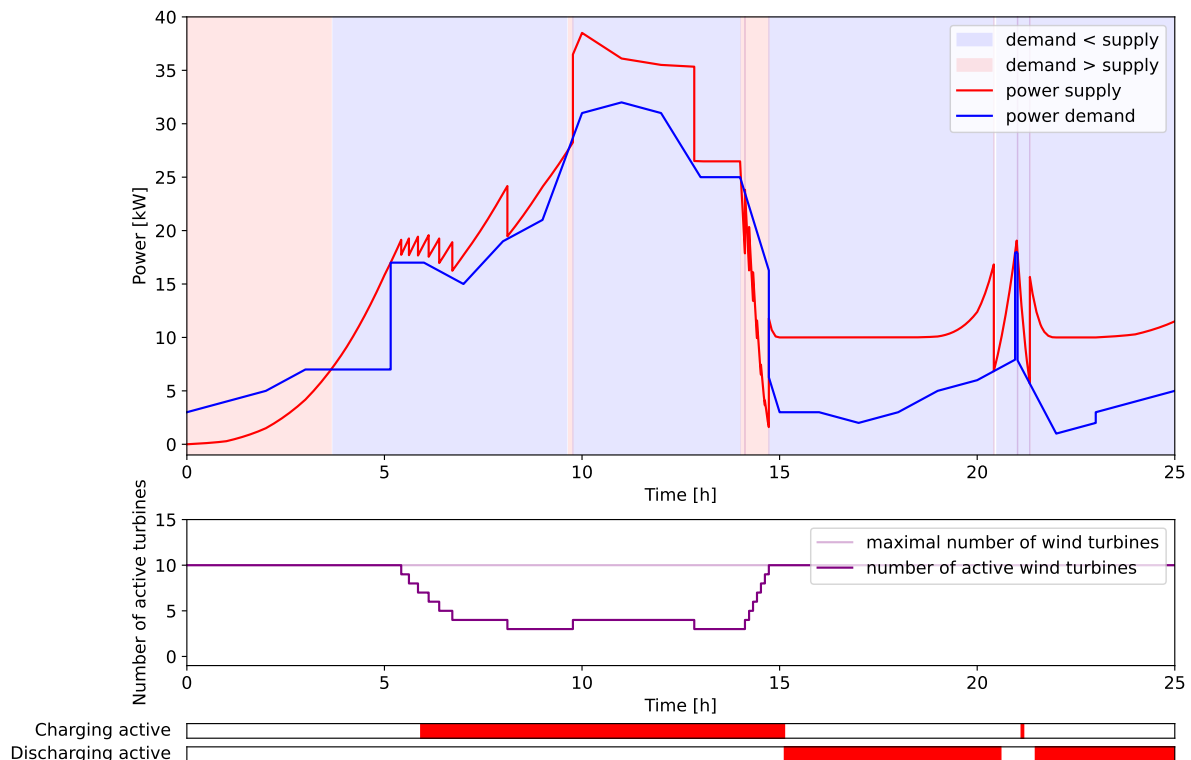
If the power supply is larger than the demand of the consumers, then:

1. Activate the battery charging if charging power would not overload the power supply.
2. If it is still too high, deactivate wind turbines.

If the power supply gets smaller than the demand of the consumers, then:

1. Reactivate wind turbines.
2. If all turbines are active, discharge the battery.

The number of active wind power plants is controlled with the integer controller. These rules should lead to a



**Figure 10.** The upper plot shows the power supply and demand for the simple VSS grid from Figure 9. The background is colored according to the difference between power supply and demand. The plot shows that the rules lead to the desired grid behavior and prove that VSS can be modeled in this case. The lower plot shows the number of active wind turbines. In the two bars at the bottom, the red color shows if the battery charging and discharging is active.

system, where it is tried to have an power supply larger than the consumption but the supply should not exceed the demand unnecessarily.

The resulting curves for power supply and demand are shown in Figure 10. The background is colored blue if the demand is smaller than the supply and it is red otherwise. The lower part in Figure 10 shows if the battery is charging or discharging indicated by a red bar. It also shows the number of active wind turbines.

Figure 10 shows that at the moment when the power supply covers the demand and the charging power of the battery at  $t = 5$ h, the battery start to charge. Since the power supply is still rising, the wind turbines are deactivated successively. At around  $t = 10$ h the power demand starts to rise above the supply so a wind turbine is reactivated but again deactivated at  $t = 13$ h when the power demand decreases. After about 14h the available wind and solar energy decreases significantly so despite reactivating all wind turbines step by step, the power demand can not be covered. So the battery charging is deactivated at  $t = 15$ h and since this is not enough the discharging of the battery starts. A short natural energy peak from  $t = 20$ h and  $t = 22$ h stops the discharging for a short period. At the beginning of the simulation, the discharging is inactive despite a shortage of the power supply because the battery is initialized empty and therefore cannot discharge.

In conclusion, the result confirms that the system be-

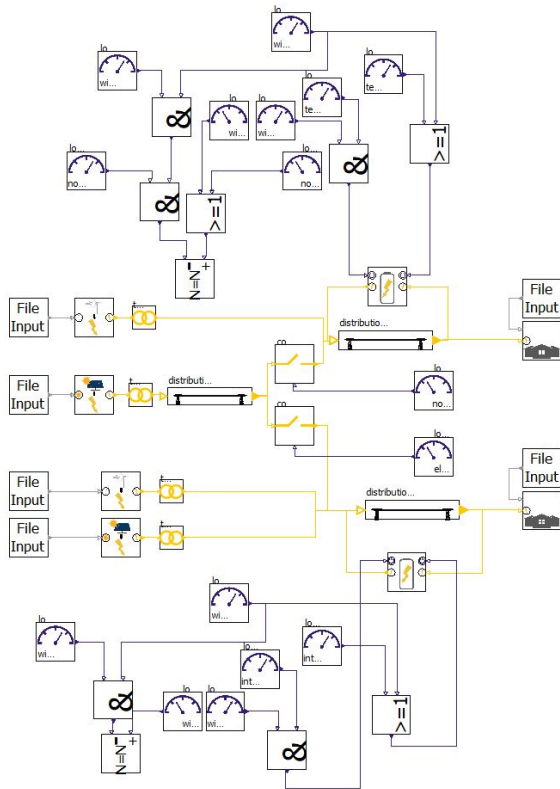
haves like desired and that the rules change the system accordingly. This also proves that VSS can be modeled using Petri nets for this case.

## 4.2 Example 2 – Conditional Connections

After example 1 showed that PNs can model VSS in Modelica, a more complex model shall be investigated now. As shown in Figure 11 the system consists of two grids. Both of them are similar to the model presented in example 1 but only one consumer is connected to each of the grids. Also, the number of wind turbines and the PV area is different. However, the main difference is that one of the PV power plants is not statically connected to the upper distribution grid but dynamically connectable to one of both grids during run time.

The implemented rules for the batteries and the number of active wind power plants is basically the same as in example 1. Some modifications are done to avoid conflicts for the different state of the conditional connection. The upper PV power plant will always be connected to the distribution grid, with the smaller difference between supply and demand. In this difference, the power supply of the PV power plant itself is not included.

In Figure 12 the results of the simulation are shown. As expected the PV power plant will be connected to the grid with a smaller difference between supply and demand. While the supply can not cover the demand for both grids, the PV power plant is reconnected from the lower to the



**Figure 11.** Example for a more complex VSS energy grid. A conditional connection can connect a solar energy park to either the upper or the lower distribution grid. The charging and discharging of the battery as well as the controlling of the number of active wind turbines is controlled like in example 1 but adjusted to work with the conditional connection.

upper grid at  $t = 1$  h because at this moment the lack of power is higher. The actual positive affects can be seen from  $t = 4$  h where for the upper grid the supply can cover the demand by itself and the lower grid which is connected to the PV power plant can only cover its demand because of the connection. Here the connection even leads to an activation of the battery charging.

It should be noted, that a clear definition of rules is very important for the success of such simulations. Inconsistencies will lead to crashes of the simulation. Especially for complex systems where multiple components affect each others activation and deactivation rules, it can get very difficult to define these rules sufficiently well.

## 5 Conclusion

This paper presented an approach to model VSS energy grids in Modelica based on Petri nets. While the implementation is still experimental and the technical components are simplified, it was shown that Petri nets are suitable for modeling systems with a variable structure that changes during simulation time in Modelica. To demonstrate this, two examples for energy grids with different VSS types were presented. These two examples showed that different VSS can be modeled with PNs. Both mod-

els behaved like expected due to the defined rules for state transitions.

To ensure that the used Petri nets keep properties of the flow of energy while enabling component-based models, so called energetic places and transitions were defined. Based on the PNlib Modelica library, these special Petri nets were implemented to create a library called PNRG that can be used to model VSS energy grids.

However, the capabilities of modeling VSS are limited. While activating or deactivating components is possible, adding variables, objects or components dynamically in run time is currently not possible.

## 6 Outlook

Although the presented examples showed that Petri nets can be used to model VSS, a lot of work has to be done for excellent energy grid modelling using Petri nets. In the first place, the realism of the implementations for different components should be increased. For example, additional effects influencing the charging and discharging powers of the batteries could be implemented. One approach could be to use existing models and libraries for these detailed calculations. Also, more parameters should be implemented to increase the realism of the modeling. Implementing parameters like voltages, phase angles and frequencies are needed to verify how stable the simulated energy grids under state transitions are.

As described in the introduction, one of the motivations for studying VSS is to simulate sector coupling systems. Therefore a logical next step would be to implement and study components like electrolyzers, gas power plants or gas and heat consumers.

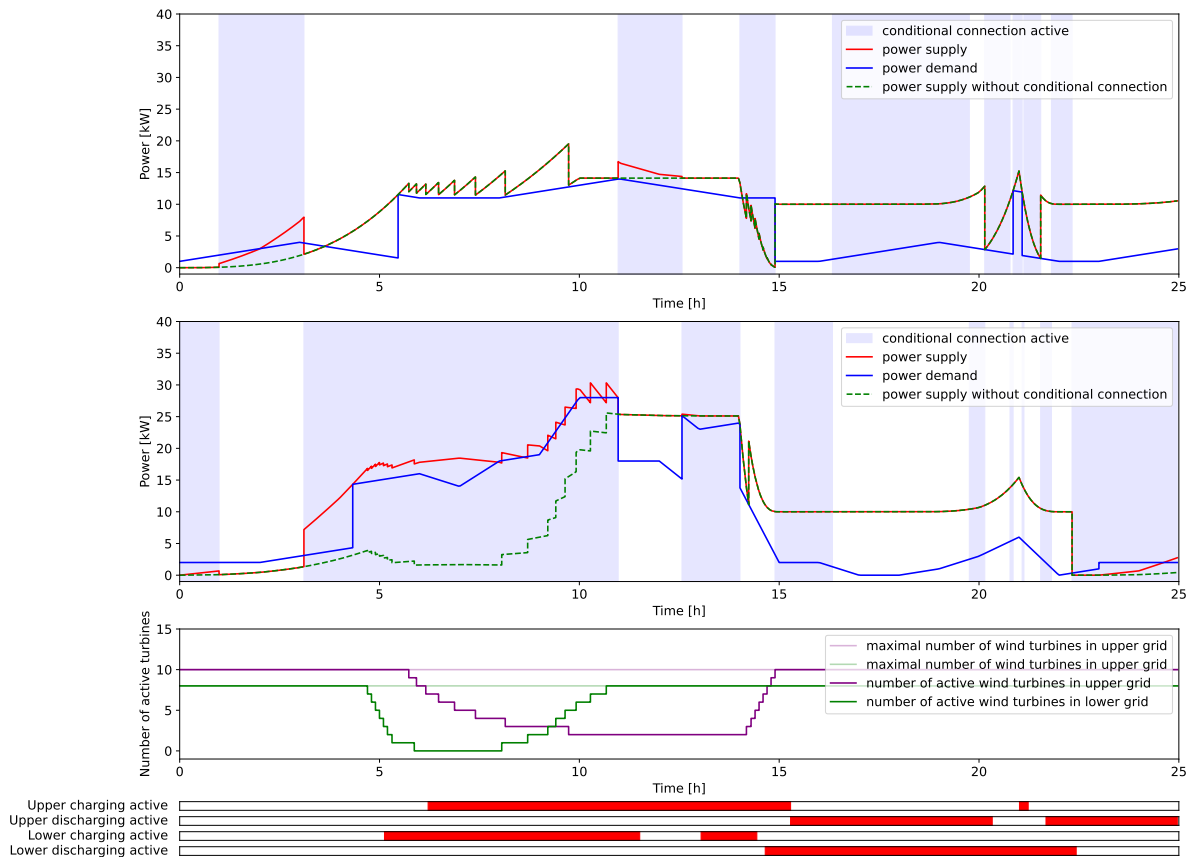
Also an advanced formalism to define the state transitions to avoid conflicts is an interesting subject for further investigations. Especially for the control sub-net, verification methods that are defined for Petri Nets could be studied to avoid conflicts. (Svadova and Hanzálek 2004) presents a method for generating reachability graphs for hybrid Petri nets that is also applied in (Lu et al. 2016) for the analysis of the micro grid models. Therefore, this might be one approach for an analysis tool for the here described grids.

In the end, the results from this approach need to be verified with other energy grid simulation models, which are not based on Petri nets. This comparison is important, not only to evaluate how realistic the single components work but also to validate the realism of the whole grid.

## Acknowledgements

The authors would like to thank the Boysen–TU Dresden–Research Training Group for the financial and general support that has made this contribution possible. The Research Training Group is co-financed by the Friedrich and Elisabeth Boysen Foundation and the TU Dresden. The work is also partially supported by the German Research Foundation (SFB/TRR 339, Project ID453596084).





**Figure 12.** Simulation output for the second example. The upper plot shows the power supply and demand for the upper distribution grid from Figure 11, the second plot for the lower grid. The background is colored according to if the PV power plant is connected to the respective grid. The lower plot shows the number of active wind turbines. In the four bars at the bottom, the red color shows if the battery charging and discharging is active.

## References

- Agerwala, Tilak (1974). *Complete Model for Representing the Coordination of Asynchronous Processes*. The Johns Hopkins University, Baltimore, Maryland.
- Alhamali, Abdulwahab et al. (2016-12). “Review of Energy Storage Systems in electric grid and their potential in distribution networks”. In: pp. 546–551. DOI: 10.1109/MEPCON.2016.7836945.
- Benveniste, Albert et al. (2019-10). “Multi-Mode DAE Models - Challenges, Theory and Implementation”. In: *Computing and Software Science: State of the Art and Perspectives*. Vol. LNCS-10000. Lecture Notes in Computer Science. Springer, pp. 283–310. DOI: 10.1007/978-3-319-91908-9\_16. URL: <https://inria.hal.science/hal-02333603>.
- David, René and Hassane Alla (1987). “Continuous Petri nets”. In: *8th European Workshop on Application and Theory of Petri Nets, Zaragoza*, pp. 275–294.
- David, René and Hassane Alla (2008-07). “Discrete, Continuous, and Hybrid Petri Nets”. In: *Control Systems Magazine, IEEE* 28, pp. 81–84. DOI: 10.1109/MCS.2008.920445.
- Esperon, Daniel, Alexandra Mehlhase, and Thomas Karbe (2015-07). “Appending Variable-Structure to Modelica Models (WIP)”. In: *Proceedings of the Conference on Summer Computer Simulation 2015*.
- Fridgen, Gilbert et al. (2020). “A holistic view on sector coupling”. In: *Energy Policy* 147, p. 111913. DOI: <https://doi.org/10.1016/j.enpol.2020.111913>. URL: <https://www.sciencedirect.com/science/article/pii/S0301421520306248>.
- Heckel, Jan-Peter et al. (2022). *Investigation of dynamic interactions in integrated energy systems*. DOI: 10.15480/882.4678. URL: <http://hdl.handle.net/11420/13866>.
- Liu, Tao et al. (2022-05). “Stability and Control of Power Grids”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 5. DOI: 10.1146/annurev-control-042820-011148.
- Lu, Xiaoyu et al. (2016). “Hybrid Petri nets for modeling and analysis of microgrid systems”. In: *IEEE/CAA Journal of Automatica Sinica* 3.4, pp. 349–356. DOI: 10.1109/JAS.2016.7510070.
- Mai, Carl et al. (2018-02). “Adaptive Petri Nets -A Petri Net Extension for Reconfigurable Structures”. In: *Proceedings of the Tenth International Conference on Adaptive, Self-Adaptive Systems, and Applications*.
- Mehlhase, Alexandra (2015-06). “Konzepte für die Modellierung und Simulation strukturvariabler Modelle”. PhD thesis. DOI: 10.13140/RG.2.1.3041.2007.
- Murata, Tadao (1989). “Petri nets: Properties, analysis and applications”. In: *Proceedings of the IEEE* 77.4, pp. 541–580. DOI: 10.1109/5.24143.
- Nytsch-Geusen, Christoph et al. (2005-04). “MOSILAB: Development of a Modelica based generic simulation tool supporting model structural dynamics”. In: *Proceedings of the 4th International Modelica Conference, Hamburg, Germany*.

- Petri, Carl Adam (1962). *Kommunikation mit Automaten*. Technische Hochschule Darmstadt. URL: <https://edoc.sub.uni-hamburg.de/informatik/volltexte/2011/160/>.
- Proß, Sabrina and Bernhard Bachmann (2012-11). “PNlib - An Advanced Petri Net Library for Hybrid Process Modeling”. In: Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany, pp. 47–56. DOI: 10.3384/ecp1207647.
- Senkel, Anne et al. (2021). “Status of the TransiEnt Library : transient simulation of complex integrated energy systems”. In: *14th International Modelica Conference 2021*. Linköping Electronic Conference Proceedings, pp. 187–196. URL: <http://hdl.handle.net/11420/11525>.
- Svadova, Martina and Zdeněk Hanzálek (2004-11). “An algorithm for the evolution graph of extended hybrid Petri nets”. In: vol. 5, 4905–4910 vol.5. ISBN: 0-7803-8566-7. DOI: 10.1109/ICSMC.2004.1401308.
- Tinnerholm, John (2022). *A Composable and Extensible Environment for Equation-based Modeling and Simulation of Variable Structured Systems in Modelica*. Linköping University Linköping University, Software and Systems, Faculty of Science Engineering. Licentiate Thesis. DOI: 10.3384/9789179293680.
- Tinnerholm, John, Adrian Pop, and Martin Sjölund (2022). “A Modular, Extensible, and Modelica-Standard-Compliant OpenModelica Compiler Framework in Julia Supporting Structural Variability”. In: *Electronics* 11.11. ISSN: 2079-9292. DOI: 10.3390/electronics11111772. URL: <https://www.mdpi.com/2079-9292/11/11/1772>.
- Tinnerholm, John, Adrian Pop, Martin Sjölund, et al. (2020-11). “Towards an Open-Source Modelica Compiler in Julia”. In: pp. 143–151. DOI: 10.3384/ecp2020174143.
- Utkin, Vadim I. (1977). “Variable structure systems with sliding modes”. In: *IEEE Transactions on Automatic Control* 22.2, pp. 212–222. DOI: 10.1109/TAC.1977.1101446.
- Valk, Rüdiger (1978). “Self-modifying nets, a natural extension of Petri nets”. In: *Automata, Languages and Programming*. Ed. by Giorgio Ausiello and Corrado Böhm. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 464–476. ISBN: 978-3-540-35807-7.
- Wolf, Karsten (2019). “How Petri Net Theory Serves Petri Net Model Checking: A Survey”. In: *Transactions on Petri Nets and Other Models of Concurrency XIV*. Ed. by Maciej Koutny, Lucia Pomello, and Lars Michael Kristensen. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 36–63. ISBN: 978-3-662-60651-3. DOI: 10.1007/978-3-662-60651-3\_2. URL: [https://doi.org/10.1007/978-3-662-60651-3\\_2](https://doi.org/10.1007/978-3-662-60651-3_2).
- Zimmer, Dirk (2010). *Equation-based modeling of variable-structure systems*. Swiss Federal Institute of Technology, Zürich. URL: <https://doi.org/10.3929/ethz-a-006053740>.