TOWARDS FORMAL VERIFICATION OF SMART GRID APPLYING IN VIETNAM: SOME FIRST EXPERIMENTATION

Thang Bui [(1)], *Liem Nguyen [(1)], *Anh D. Nguyen [(1)], Tuyen D. Nguyen [(1)], *Tuan Bui [(1)], *Huan M. Luong [(2)]

[(1)] Ho Chi Minh City University of Technology - Vietnam National University-Ho Chi Minh City, 268 Ly Thuong Kiet St., Dist. 10, Ho Chi Minh City, Vietnam

[(2)] Sai Gon University, Dist. 5, Ho Chi Minh City, Vietnam

{bhthang, nvliem, duyanhnguyen, ndtuyen, tuanbc88}@hcmut.edu.vn; huanlm@sgu.edu.vn.

ABSTRACT

Nowadays, smart grids are used widely around the world when they enable detecting, reacting and pro-acting to changes in usage and multiple issues of the electricity system and more interestingly, they have self-healing capabilities. In Vietnam, some smart grids have been built and operated recently. To ensure the effectiveness of the smart grids, the correctness of the system designs must be studied carefully before the explosion of the use of smart grids, especially in developing countries like Vietnam.

In this research, we proposed a new approach to represent smart grids using Colored Petri Net (CPN). A verification method has also been developed including an approach of smart grid representation using Colored Petri nets. Then, some experimentation has been conducted to show the usefulness of the proposed method.

Keywords: *smart grid verification, smart grid testbed, formal method.*

1. INTRODUCTION

Smart Grid [1] is a grid power system using information and communication technology to optimize the transmission and distribution of electricity between producers and consumers and consolidate the electricity infrastructure with the inter-information infrastructure. The grid has the basic functions of: (1) resisting intentional attacks against the system both physically and in the computer network; (2) reduce the amount of energy consumed on the wire, improve the quality of electricity; (3) reducing production and transmission costs, and upgrading costs by differentiating electricity consumption; and (4) capable of self-healing in the event of a power failure. Smart grid has begun to be deployed in Vietnam during this decade and is still being deployed [2].

Smart grid studies have been implemented for a long time (see also UCLA Smart Grid Energy Research Center - http://smartgrid.ucla.edu/) and spans from equipment studies, systems, communications, optimization, network security, ... These studies, before being deployed in practice, need simulation and testing steps for the correctness confirmation. Although formal verification is the most noticeable method for verifying the correctness, it requires that the system has to be specified formally and it suffers from the well-known state space explosion problem in practice.

In general, a formal specification is a specification that represents a formal language (with the lexicon, syntax and semantics of that language formally defined). [3]. In this way, a formalized system is an abstraction of the real system [4] to focus on the representation of what the system does. At that time, the use of mathematically based formal verification methods will either prove the correctness of the whole or show the irrationality of a given system by its formal specification and attribute to be proved [5].

Specifically for smart grids, recent studies have shown that properties like load balancing or the probability of a fault occurring in the grid (and fault resilience) can possibly be tested by the formal methods [6], [7], [8]. However, these studies also show that the effectiveness of formal verification (test time, computer resources used, and scalability) depends on how the system is modeled and formal verification techniques.

Research on formal methods focuses on two main branches, formal specification, and formal verification [5]. The specification methods have many research directions such as history-based, state-based, transition-based, functional, special operational or higher-order functions [4]. Typical contributions to this branch are time-logical representations [9], representation languages such as PROMELA [10], model languages such as Petri nets. In general, studies on formal specifications attempt to construct representation languages / methods that represent a system's characteristics.

In the second direction of research, researchers focus on modeling the system and modeling these models. This branch consists of two main approaches: theorem proving and model checking [5]. While the theorem proving is based on inference laws to prove the correctness of the system, the model checking is based on search algorithms to find counterexamples of the system's improperness. Currently, there are many powerful tools to help scientists continue to implement new tests, as well as help users test their practical systems. Some works such as SPIN model test tool (http://spinroot.com), NuSMV model test tool (http://nusmv.fbk.eu), model test tool probability PRISM (https://www.prismmodelchecker.org), a proof of Isabelle's theorem (http://isabelle.in.tum.de).

The specific studies on formal methods in the field of smart grid can include the two latest research branches [6], [7] and [8]. The first branch of research [6], [7] proposes to apply the NuSMV tool in testing some properties such as load balance, resilience of the mesh system. Research results (using NuSMV) show the feasibility of applying model checking tools in smart grid verification. However, it also shows an inability to test more complex properties due to limited computing resources (memory). Another possible reason is that the modeling approach presented by the study is still too complex to test form. In addition, the study has not shown ways to use more scalable studies such as abstraction or / and random path [11].

The second branch of research [8] focuses on fault probability and system recovery probability in smart grids using both wired and wireless communication. Similar to the study above, this work also models the system using a model testing tool called PRISM. However, the model used to illustrate the project is still quite small and the feature to be tested is still simple.

The studies on modeling and testing the system model have been studied by the authors Le Ngoc Kim Khanh, Bui Hoai Thang and Quan Thanh Tho [12][13][14][15][16][17] - [18]. These studies focus on modeling wireless sensor networks and using abstraction techniques, clustering, etc., applying model testing algorithms to verify congestion characteristics of sensor networks. wireless. In addition, the congestion probability is also taken into account when applied in practice with unstable network parameters.

In addition, the research on improving the effectiveness of model checking has also been studied by author Bui Hoai Thang and colleagues [19], [20], [21], [22], [11] with abstract methods, finding heuristic search, randomization, parallelization, indexing, ...

Studies on smart grid verification in Vietnam seemingly have not been published.

This research contributes the following three-folds: (1) smart grid representation using Color Petri nets; (2) a smart grid case study consisting of a set of sample smart grids of three types: many power sources, many consumers, unstable power sources; (3) Some verification experimentation conducted on the case study to show the usefulness of the proposed method.

The rest of this paper is organized as follows. In the next section, background information is presented. The proposed formal representation approach for smart grids is then presented in section 3. The testbed and some experimental results are presented in section 4. The last section is for conclusion and future works.

2. BACKGROUND

2.1 Smart grids

As defined by smartgrid.gov, the smart Grid is a developing network of new technologies, equipment, and controls working together to respond immediately to our 21st century demand for electricity. The Smart Grid represents an unprecedented opportunity to move the energy industry into a new era of Reliability, Availability, and Efficiency that will contribute to our Economic and Environmental health.

During operation, the smart grid can be listed by level such as: the customer, the distribution system, and the transmission system. At the level of the customer, it may include some things like meters that can be read automatically, meters that communicate to customers, control of customers' loads, and flexibility in the use of time-of-day or time-of-use meters. At the level of the distribution system may include distribution system automation, selective load control, managing distributed generation and "islanding". And the level of the transmission system may include measurement of phase and other advanced measurements, and other advanced control devices, distributed and autonomous control. [23]

In summary, the goal of smart grid designing is to provide grid observability, create controllability of assets, enhance power system performance and security, and reduce costs of operations, maintenance, and system planning.

In the research of smart grid, there are a lot of computational tools for modeling and analysis of Power Generation and Transmission Systems of the Smart Grid. [2]

2.2 Formal verification

Formal verification is a research direction aimed to prove the correctness of a system with respect to a certain property represented in a formal specification. Theoretically, the verification is a process of finding a formal proof on a formal/mathematical model of the system under test. There are two main approaches in formal verification: model checking and theorem proving [Formal Methods: State of the Art and Future Directions].

In model checking, a finite model of a system is used for searching evidence of violation of the desired property. The system is verified when no such violation evidence exists or a counter-example will be shown. It is a very practical approach, even though in the explosion of the state space, the search (for violation) can stop within a limit of computer resources and running time and return the correctness with some confidence level [Probabilistic congestion of wireless sensor networks: a coloured petri net based approach]. The well-known logic used in model checking is temporal logic, which aims to capture the temporal aspect of the property. For example, one can check if a smart grid satisfies the property “Definitely, the top priority consumer will receive enough energy as required”. Although, in theory, the search in the state space is exhaustive, there are many researches in the model checking field to overcome this drawback [Abstraction-Guided Model Checking Using Symbolic IDA* and Heuristic Synthesis], [Formal verification based on guided random walks] [A bitwise-based indexing and heuristic-driven on-the-fly approach for Web service composition and verification] [Smaller to sharper: efficient web service composition and verification using on-the-fly model checking and logic-based clustering] [Congestion verification on abstracted wireless sensor networks with the WSN-PN tool].

In theorem proving, both the system and its desired properties are expressed as formulas in some mathematical logic, which is based on axioms and inference rules. By that logic, it can deal directly with infinite state space, while model checking can work only with finite state space. Some of the theorem techniques are used in practice with excellent results [Web Service Composition Automation Based on Term Rewriting System].

2.3 Colored Petri Nets

A Petri net is a weighted, directed graph, in which the nodes are separated into places and transitions, to model systems. Directed arcs join places to transitions (as inputs) and transitions to places (as outputs). The Petri net works by transmitting tokens from (input) places to transitions and from transitions to (output) places (firing). The amount of tokens transmitting is defined by the weight of directed arcs. Colored Petri net (CPN) is an extension of Petri net which combines the strengths of Petri nets with the expressive power of functional programming languages by adding more features and properties to tokens, places, and transitions resulting in more classes of high level. It allows tokens to have a data value attached to them called the token color, and each place represents a color set. Moreover, arc-expressions (an extended version of arc weights in classical Petri nets) specify which tokens can flow over the arcs. Guards that are Boolean expressions defining additional constraints on enabling transitions.

A CPN is a tuple $CPN = (C; B; P; T; V; F)$, where

- C is a finite set of color sets.
- B is a bag of tokens (value) of colors $c \in C$.
- P is a finite set of places.
- T is a finite set of transitions.
- V is a finite set of variable $v \in V$.
- F is a finite set of functions.

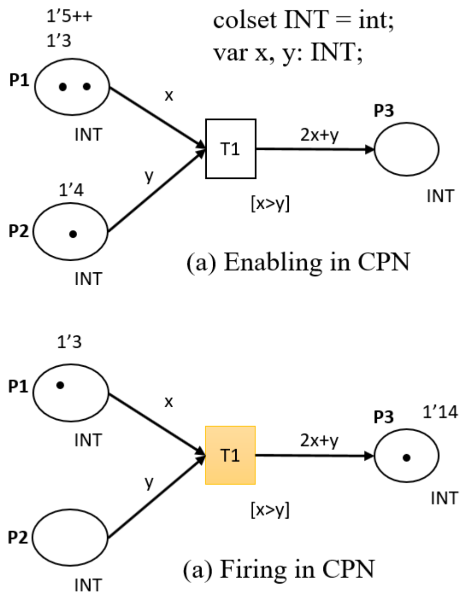

Fig. 1: Colored Petri Net (CPN)

For example, Figure 1 shows a CPN in its two stages: enabling and firing. In Figure 1(a), place P1 contains 2 bags of integer tokens (one with 5 tokens and another with 3 tokens) and place P2 contains a bag of 4 integer tokens. They are the input places for transition T1 with the guard expression x>y (from P1 and P2, respectively) as the condition to enable the transition. In Figure 1(b), the transition T1 has been fired when the guard expression is confirmed (the bag of 5 tokens in P1 and the bag of 4 tokens in P2). It took (those) tokens from place P1 and P2 and generated tokens in P3 (a bag with 14 integer tokens as the result of the output arc-expression 2x+y).

3. SMART GRID REPRESENTATION

In this paper, we use a smart grid whose topology depicted in Figure 2, as a demonstration case study 01.

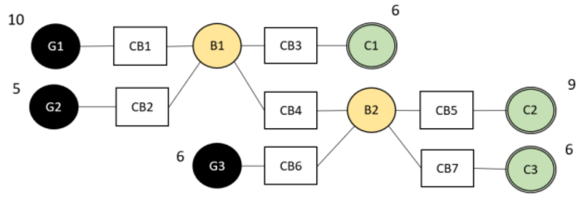

Fig. 2: Topology of sample smart grid. A filled circle denotes a generator, a hollow circle denotes a bus, a double-lined circle denotes a consumer (or load) and a square denotes a circuit breaker (CB).

The smart grid in Figure 2 consists of three generators, three consumers, two buses and seven circuit breakers. A generator can either be a normal generator or a smart generator. These elements will be called nodes in the grid.

In this paper, we assume that these generators are normal generators that can generate stable power.

Generator G1 can generate 10MW, G2 can generate 5MW and G3 can generate 6MW. Three consumers require 6MW, 9MW and 6MW respectively. The total amount of power produced can satisfy the total amount of power required.

Table 1. The parameter type of the Grid in Figure 2.

	Generator	Consumer
Id	Number	Number
Type	Enum	Enum
Capacity	Number	Number

We also assume that all elements in a smart grid have standard parameters such as id and type. Moreover, some of them such as generators and consumers have additional parameters such as capacity for their generating capacity or consuming capacity. A configuration of the grid in Figure 2 is described in Table 2.

Table 2. The parameter configuration of the Grid in Figure 2

	Capacity
Generator 1	10
Generator 2	5
Generator 3	6
Consumer 1	6
Consumer 2	9
Consumer 3	6
Generated	0

3.1. Smart grid representation using Color Petri nets

Based on characteristics of a Smart Grid, we proposed that the grid can be represented using two generic components: generation and transmission/ consumption. Each of these components is modeled by a CPN. An illustration of those representations is in Figure 3. Figure 3(a) is a pure petri net created from the topology in figure 2 by reducing the busses and CBs. The left part of the petri net describes the power generation with three Generators G1, G2 and G3. From these three generators, power is generated through transitions "gen1", "gen2", "gen3" and is stored in the place "Generated". After power is generated, power will be transferred and consumed in place "Consumer 1", "Consumer 2" and "Consumer 3" through transitions "trans1", "trans2" and "trans3", respectively.

We then leverage the power of the Color Petri net to create a simpler net as shown in Figure 3(b). By folding places and transitions with similar properties, and defining the functions "fn_gen", "fn_trans", "fn_cons" in Listing 3 to describe power generation, transmission and consumption. Figures 4 and 5 show the details of two folded components, which will be explained in more detail later.

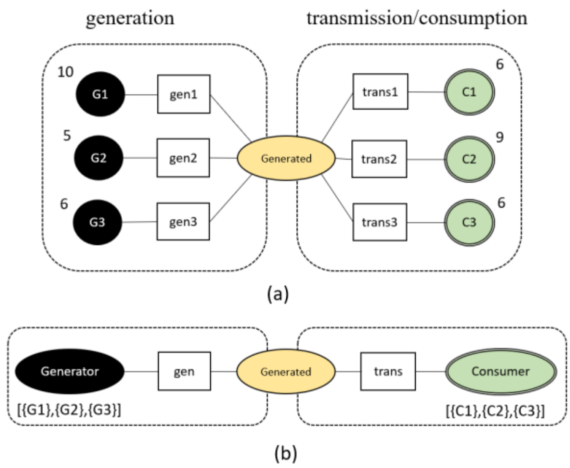

Fig. 3: Two components: generation and transmission (a); folded generation and folded transmission (b)

The declarations are presented using the CPN-ML syntax [25] as follows.

Color set types:

+ "IDX": integer, a unique id of a node in the grid.
+ "TYPE": enumeration = ["GEN", "CON"] for Generator, and Consumer (Load).
+ "CAPACITY": integer, the capacity of a generator or consuming capacity of a consumer.
+ "NODE": a record of IDX *TYPE *CAPACITY, a node in the grid.
+ "POWER": an amount of power.
+ "NODE_POWER": a product of "NODE" and "POWER", for the amount of power generated or consumed by the NODE.

A declaration of the color sets which are used in the grid is described in Listing 1.

Listing 1: Declarations of the color sets.

```
colset IDX = int;
colset TYPE = with GEN|CON;
colset CAPACITY = int;
colset POWER = int;

colset NODE = record i:IDX*t:TYPE*c:CAPACITY;
colset NODE_POWER = product NODE*POWER;
```

Configuration:

The configuration of the grid described in Table 1 and 2 is in Listing 2. It is also called the initial marking of the CPN.

Listing 2: Initial marking of the "Generator" place and the "Consumer" place

```
val initGenerator: NODE_POWER =
[({i=1,t=GEN,c=10},10)
,({i=2,t=GEN,c=5},5)
,({i=3,t=GEN,c=6},6)]
;
val initConsumer: NODE_POWER =
[({i=4,t=CON,c=6},0)
,({i=5,t=CON,c=9},0)
,({i=6,t=CON,c=6},0)]
;
val initGenerated: POWER = 1`0;
```

Listing 2 shows the initial marking of the "Generator", "Generated" and "Consumer" place. Generators are initialized with full energy production capacity. Generated place contains zero power at the initial stage.

Listing 3: Three functions are used in the CPN

```
fun fn_gen(p1: POWER,p2: POWER) = p1+ p2;
fun fn_cons(n: NODE, p: POWER) =(n,p+(#c(n)));
fun fn_trans(n:NODE, p:POWER) = p-(#c(n));
```

3.2 Smart grid component representation

In this section, we describe more about the components mentioned above. The generation component has two places “Generators”, “Powers” and a transition “gen”. The Generators place contains all generators of the smart grid. For example, the Generator place in Figure 4 contains three generators G1, G2 and G3 of the smart grid in Figure 2, initial marking of the "Consumer" place followed by List 2. The pattern "({i=1,t=GEN,c=10},8)" represents Generator G1 has id = 1 with a maximum power generation capacity of 10. The G1 is being set up to generate power is 8 currently (80% of maximum capacity) . With an approach like this, we can easily simulate a stable or unstable power source as well as on/off the power source..

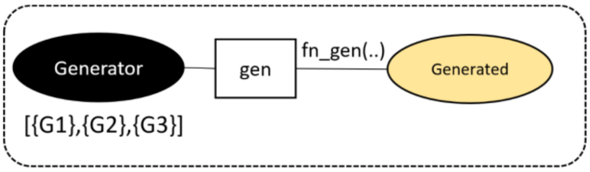

Fig. 4: The folded generation component.

In this component, the “gen” transition fires when taking one NODE_POWER token from the place “Generator”, this color token represents a generator and its associated power generation. And a POWER token from the place “Generated”, which represents the total amount of electricity that has been produced and stored. The function “fn_gen” will sum the amount of electricity available from the generator and the amount of electricity available in the “Generated” place to create a new total.

Figure 5 depicts the state of the grid implemented on the CPN Tool after the transition “gen” fires once upon receiving G2 tokens ({i=2,t=GEN, c=5}, 5) and 1`0 token power to generate 1`5 token power.

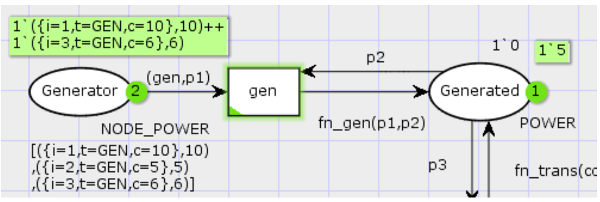

Fig. 5: The implementation of the generation component on CPN tool.

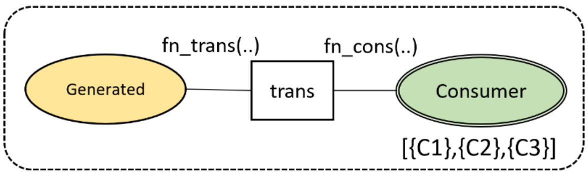

Fig. 6: The folded transmission component.

After successfully generated, the electric power is ready to be transmitted and consumed using the transmission component. There are two places in this component: “Generated” and “Consumer”. The “Consumer” place in Figure 6 contains all consumers of the grid, initial marking of the "Consumer" place followed by List 2. For example, ({i=4,t=CON,c=6}, 0) represents the Consumer 2 with id 4, the demand for power is 6, and the amount of power received is 0.

Transition “trans” fires when it receives one token from place “Generated” and one from place “Consumer”. The function “fn_trans” and “fun_cons” are used for fransfering and consuming power between two places “Generated” and “Consumer”

Figure 7 describes a state of the grid when the place “Consumer 1” received token "1`6" while the remaining token "1`9" at place “Generated”.

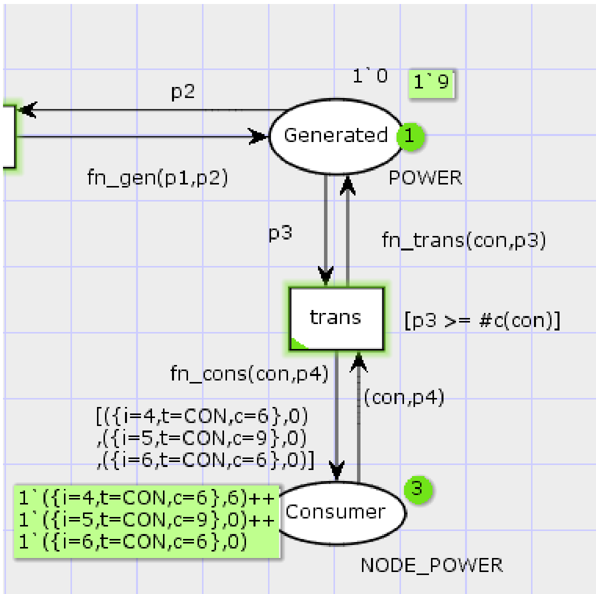

Fig. 7: The implementation of the transmission component on CPN tool.

The full model of the demonstration smart grid is illustrated in Figure 7. It can be seen that no matter what the complexity of the smart grid, the presentation model in CPN is small.

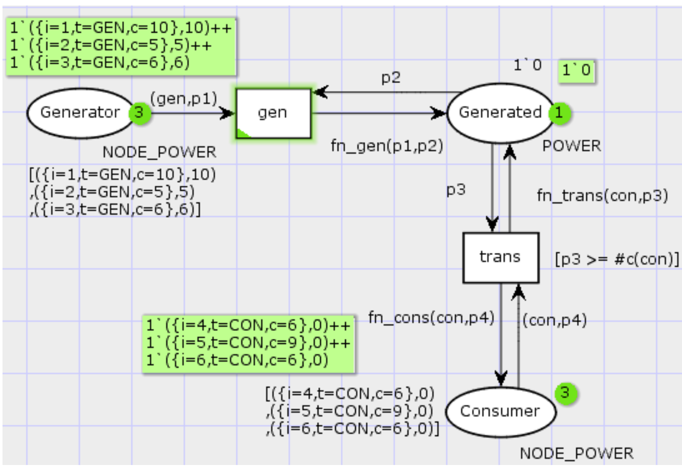

Fig. 8: The full model on CPN tool.

3.3 State Space

The state space in CPN is a set of possible configurations of system can have. In this paper, it is a set of possible states of the grid in figure 2. So, generating state space in CPN model is an important task. We use the integrated State Space module in CPN to analyze state space and Graphviz library in figure 9 for exporting state space to a dot file, then we create a graph from the dot file.

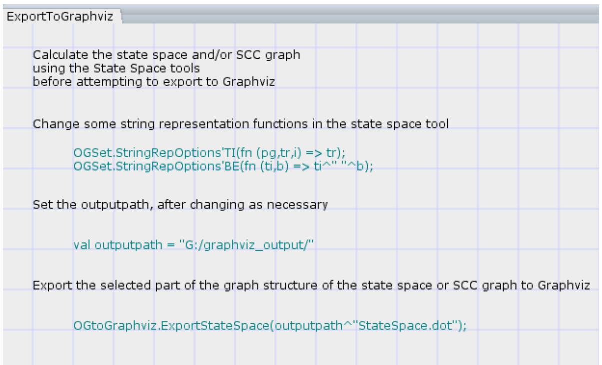

Fig. 9: The Graphviz code for exporting state space to dot file.

Figure 9 represents a block of code in the CPN tool for exporting state space to the “StateSapce.dot” file.

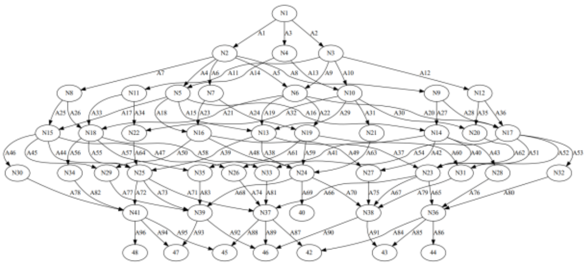

Fig. 10: A full state space.

Listing 4: a part of State space report and statespace.dot file

```
CPN Tools state space report for:
StateSpace report:
 State Space
     Nodes:  48
     Arcs:   96
     Secs:   0
     Status: Full
…

StateSpace.dot file:
digraph cpn_tools_graph {
  N40 [label="40:
sgrid'Generated 1: 1`3
sgrid'Generator 1: empty
sgrid'Consumer 1: 1`({i=4,t=CON,c=6},0)++
1`({i=5,t=CON,c=9},18)++
1`({i=6,t=CON,c=6},0)
"];
  N42 [label="42:
…
N1 -> N4 [ label="A3:
1->4:gen {p2=0,gen={i=2,t=GEN,c=5},p1=5}" ];
  N1 -> N3 [ label="A2:
1->3:gen {p2=0,gen={i=3,t=GEN,c=6},p1=6}" ];
  N1 -> N2 [ label="A1:
1->2:gen {p2=0,gen={i=1,t=GEN,c=10},p1=10}" ];
…
```

Listing 4 consists of a part of state space report and a part of StateSpace.dot file.

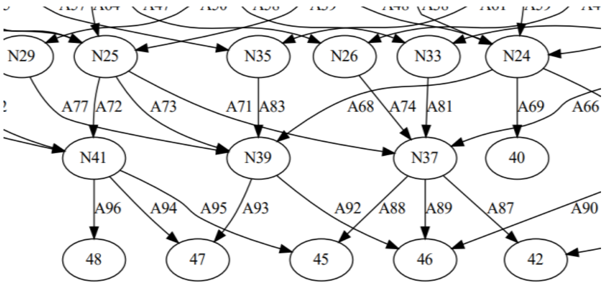

Fig. 11: a part of full state space.

Figure 11 shows a part of full state space of figure 8. Nodes from 40 to 48 describe a final state of the grid. For example, Node 40 in Listing 4 shows a result when Consumer 1 2 and 3 receive 0, 18 and 0 powers respectively and 1`3 redundant power tokens at Place "Generated"

3.4 Discussion

In previous studies by other authors [26], all generators are defined as places in a Petri Net, and try to represent the net as similar as possible to the network topology. It is good for engineers to simulate the net as they are familiar with the topology. However, for large or larger networks, the Petri nets are too complicated to be represented and simulated and are too hard to be upgraded. In our proposed approach, all generators (and their parameters) are color tokens, and the whole topology is reduced to a few simple components with flexible configuration. For example, the net in [26] with 11 generator nodes (P0 to P11) can be represented using our approach with only one node (Node P using color token [{i=0,p=1}, {i=1,p=1, {i=2,p=4}] with i is name of node, p means power)as illustrated in Figure 12.

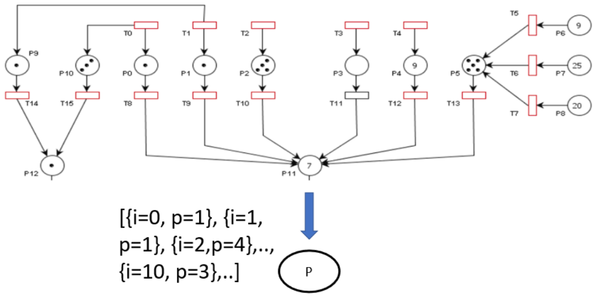

Fig. 12: the representation of the net in [26] using our proposed approach (the top half of the figure is a part of the net in [26], and the bottom of the figure is our proposed approach)

Therefore, our nets are easy to be updated, to be re-configured. Tools to convert from a smart grid to our net will be provided.

4. EXPERIMENTATION

4.1 Experimental setup

In this section, we try to demonstrate how to represent more practical mode of a smart grid using our proposed approach. Table 3 shows configuration of the grid in two other case studies.

In case study 02, we assume unstable power supply and consumer demand unchanged. Generator 1 can change from 80% to 100% of its capacity (9 to 12 power). With this simulation, we create a state space from which we can analyze some situations such as Consumers 1, 2 and 3 need corresponding amounts of electricity 6, 9 and 6, then Generators need At what level of power production will be sufficient and what is the most optimal and economical configuration.

In case study 03, we assume unstable power supply and consumer demand unchanged. Generator 1 can change from 80% to 100% of its capacity (9 to 12 power) like case study 02 and they can be shut down randomly.

Table 3. The configuration of the Grid in three case studies.

	Configuration
Case study 01	Capacity power of generators are fixed
Case study 02	Generators can change from 80% to 100% of its capacity.
Case study 03	Generators can change from 80% to 100% of its capacity and on-off randomly.

4.2 Experimental results

In this section, we analyze the results of three case studies. Table 4 shows the state space report generated of 3 case studies. In case study 02, the report consists of 367 nodes and 5474 arcs. From the generated state space, we find 6 states where Consumers achieve the desired amount of power e.g.: N326, N344, ... In case study 03, the report consists of 536 nodes and 13691 arcs. We can see a huge increase in the number of nodes and arcs. Especially the number of arcs nearly tripled. From the generated state space, we also find 6 states where Consumers achieve the desired amount of power

However, to find out the configuration for the Generators, it is necessary to analyze the arcs of the graph.

Table 4. The results of three case studies

	Nodes	Arcs	Excepted nodes
Case study 01	48	96	1
Case study 02	367	5474	6
Case study 03	536	13691	6

5. CONCLUSIONS

In this paper, we have modelled the electrical smart grid in the standard case. It is to confirm that the smart grid can be effectively represented in CPN by our proposed approach. More interesting, the complexity of the smart grid seems not to be the complexity of the CPN model when all components and topology of the grid can be modelled in a small CPN and configuration/ arc-expression/ functions.

One of the good thing in our approach is that, any smart grid represented using this model can be upgrade easily as demonstrated in this paper. Unfortunately, there many other features of smart grids have to be studied more carefully. In the near future, we are going to study how to represent and verify more features such as: (1) power loss in smart grid, (2) using batteries to re-balance to smart grid in on-/off-peak period, etc. Moreover, studies on avoiding formal verification drawbacks such as state space explosion in checking the properties have also been carried out carefully.

Last but not least, the tool for engineers to describe the smart grid topologies and convert into our CPN models has also been provided in the near future.

ACKNOWLEDGMENT

This research is funded by grant number C2020-20-32 from Vietnam National University - Ho Chi Minh City.

REFERENCES

[1] https://en.wikipedia.org/wiki/Smart_grid

[2] The World Bank, *Smart Grid to Enhance Power Transmission in Vietnam*, 2016, https://openknowledge.worldbank.org/handle/10986/24027

[3] Ian Sommerville, *Software Engineering*, 9th ed., Addison-Wesley, 2010, ISBN-10: 0137035152.

[4] Axel van Lamsweerde. 2000. *Formal specification: a roadmap*. In Proceedings of the Conference on The Future of Software Engineering (ICSE '00). ACM, New York, NY, USA, 147-159. DOI=10.1145/336512.336546 http://doi.acm.org/10.1145/336512.336546

[5] Edmund M. Clarke , Jeannette M. Wing, *Formal Methods: State of the Art and Future Directions*, ACM Computing Surveys, vol. 28, pp. 626-643, 1996

[6] Patil S, Zhabelova G, Vyatkin V, McMillin B. *Towards formal verification of smart grid distributed intelligence: FREEDM case*. In IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society. Institute of Electrical and Electronics Engineers. 2016. p. 3974-3979. 7392719 https://doi.org/10.1109/IECON.2015.7392719

[7] Drozdov, Dmitrii & Patil, Sandeep & Yang, Chen-Wei & Zhabelova, Gulnara & Vyatkin, Valeriy. (2018). *Formal Verification of Protection Functions for Power Distribution Networks*. 3550-3555. 10.1109/IECON.2018.8592802.

[8] Naseem, Syed Atif & Uddin, Riaz & Hasan, Osman & Fawzy, Diaa. (2018). *Probabilistic Formal Verification of Communication Network-based Fault Detection, Isolation and Service Restoration System in Smart Grid*. Journal of Applied Logics — IFCoLog Journal of Logics and their Applications. 5.

[9] A. Pnueli, *The Temporal Logics of Programs*, Proc. 18th IEEE Symp. on Foundations of Computer Science, 1977, 46-57.

[10] G.Holzman, *Design and Validation of Computer Protocols*. Prentice Hall, 1991.

[11] Thang H. Bui, Albert Nymeyer, *Formal Verification Based on Guided Random Walks*, 7th International Conference on Integrated Formal Method (iFM 2009), 2009, Dusseldorf - Germany

[12] Khanh Le, Thang Bui, and Tho Quan, *State Space Reduction on Wireless Sensor Networks Verification Using Component-Based Petri Net Approach*, REV Journal on Electronics and Communications, Special Session on "Advanced Computational Methods, Systems, and Industrial Engineering Applications", 5 (3-4), 2015

[13] Lê Ngọc Kim Khánh, Lê Quốc Vũ, Tân Quốc Tiến, Bùi Hoài Thắng và Quản Thành Thơ, *Trừu tượng hoá dựa trên thành phần để kiểm tra tắc nghẽn trên mạng cảm ứng không dây sử dụng mạng Petri*, Scientific Journal of CanTho University, Special issue , 105-112, 2015

[14] Khanh Le, Thang Bui, Tho Quan, Laure Petrucci, and Étienne André, *Congestion Verification on Abstracted Wireless Sensor Networks with the WSN-PN Tool*, IACT Conference (ICSCC 2015), 2015, Singapore - Singapore

[15] Khanh Le, Thang Bui, Tho Quan, Laure Petrucci, and Étienne André, *Component-Based Abstraction of Petri Net Models: An Application for Congestion Verification of Wireless Sensor Networks*, the Sixth International Symposium on Information and Communication Technology (SoICT 2015), 2015, Hue - Việt Nam

[16] Khanh Le, Thang Bui, Tho Quan, Laure Petrucci, and Étienne André, *Congestion Verification on Abstracted Wireless Sensor Networks with the WSN-PN Tool*, Journal of Advances in Computer Networks, 4(1), 33-40, 2016

[17] Khanh Le, Thang Bui, Tho Quan, Laure Petrucci, *A Framework for Fast Congestion Detection in Wireless Sensor Networks Using Clustering and Petri-Net-based Verification*, International Workshop on Petri Nets and Software Engineering 2016, 2016, Toruń - Poland

[18] Khanh Le, Thang Bui and Tho Quan, *A Formal Approach for Congestion Constraints Verification on Wireless Sensor Networks*, The Third Asian Conference on Information Systems - ACIS 2014, 2014, Nha Trang - Việt Nam

[19] Khai T. Huynh, Tho T. Quan and Thang H. Bui, *A bitwise-based indexing and heuristic-driven on-the-fly approach for Web service composition and verification*, Vietnam Journal of Computer Science, 2016, 1-16, 2016

[20] Thang H. Bui, *Designing parallelizing Random-Walk Based Model Checking*, Phát triển Khoa học và Công nghệ, K5-2015 (18), 108-118, 2015

[21] Thang Hoai Bui, Phuong Ba Khanh Dang, *Model checking Petri Nets using NuSMV*, Khoa học và Công nghệ, 49 (4A), 123-140, 2011

[22] Phan T. H. Nguyen, Thang H. Bui, *A Multiple Refinement Approach in Abstraction Model Checking*, Computer Information Systems and Industrial Management - 13th IFIP TC8 International Conference, CISIM 2014, 2014, Ho Chi Minh City - Việt Nam

[23] Morgan, M. Granger and Apt, Jerome and Lave, Lester B. and Ilic, Marija and Sirbu, Marvin A. and Peha, Jon M., *The Many Meanings of 'Smart Grid'*, 2009.

[24] Khizir Mahmud, Animesh K. Sahoo, Edstan Fernandez, Padmanaban Sanjeevikumar, and Jens Bo Holm-Nielsen, *Computational Tools for Modeling and Analysis of Power Generation and Transmission Systems of the Smart Grid*, 2020.

[25] CPN tools, http://cpntools.org.

[26] Dey, A., Chaki, N., & Sanyal, S, *Modeling Smart Grid using Generalized Stochastic Petri Net*, 2011.

[27] Rasel Mahmud, Anupam Mukherjee, Arash Nejadpak, *A Survey on Smart Grid Metering Infrastructures: Threats and Solutions*, 2015.

[28] K. Jensen, *A Brief Introduction to Coloured Petri Nets*, Monographs in Theoretical Computer Science. Springer-VerlagJensen K, 1997.

[29] K. Jensen, *Coloured Petri Nets — Basic Concepts, Analysis Methods and Practical Use, 1997*.

[30] K. Jensen, *Colored Petri Nets: A High-level Language for System Design and Analysis*, 1991.

[31] K. Jensen, *Colored Petri Nets: A High-level Language for System Design and Analysis*, 1991.

[32] K. Jensen, *An Introduction to the Theoretical Aspects of Coloured Petri Nets*, 1994.

[33] K. Jensen, *Condensed State Spaces for Symmetrical Coloured Petri Nets*, 1996.

[34] Vijay Gehlot, *From Petri NETS to Colored Petri NETS: A Tutorial Introduction to NETS Based Formalism For Modeling And Simulation*, 2019.

[35] Chen, Thomas & Sanchez-Aarnoutse, Juan & Buford, John, *Petri Net Modeling of Cyber-Physical Attacks on Smart Grid*, 2011.

[36] Pózna, Anna Ibolya & Fodor, Attila & Gerzson, Miklos & Hangos, Katalin, *Colored Petri net model of electrical networks for diagnostic purposes*, 2018.

[37] Dotoli, Mariagrazia & Fanti, Maria Pia & Mangini, Agostino & Ukovich, Walter, *Fault detection of discrete event systems using Petri nets and integer linear programming*, 2009.

[38] Ashouri, Ahmad & Jalilvand, Abolfazl & Noroozian, Reza & Bagheri, Amir, *Modeling and evaluation of the power system protection using Petri nets*, 2010.