Formal approach for Evolutionary Microgrids Modeling and Simulation

Aya Attia, Clément Foucher, Luiz Fernando Lavado Villa
LAAS-CNRS, Université de Toulouse, UPS, Toulouse, France

RESUME
The evolution of microgrids over time may lead to changes in its models that involve several disciplines like electrical engineering, control systems, and optimization. These models need to be updated regularly whenever there is a change in the composition of the system, as changes in the system's composition directly affect the governing equations of power flow and network dynamics. However, some traditional approaches lack a formal framework to update automatically the models that reflect these composition changes, mainly when considering complex systems integrating several modeling paradigms. To tackle with this, we propose a formal approach based on graph and DEVS formalism that will enable rigorous representation of microgrid behavior and its dynamic structural evolution. Our approach ensures that system equations are automatically updated as the microgrid composition evolves, allowing for more accurate, scalable, and flexible simulations. We validate the efficiency and applicability of our approach through its implementation in Power Flow Analysis (PFA), demonstrating its capability to handle dynamic structure systems across multiple domains.

Microgrid, Power Flow Analysis, graph transformation, DEVS, Theory of Modeling and Simulation

1. INTRODUCTION

The rapid evolution of microgrids introduces considerable challenges in providing formal representation of its models and simulating them, mainly due to changes in their composition. An example of evolutionary microgrids is illustrated in Figure 1.

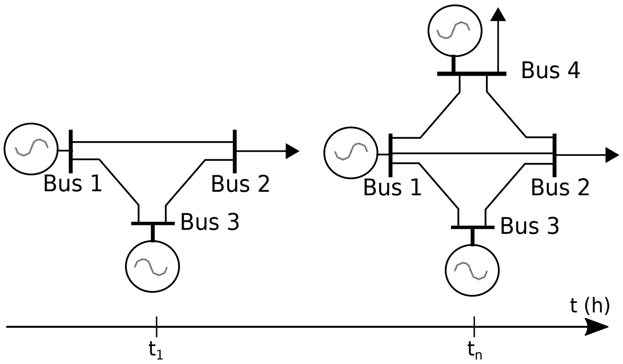


FIG. 1. Example of evolutionary microgrid over time

While generators, loads, and storage devices are added or removed or updated, the models representing the power flow have to be updated at any time. Such structural changes directly affect the mathematical equations describing power flow and network dynamics, for this reason, a method for dynamic adaptation of the models is required.

Additionally, a fundamental challenge in microgrid modeling arises from the fact that its models are loosely defined across different fields, necessitating a system theory backed by a sound mathematical foundation for its semantics [1]. Moreover, modeling microgrids is an intrinsically multi-discipline task, hence interfacing different disciplines, and therefore different model types. Such complexity calls for a framework that is able not only to cope with dynamic structural changes but also to ensure consistency and accuracy.

In this paper, a new approach is proposed that will handle this problem, including the DEVS formalism for rigorous, formal modeling, and graph transformation for presenting changes in behavior and composition of the microgrid over time. Our approach ensures that system equations are dynamically adapted in response to structural modifications, enabling a more flexible and accurate simulation of evolving microgrids. To demonstrate the effectiveness of our methodology, we validate it through its application in Power Flow Analysis (PFA).

2. BUILDING COMPONENTS OF OUR METHOD : GRAPHS AND DEVS FORMALISMS

In this section, we lay the groundwork for building components of our method.

2.1. *Graphs*

Graph theory is [2] a fundamental area of mathematics widely used in control systems to model the interconnections between system elements. For microgrid representation, graphs serve as a powerful tool for representing the system's structure and dynamics. By representing microgrid components such as generators, loads, and storage devices as nodes, and their interconnections as edges, graphs provide a clear and intuitive visualization of the microgrid's configuration. For instance, a study discusses how graph theory can model the connections between elements in microgrids [3], emphasizing its applicability in power flow analysis.

To represent its dynamic structure behavior, features of graphs provide powerful framework to represent it among these features graph transformation. Graph transformation techniques offer a formal approach for modeling dynamic structure systems. The intention of graph transformation techniques [4] is the modification of graphs by applying certain rules in accordance with changes that the system's structure undergoes over time. This is of particular interest for microgrids, where components are added and removed or replaced frequently.

An example of graph transformation is illustrated in Figure 2, where L_1 represents the pattern graph that must be present in the current graph to apply the rule instruction shown in R_1. In this example, the pattern graph specifies three interconnected nodes : S, L, and G, representing Slack, Load, and Generator respectively. The rule states that if a graph contains this pattern, a new generator node should be added to the existing structure, thereby augmenting the microgrid's generation capacity.

2.2. *DEVS formalism*

The Discrete Event System Specification (DEVS) formalism, introduced by Zeigler [5], provides a rigorous framework for modeling discrete-event systems, establishing a solid foundation for the formal description of microgrid behavior. DEVS defines two fundamental types of models : atomic models and coupled models. Atomic models represent the basic behavioral units and define the state transitions and outputs of individual com-

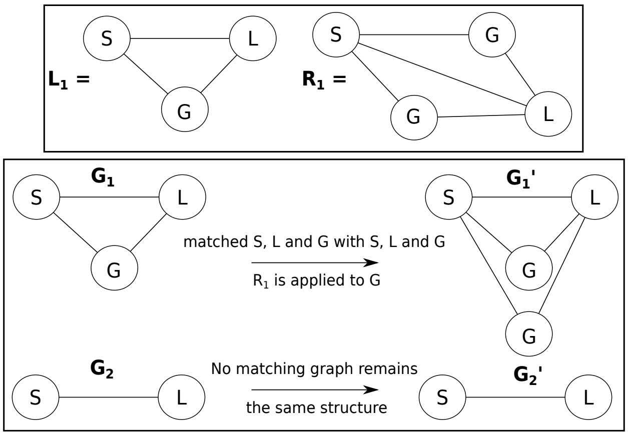


FIG. 2. Example of application of graph transformation for microgrids

ponents. Coupled models specify the structural composition of systems by coupling atomic and/or other coupled models, representing the hierarchical and modular interconnections between components.

DEVS is inherently hierarchical and modular, allowing the representation of microgrid components and their interconnections, which evolve over time. It allows for the formal modeling of complex systems by defining components that are hierarchically related and connected, following rules defined within the formalism. It characterizes the structure and behavior of a system, using sets and systems theory mathematics in DEVS. This formalism enables real-world systems modeling and simulation by providing formal description of its components and relationship between them, adhering to the rules defined within the formalism.

A DEVS atomic model [5] is described by Equation (1) :

$$A =< X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta > \qquad (1)$$

Where :

- $X = \{(p, v) \mid p \in InPorts, v \in X_p\}$ is the set of input ports and values,
- $Y = \{(p, v) \mid p \in OutPorts, v \in Y_p\}$ is the set of output ports and values,
- S is the set of sequential states,
- $\delta_{ext} : Q \times X \rightarrow S$ is the external state transition function, with :
  - $Q = \{(s, e) \mid s \in S, 0 < e < ta(s)\}$ is the total state set,
  - e is the time elapsed since last transition
- $\delta_{int} : S \rightarrow S$ is the internal state transition function,
- $\lambda : S \rightarrow Y$ is the output function
- $ta : S \rightarrow \mathbb{R}_0^+ \cup \infty$ is the time advance function

In DEVS, a model is defined by a set of inputs X, outputs Y and states S. Each element of X and Y sets is defined by a pair of key and value and represent a potential event v on a port p. In the DEVS formalism, components state belong to the S set and can be changed using transition functions. When an input event occurs, δ_{ext} the external transition function is executed to change the component's state. If no external event occurs, the component will stay in its state until the period of time returned by time advance function ta(s) is reached, then it will change its state using the internal transition. When the state change, either as a result of internal or external event, the result of ta(s) will change accordingly. The component whose ta is minimal amongst all components is said to be imminent. The output function λ is executed when the state changed as a result of a component being imminent. After delivering the output, the imminent components list is updated.

A DEVS coupled model [5] is described by Equation 2 :

$$N =< X, Y, D, \{M_d \mid d \in D\}, EIC, EOC, IC > \qquad (2)$$

Where :

- D is the set of the component names,
- For each $d \in D$, $\{M_d\}$ is a DEVS model,
- External input coupling connects external inputs to component inputs :
  $EIC \subseteq \{((N, ip_N), (d, ip_d)) \mid ip_N \in IPorts_N, d \in D, ip_d \in IPorts_d\}$
- External output coupling connects component outputs to external outputs :
  $EOC \subseteq \{((d, op_d), (N, op_N)) \mid op_N \in OPorts_N, d \in D, op_d \in OPorts_d\}$
- Internal coupling connects component outputs to component inputs :
  $IC \subseteq \{((a, op_a), (b, ip_b)) \mid a, b \in D, op_a \in OPorts_a, ip_b \in IPorts_b\}$

Coupled models dictate the system composition by describing how components will interact during simulation. In a coupled model a set of the network inputs X and outputs Y are defined. The D set contains the names of the components contained into this model. A coupled model connects the components to each other thanks to connections defined into EOC, EIC, and IC sets.

Figure 3 provides the architecture of an example of interconnected coupled and atomic models. This hierarchical structure demonstrates how complex systems can be constructed through component description within the DEVS formalism.

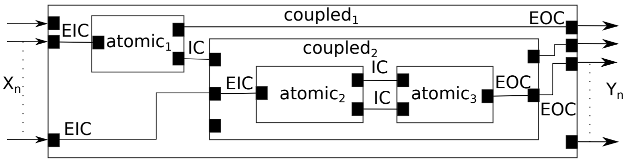


FIG. 3. Architecture of an example of interconnected DEVS atomic and coupled components

In the context of a microgrid, DEVS addresses the challenge of providing formal descritpion of its models. This involves the hierarchical, modular nature of its description entailed in DEVS, using its mathematical rigor to model effectively and simulate emerging microgrid compositions and behaviors.

Graph transformation techniques integrated with the DEVS formalism enhance the modeling and simulation of microgrids. DEVS enables the rigorous definition of system behaviors, while graph transformations manage structural changes. In this way, this combination guarantees that during the evolution of the microgrid, its structure and behavior will be properly represented and simulated. One such example follows the study concerning modeling and simulation of renewable sources in smart grids by DEVS formalism and proves to be quite efficient [6].

3. FORMAL APPROACH FOR EVOLVING MICROGRIDS REPRESENTATION

Our proposal enable to represent a microgrid as graph where the vertices represent microgrid components and edges represent the links between these components. Figure 4 illustrates the representation of a microgrid using graph theory.

For our proposal, we developed a modular simulator which architecture is described in Figure 5. The simulation of microgrid is ensured using PythonPDEVS simulator [7].

The simulator is structured into a clock and three main DEVS coupled components *data generation*, *equations generation* and *power flow analysis*. Each coupled component consist of a set of interconnected atomic components.

The *data generation* component is responsible for handling graph-based system representations, ensuring that structural modifications are accurately captured and integrated. This component contains atomic DEVS components such as the *decsion*

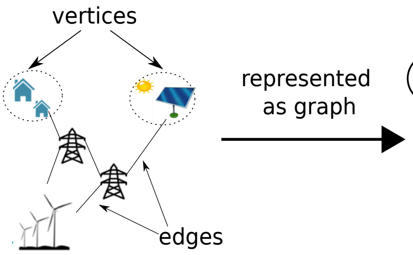


FIG. 4. Microgrid representation as graph

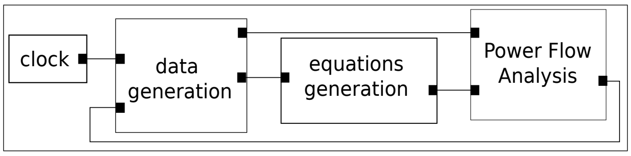


FIG. 5. Architecture of PFA simulator

driven data generator, *process input data* and *checker*, which are responsible for generating input data in the form of graphs and validating its correctness. Figure 6 provides a zoom of the constituents of *data generation* component.

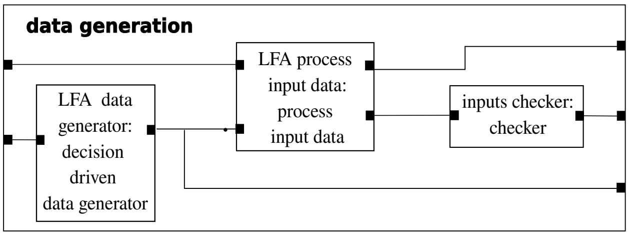

data generation
LFA data generator: decision driven data generator
LFA process input data: process input data
inputs checker: checker

FIG. 6. Data generation coupled component

The *equations generation* produces the required mathematical expressions, facilitating the computation of system dynamics. This coupled DEVS component composed of atomic DEVS components named *symbols generator* and *equations generator* which are secialized for PFA. These specialized components are responsible for generating the mathematical representation of microgrid elements, including mathematical symbols, equation functions, and other related constructs. Figure 7 provides an overview of the constituents of *equation generation* component.

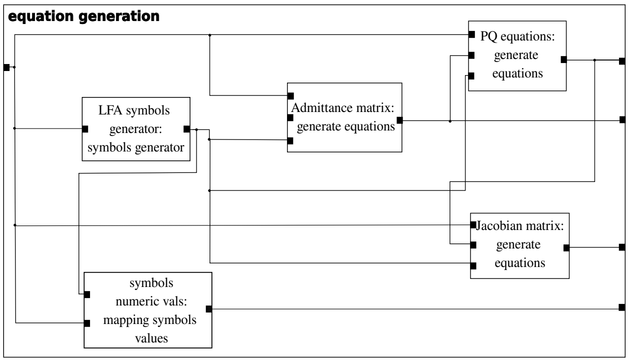

equation generation
PQ equations: generate equations
LFA symbols generator: symbols generator
Admittance matrix: generate equations
Jacobian matrix: generate equations
symbols numeric vals: mapping symbols values

FIG. 7. Equation generation coupled component

Lastly, the *Power Flow Analysis* component that performs the core computations to determine the steady-state operating conditions of the microgrid. This component is composed of interconnected atomic components such as *compute equations*, *convergence checker*, *solve equations* and *checker*. Figure 8 provides a zoom of the constituents of *Power Flow Analysis* component. Once the core computations are completed, this component provides results as a graph containing the computed data and sends them to the *data generation* component. The *data generation* component then interprets the provided data and determines, based on the simulation script, if there is any need for structural changes in the microgrid, such as the addition of new loads or generators, or the replacement of existing ones. It then decides how and which data to generate to reflect these changes in the form of a graph.

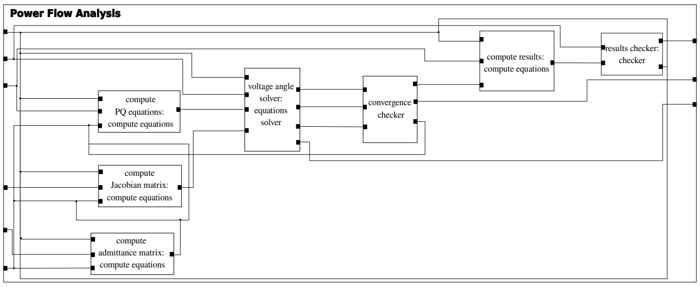

Power Flow Analysis
compute PQ equations: compute equations
voltage angle solver: equations solver
convergence checker
compute results: compute equations
results checker: checker
compute Jacobian matrix: compute equations
compute admittance matrix: compute equations

FIG. 8. Power Flow Analysis component

Detailed description of these components and their implementation in executable code is available in a git repository [8].

4. IMPLEMENTATION RESULTS

We tested our simulator for a microgrid case initially composed of two buses, each representing a microgrid component. When the load reaches a threshold of P_l = 2.2pu, a new generator bus is automatically added and the microgrid structure changed. The generated graph incorporates an additional generator compared to the previous configuration, as illustrated in Figure 9.

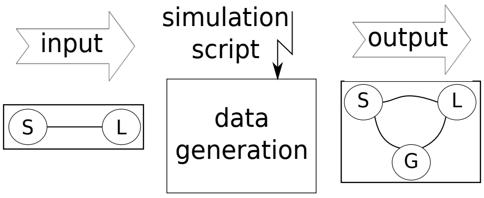
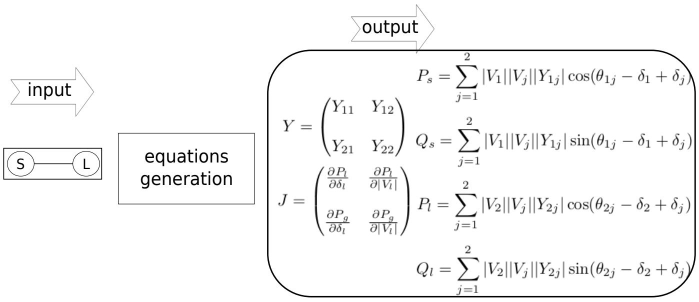


FIG. 9. Load active power evolution and generator addition trigger

As new generator is added, the active and reactive power equations are altered as shown in Figures 10,11. For two bus microgrid the equations were the sum of two, while for three bus microgrid as a new bus is added, the sum become the contribution of the three buses.



$$Y = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix} \quad J = \begin{pmatrix} \frac{\partial P_l}{\partial \delta_l} & \frac{\partial P_l}{\partial |V_l|} \\ \frac{\partial P_g}{\partial \delta_l} & \frac{\partial P_g}{\partial |V_l|} \end{pmatrix}$$

$$P_s = \sum_{j=1}^{2} |V_1||V_j||Y_{1j}| \cos(\theta_{1j} - \delta_1 + \delta_j)$$

$$Q_s = \sum_{j=1}^{2} |V_1||V_j||Y_{1j}| \sin(\theta_{1j} - \delta_1 + \delta_j)$$

$$P_l = \sum_{j=1}^{2} |V_2||V_j||Y_{2j}| \cos(\theta_{2j} - \delta_2 + \delta_j)$$

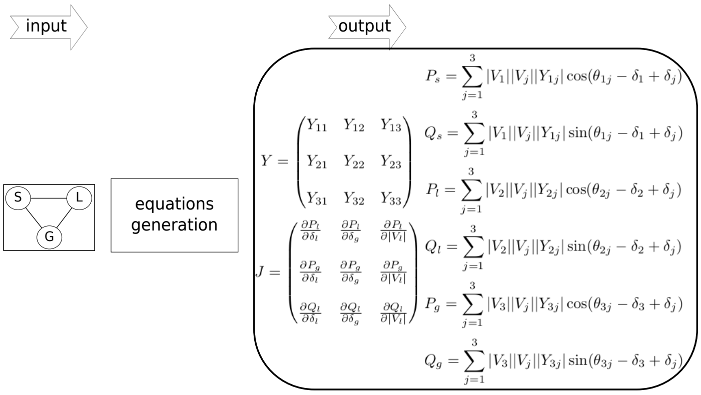
$$Q_l = \sum_{j=1}^{2} |V_2||V_j||Y_{2j}| \sin(\theta_{2j} - \delta_2 + \delta_j)$$

FIG. 10. Power flow equations of two bus microgrid

Figure 12 illustrates our study case. At t=3h the expansion is applied to the structure of the micro-grid and the results change.

5. CONCLUSION

This paper presents a formal approach combining DEVS with graph transformation to model and simulate microgrids and its structural evolution over time. The microgrid composition is represented using graphs, its structural evolution is represented



$$Y = \begin{pmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{pmatrix}$$

$$J = \begin{pmatrix} \frac{\partial P_l}{\partial \delta_l} & \frac{\partial P_l}{\partial \delta_g} & \frac{\partial P_l}{\partial |V_l|} \\ \frac{\partial P_g}{\partial \delta_l} & \frac{\partial P_g}{\partial \delta_g} & \frac{\partial P_g}{\partial |V_l|} \\ \frac{\partial Q_l}{\partial \delta_l} & \frac{\partial Q_l}{\partial \delta_g} & \frac{\partial Q_l}{\partial |V_l|} \end{pmatrix}$$

$$P_s = \sum_{j=1}^{3} |V_1||V_j||Y_{1j}| \cos(\theta_{1j} - \delta_1 + \delta_j)$$

$$Q_s = \sum_{j=1}^{3} |V_1||V_j||Y_{1j}| \sin(\theta_{1j} - \delta_1 + \delta_j)$$

$$P_l = \sum_{j=1}^{3} |V_2||V_j||Y_{2j}| \cos(\theta_{2j} - \delta_2 + \delta_j)$$

$$Q_l = \sum_{j=1}^{3} |V_2||V_j||Y_{2j}| \sin(\theta_{2j} - \delta_2 + \delta_j)$$

$$P_g = \sum_{j=1}^{3} |V_3||V_j||Y_{3j}| \cos(\theta_{3j} - \delta_3 + \delta_j)$$

$$Q_g = \sum_{j=1}^{3} |V_3||V_j||Y_{3j}| \sin(\theta_{3j} - \delta_3 + \delta_j)$$

FIG. 11. Power flow equations of three bus microgrid

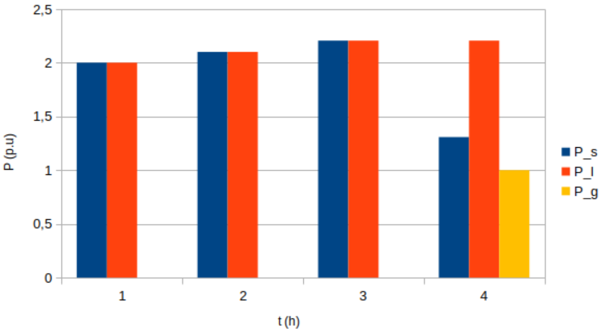


FIG. 12. Charts of slack, load and generator active power

using graph transformation, while the simulation is carried by DEVS simulator. The simulator processes the updated graphs iteratively, maintaining mathematical rigor and flexibility.

The approach is demonstrated through Power Flow Analysis. However, challenges remain, including scalability for larger microgrids. Additionally as part of future work, we aim to validate our approach by comparing our simulation results with those obtained from other simulators, such as those provided by Linux Foundation, PyPSA, etc.

6. ACKNOWLEDGEMENT

We would like to acknowledge Khalil Drira for his valuable guidance and support.

7. RÉFÉRENCES

[1] A. Boche, L. F. L. Villa, and C. Foucher, "Study of the sustainability of microgrids using systemic models based on the theory of modeling and simulation," Ph.D. dissertation, Université Paul Sabatier (Toulouse 3), 2023.

[2] A. Bickle, *Fundamentals of graph theory*. American Mathematical Soc., 2020, vol. 43.

[3] A. K. Reyes, A. I. Hernandez, R. M. Gutierrez, N. Bolívar, D. A. Jimenez, J. D. Bastidas, and J. Solano, "A novel extended graph strategy to model microgrids," in *2019 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 2019, pp. 1–6.

[4] B. König, D. Nolte, J. Padberg, and A. Rensink, "A tutorial on graph transformation," *Graph Transformation, Specifications, and Nets : In Memory of Hartmut Ehrig*, pp. 83–104, 2018.

[5] B. Zeigler, A. Muzy, and E. Kofman, *Theory of Modeling and Simulation : Discrete Event & Iterative System Computational Foundations.* Academic press, December 2018.

[6] M. Jarrah, "Modeling and simulation of renewable energy sources in smart grid using devs formalism," *Procedia Computer Science*, vol. 83, pp. 642–647, 2016.

[7] Y. Van Tendeloo and H. Vangheluwe, "Pythonpdevs : a distributed parallel devs simulator," in *Proceedings of the Symposium on Theory of Modeling & Simulation : DEVS Integrative M&S Symposium*, 2015, pp. 91–98.

[8] A. Attia, "Approach to model simulate load flow analysis," https://gitlab.laas.fr/aattia/load_flow_analysis_simulator, 2025.