Simuleau: a tool for hybrid and batches Petri nets

Leonardo Brenner, Rabah Ammour, Isabel Demongodin

Simuleau: a tool for hybrid and batches Petri nets ⋆

Leonardo Brenner * **Rabah Ammour** ** **Isabel Demongodin** **

* *Université de Reims Champagne-Ardenne, CEA, LRC DIGIT, LICIIS, chaire OSMIUM, Reims, France, (email: leonardo.brenner@univ-reims.fr)*
** *Aix-Marseille Université, CNRS, LIS, Marseille, France, (email : {rabah.ammour, isabel.demongodin}@lis-lab.fr)*

Abstract: Simuleau is a modeling, analysis and simulation tool for discrete event models expressed by batches Petri nets, a formalism that enriches hybrid Petri nets of David and Alla. Batches PNs incorporate timed-transition discrete PNs, constant continuous PNs, and new types of nodes called batch nodes. This formalism and Simuleau have been exploited among diverse application areas such as manufacturing systems, communication networks, and traffic road. After presenting batches PN formalisms and their application domains, this paper focuses on the main characteristics of Simuleau, including the tool structure, its menu interface, the input model description and the outputs obtained after simulation.

Keywords: Discrete Event Systems, Batches Petri Nets, Hybrid Petri Nets, Tool, Simulation

1. INTRODUCTION

Petri nets (PNs) are a widely recognized formalism for modeling and analyzing a very large range of discrete event dynamic systems. Thanks to fluidization concepts (Silva et al., 2011), continuous and hybrid PNs (David and Alla, 2005) have emerged in the 90's allowing the flow of tokens to be considered as continuous variables. Few years later, an extension of hybrid PNs has been proposed (Demongodin and Prunet, 1992) to express variable delays in continuous flows, but also to represent certain accumulation phenomena with the concept of density related to the flow. More precisely, generalised batches Petri nets (GBPNs) (Demongodin, 2001) are considered as a class of hybrid formalism, which enhance continuous and hybrid PNs with special nodes, called batch places and batch transitions. Batch places are characterized by three continuous parameters: a maximal speed, a maximal density, and a length that, allowing a hybrid representation of the linear relations between flow and density to be expressed within a single node. The batch place markings are composed of batches, i.e., a group of entities moving, at a certain speed, inside a batch place. A batch place combines both discrete events and switched linear continuous dynamics in a single structure. In addition to this hybrid behavior, GBPNs are a combination of timed-transition discrete PNs and constant continuous PNs, also known as continuous PNs with finite server semantics in the PN literature (Giua and Silva, 2017). GBPN has the ability to model a larger class of hybrid systems and to provide efficient algorithms for simulation and/or analysis of such systems.
Several tools exist to model, analysis and, simulate PNs and their extensions. An extensive list can be obtained at `https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html`. To the best of our knowledge, only few of them are dedicated to hybrid PNs, such as SimHPN and Mochy, which appear to be the only tools currently maintained. SimHPN (Júlvez et al., 2012) is a MATLAB toolbox devoted to hybrid and continuous PNs under infinite server semantics. Mochy (Hélouët and Thébault, 2023) is a toolbox developed in Java that takes into account stochastic, timed, and hybrid PNs. It has been developed as a fast simulation platform with the aim of testing traffic management policies for metro networks. This paper is dedicated to Simuleau, a tool developed in C++ that enables the modeling, simulation and analysis of systems described by timed discrete PNs, continuous PNs with single server semantics, hybrid PNs and of course batches PNs.

The content of this paper is as follows. Section 2 presents the concepts of the GBPN formalism, next describes some extensions of such a formalism, and finally highlights its efficiency in regards to some application areas. Section 3 is dedicated to Simuleau, the tool that supports the class of batches PN formalisms. The final section includes conclusions and perspectives.

2. CONCEPTS AND APPLICATIONS OF GENERALISED BATCHES PETRI NETS

Let us first give some concepts taken from (Demongodin, 2001) and (Demongodin and Giua, 2014).

2.1 Concepts in GBPN

A *generalised batches Petri net* (GBPN) is a bipartite graph composed of three types of places, $P = P^D \cup P^C \cup P^B$, and three types of transitions, $T = T^D \cup T^C \cup T^B$: discrete places and transitions, continuous places

⋆ This work has been partially supported by the French National Research Agency under grant agreement ANR-22-CE10-0002.

and transitions, and batch places and transitions (see Figure 1 for their graphical representation). A batch place is characterized by three continuous parameters, $\gamma(p_i) = (V_i, d_i^{\max}, S_i) \in \mathbb{R}^3_{0+}$, which represent, respectively, a *maximal transfer speed*, a *maximal density* and a *length*. These characteristics associated with a batch place allow a hybrid representation of the linear relations between flow and density to be expressed in a single node. With each transition is associated a non negative number: a *firing delay*, d_j, for a discrete transition, and a *maximal firing flow*, Φ_j, for a continuous or batch transition. At time τ, the marking, $\boldsymbol{m}(\tau)$, assigns to each discrete place an integer number, to each continuous place a nonnegative real number and, to each batch place p_i, a series of batches, (i.e., a group of discrete entities) ordered by their head positions, $m_i(\tau) = \{\beta_i^1(\tau), \ldots, \beta_i^k(\tau)\}$. A *batch* β_i^r of batch place p_i is also characterized by three continuous variables: $\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau)) \in \mathbb{R}^3_{0+}$, where $l_i^r(\tau)$ is the length, $d_i^r(\tau)$ is the density and, $x_i^r(\tau)$ is the head position. If the head position of a batch is equal to the length associated with the batch place, i.e., $x_i^r(\tau) = S_i$, this batch is called an *output batch* denoted as $O\beta_i^r$. An *output density* $d_i^{\text{out}}(\tau) = d_i^r(\tau)$, is associated with batch place p_i containing an output batch $O\beta_i^r$. Note that $d_i^{\text{out}} = 0$ when no output batch exists for batch place p_i. Moreover, a batch place can have at most one output batch.

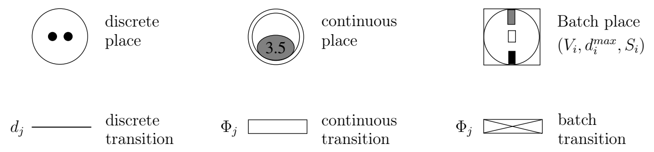

Fig. 1. Nodes of batches Petri nets

Example 1. Figure 2 shows a simple example of a GBPN model with discrete, continuous, and batch nodes (place/-transition). The batch place is characterized by a maximal speed of 20, a maximal density of 100, and a length of 10. The marking of this batch place includes a batch $O\beta_1^{Bp}$ with a length of 5, a density of 50, and a position of 10, which implies that this batch is an output batch. ♢

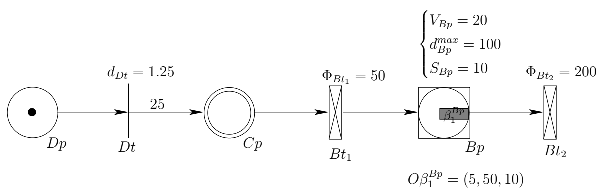

Fig. 2. A simple GBPN model

During the dynamics, inside the batch places, batches can be: i) *created*, when the input flow of the place is not null; ii) *merged*, when two batches with the same density are in contact; iii) *destroyed*, if the length of a batch becomes null and if it is not a created batch. Their movements are governed by hybrid dynamics that switch between free behavior and accumulated behavior, depending on the characteristics of the place and of the batch itself. In free behavior, a batch moves at the speed of place p_i and the value of its density remains constant within interval $[0, d_i^{max}]$. In such a case, the flow of batch $\beta_i^r(\tau)$ is characterized by: $\varphi_i^r(\tau) = V_i . d_i^r(\tau)$. In accumulated behavior, it moves at a lower speed of the place and its density is equal to the maximal density of place p_i, i.e., $d_i^r(\tau) = d_i^{max}$. Thus, it is forced to adapt its own flow to the output transition flow of the place. The flow-density relation that governs batches moving through batch place p_i is represented in Figure 3.

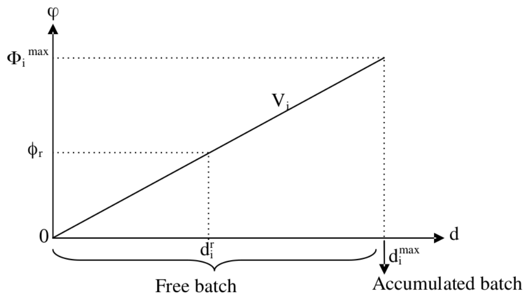

Fig. 3. Flow-density relation in batch places

The enabling and firing conditions of discrete transitions are those of the classical timed-transition PN (David and Alla, 2005) with a preselection policy for which the marking of discrete or continuous places is reserved. Details on conditions of continuous and batch transitions, could be found in Section 2.3 of Demongodin and Giua (2014).

In order to represent the quantity of firing of transitions by time unit, an *instantaneous firing flow* (IFF), noted $\varphi_j(\tau) \leq \Phi_j$, is associated with each continuous or batch transition, $t_j \in T^C \cup T^B$. At time τ, the set of admissible firing flows forms a convex set described by linear equations (Demongodin and Giua, 2010). In particular, the firing flow of continuous and batch transitions, and the input/output flow of batch places must satisfy the following constraints: the firing flow of a transition that is not enabled is null; the total flow entering in batch place p_i should not be greater than its maximal flow $V_i \cdot d_i^{\max}$; the total flow exiting batch place p_i should not be greater than the exit flow $V_i \cdot d_i^{out}$ generated by the output batch of the place.

The behavior of a GBPN is based on a timed discrete event dynamics with linear or constant continuous evolutions between timed events. Between two consecutive events, the system is said to be in an *invariant behavior state* (IB-state), which corresponds to a period of time such that the following elements remain constant: the total marking and the reserved marking of the discrete places, the reserved marking of continuous places, the IFF of continuous and batch transitions, and the output density of batch places. The IB-state changes if and only if one (or possibly several at the same time) of the following kind of events occurs: (i) *Internal events of batch places*: a batch becomes an output batch $\beta_r = O\beta_r$; two batches meet; a batch is destroyed. (ii) *External events of batch places*: a discrete transition is fired; a continuous place becomes empty; a discrete transition becomes enabled; a batch becomes an output batch; an output batch is destroyed.

2.2 GBPN extensions

Several extensions of the GBPN have been proposed in the last decades.

a) GBPN and controlled characteristics: The first extension has been defined as controlled batches PN (CBPN) (Audry and Prunet, 1994) that introduces the possibility of controlling the maximal firing flow of transitions and/or the speed of batch places. In the same spirit, the concept of *controllable batch* has been defined in (Demongodin, 2009) by a quadruple, $C\beta_i^k(\tau) = (l_i^k(\tau), d_i^k(\tau), x_i^k(\tau), v_i^k(\tau)) \in \mathbb{R}^4_{\geq 0}$, where $l_i^k(\tau)$ is its length, $d_i^k(\tau)$ is its density, $x_i^k(\tau)$ is its head position and $v_i^k(\tau)$ is its speed. Hence, the speed of a batch can be changed by controlled action as long as it does not exceed the maximal speed of the batch place. In this extension, batches inside a batch place can move at different speeds.

When the instantaneous firing flow of continuous and batch transitions, and the transfer speed of batch places are considered as control inputs in GBPN, this leads to the definition of Controlled GBPN (CGBPN). It has been shown by (Demongodin and Giua, 2014) that the stationary behavior of this model without discrete nodes can be characterized using structural analysis. Moreover, by controlling only the flow of continuous and batch transitions of a CGBPN model without discrete nodes, the system can be driven to *a steady state*, $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$, where the marking $\boldsymbol{m}^s$ and the instantaneous firing flow vector $\boldsymbol{\varphi}^s$ remain constant for $\tau \geq \tau_s$.

b) Triangular Batches PN: A new formalism of batches PN, called Triangular Batches Petri Nets (TBPNs) has been defined (Gaddouri et al., 2016), where the batch places are replaced by Triangular batch places (TB-places) and controllable batches form their marking. As previously defined for batch place, each TB-place has the three previous characteristics, i.e., a maximal speed, a maximal density and a length, and a new one, called a maximum flow. Precisely, with each TB-place p_i are associated the quadruple $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ and a new flow-density relation that governs batches behavior (see Figure 4). The accumulation behavior of GBPN is thus extended to a congested behavior which imposes to batches a congested speed and a congested density in accordance to a *propagation speed of congestion*, denoted W_i, and a *critical density*, denoted d_i^{cri}, of TB-place p_i. Hence, each batch inside a TB-place has its own specific speed, density, and an hybrid dynamics switching between free and congested behaviors. Note that the controlled GBPN has been extended to define Controlled TBPN (CTBPN) (Gaddouri et al., 2016).

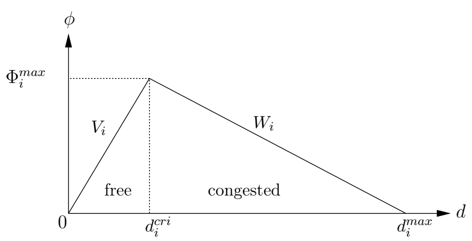

Fig. 4. Flow-density relation in triangular batch places

c) Other extensions: Several other extensions have been defined in the literature. We can cite Coloured batches PNs (Caradec and Prunet, 1997) in which it is possible to identify several batches with different characteristics moving in the same batch place. We can also cite the extension proposed by Wang and Zhou (2004) that combines batches PN and stochastic Petri nets to integrate a stochastic behavior over discrete transitions. In this extension, the set of discrete transitions is partitioned into immediate, deterministic timed, and stochastic transitions.

2.3 Applications

Batches PNs and their extensions have been exploited by several authors for different purposes and variety of application domains. We next focus on three domains related to the use of Simuleau tool.

a) High speed production systems: Batches PNs are very well suited to describe high-throughput systems in which there is a transfer of material moving through space at a predefined speed and, in which the density of the transferred material could change in the presence of accumulation. The first application in this context concerns the Perrier mineral water bottling production line system. The BPN model was mainly composed by batch nodes representing machines connected by multi conveyor belts. With this model, it has been possible to evaluate the performance of the bottling lines. Subsequently, sensors were integrated into the system and a CBPN model was proposed (Audry et al., 1994) to provide control of the manufacturing system by modifying the throughput of the machines as well as the speed of conveyors. CBPN has been also used to model flexible manufacturing systems such as the Perrier spring water processes.

b) Traffic road: The traffic road phenomena have been modeled by batches PNs and its extensions, where a section of road is simply modeled by one batch place and so, there is no need to divide a section into cells, as ~~it is~~ usually done by discrete traffic models. Moreover, a batch represents a group of vehicles that move together with the same speed in a section. In (Demongodin, 2009), the traffic road is characterized by multiple batches with the possibility that a given batch overpasses another one with a lower speed. In (Gaddouri et al., 2016) CTBPN has been proposed and used to model traffic road networks where the flow of each batch inside the place must respect the triangular diagram of the traffic road. CTBPN models are particularly well adapted to represent the congestion/decongestion phenomena of flows of vehicles and to determine the length of traffic congestion.

c) ON/OFF control: The control of GBPN models without discrete nodes allow us to drive a system from its initial state to its steady state $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$. Three control laws, based on an event-driven ON/OFF control strategy, have been developed on CGBPNs without discrete nodes. The first one, called steady-flow control (Liu et al., 2020a) limits the IFF to be lower than the steady-flow value. The second one, called maximal-flow control (Liu et al., 2020b), relaxes this limitation, and governs the transient behavior by minimizing the transitory delay at the expense of a larger number of events. The last one, called ZF-control (Liu et al., 2023), drives the GBPN model from any blocking marking to an attractive region of the steady state.

d) Other applications: Dedicated to communication systems, Svadova and Hanzalek (2003) have represented the signal propagation on the physical layer of the media access control and analyzed the transmission delay. In transportation systems, the traveller connection time on multimodal hub has been evaluated by Kaakai et al. (2005), while Durmus et al. (2012) have described the movement of consecutive trains for monitoring the train movements in railway networks.

3. SIMULEAU

Simuleau is a tool written in C++, which allows the modeling, simulation and analysis of systems described using Petri net formalisms. It is dedicated to timed discrete PNs, continuous PNs with single server semantics, hybrid PNs and batches PNs. As an academic and research tool, Simuleau has been developed to be open-source (it can be freely downloaded from `https://gitlab.lis-lab.fr/leonardo.brenner/simuleau`), simple to use (the interface, input and output file are easy to be written and read), easy to extend (textual interface allows new graphical tools to be easily plugged). In the following, we present the main features and characteristics of the tool.

3.1 Tool architecture

Simuleau's architecture is organized in four main modules: the Simuleau module, the interface module, the Batches PN module, and the scheduler module. Figure 5 shows the class diagram with the main components of Simuleau tool.

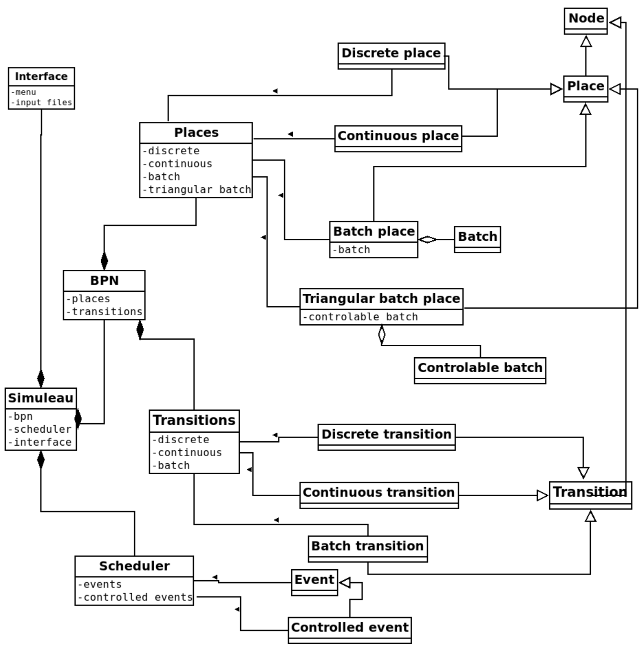

Fig. 5. Simuleau tool class diagram

a) Simuleau module. Simuleau module itself manages the other three modules and implements the main simulation loop. This loop represents the evolution algorithm, illustrated in Figure 6, that drives the dynamic of Batches PN models. The main steps can be summarized as: determining the state and flow of transitions, computing and updating the state and behavior of places for GBPN models and batches for Controllable GBPN and TBPN models (BPN evolution block), and computing the date of the next events (scheduler block). Some steps of this algorithm depend on the kind of the places and events considered in the model. For instance, the characteristics of places do not change if they are not controlled.

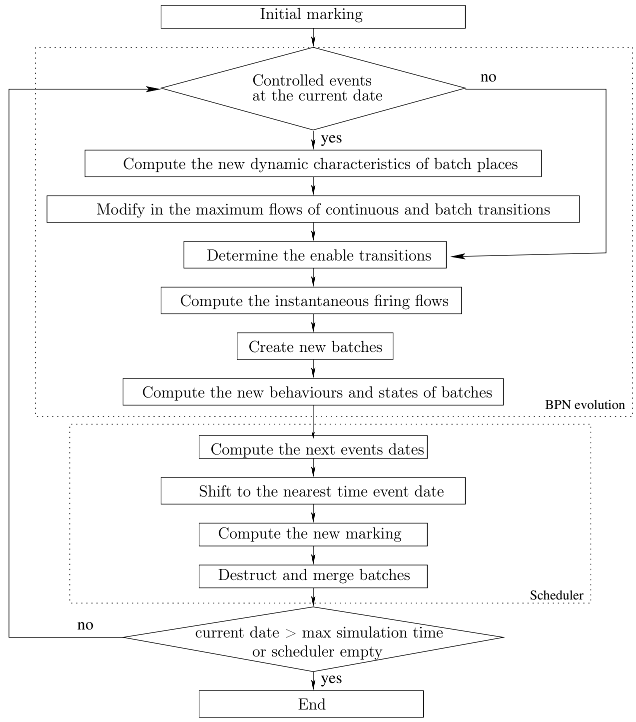

Fig. 6. Evolution algorithm of a Batches PN model

b) Interface module. To make the tool user-friendly, the interface module of Simuleau implements textual menus and a compiler for the input model file. The textual menu interface displays different options to the user, as shown in Figure 7. Option 1 allows the user to compile the input file and verifies the model (see Section 3.2 for the syntax of the input file).

```
+----------------------------------------------------+
|           This is Simuleau - the BPN tool          |
+----------------------------------------------------+

1) Compile a BPN model           4) Inspect data structures
2) Simulate a compiled BPN model 5) Dynamics facilities
3) Preferences                   6) About this version

  0) Exit Simuleau (Option 0 always exits the current menu)
```

Fig. 7. Simuleau main menu

Option 2 of the main menu leads the user to the simulation menu (see Figure 8). The user can now select an option from different simulation algorithms. One of the options simulates the system's evolution without controlled events (autonomous). The remaining options simulate the system's evolution using external controlled events (with maximal transitions flow) or controlled methods that drive the system's evolution to a steady state.

The other options of the main menu are dedicated to parameters or information on Simuleau, and are not described in this paper.

```
******* Simulating a BPN model *******

1) With controlled events        3) SF-On/Off control method
2) Without controlled events     4) MF-On/Off control method
```

Fig. 8. Simuleau simulation menu

c) BPN module. The BPN module implements all the structures and methods required to store and simulate a batches PN model. Each class in this module describes a type of place and transition that is taken into account in Simuleau. Additional types of nodes (places and transitions) can easily be implemented to extend the tool. The places and transitions classes are specialized versions of the general class and inherit its behaviors. The methods implemented in each class compute the evolution of the system between events.

Instantaneous Firing Flow (IFF) computation is a part of the BPN module and it is one of the main features of Simuleau. In the current version of the algorithm, the computation of IFF is based on linear programming methods, specifically the GLPK library is used to describe the constraints and to compute the IFF. According to the control laws, the linear system and its objective function differ. For instance, it could be a maximization of flows, or a more complex one for controlled GBPN without discrete nodes, as briefly described in Section 2.3.c.

d) Scheduler module. The scheduler module calculates, from a current IB-state, the future dates for all possible events that can occur in the model. A set of 13 events are taken into account (some of them are mentioned in section 2, others are only taken into account in specific extensions of Batches PN). The events that drive the simulation of the system are divided into three types: internal, external and controlled events. More information on these events can be found in Demongodin (2001); Demongodin and Giua (2014); Gaddouri et al. (2016); Liu et al. (2020b).

3.2 Input of Simuleau: model description

Another important feature of Simuleau is the input model description. PN models are described in a simuleau-specific input format that favors human-readability. The input file is structured in three blocks.

The first block gives a name to the model and sets the length and time units.

```
// double "/" for comments
model example;
length unity=km;
time unity=h;
```

Listing 1. Name and unity definition

The second block is dedicated to the model description. It is organized into two sub-parts. The former, indicated by the keyword `places`, describes the places of the model and its output arcs, while the latter, indicated by the keyword `transitions`, specifies the transitions and their output arcs. Different types of places (discrete, continuous, batch, and triangular) and transitions (discrete, continuous, batch) can be described. Each one has specific parameters, and some are optional (as output arc, if the node does not have it, for instance).

Example 2. A description of the batch place and batch transition Bt_2 is given below.

```
places
  place Bp (batch)
    function (20, 100, 10)
          // speed, max density, and length
    initial marking {(5, 50, 10)}
    // list of initial batches
    // {(length, density, position), ..}
    output arc Bt2

transitions
  transition Bt2 (batch)
    flow (200.0) // real value
```

Listing 2. Network description

The third block of the input file concerns controlled events. Following the formalism definitions, it is possible to change the maximal flow of a transition and the maximal speed of a place. These controlled events will be included in the scheduler, and the changes will be made at the specified time.

```
// the section of controlled events are optional
controlled events
  max_speed_change=(speed, Bp, 100, 0.4);
 // event type, concerned place, new speed, time
```

Listing 3. Controlled events and time inspection

3.3 Outputs of Simuleau

To analyse the behavior of batches PNs, an evolution graph could be constructed. It is composed by nodes, representing an IB-state. In a node of a GBPN (see Figure 9), the first part presents the marking and reserved marking of discrete places; the second part presents the flow of continuous transitions and the reserved marking of continuous places, while the marking of the continuous places is given for the begin and the end of the IB-state; the third part presents the flow of batch transitions and the marking of batch places. The characteristics of each batch are given at the right side of the node. Two nodes are linked by a transition labelled with the next event and the $\Delta\tau$ between them. A node of a TBPN slightly differs from the one given in Figure 9, as the global state of a place is replaced by the values of speed, state and behavior associated with each batch.

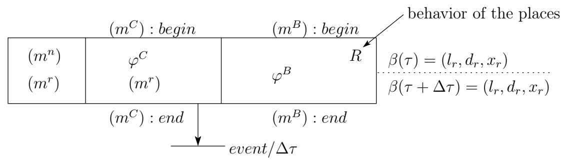

Fig. 9. Node of the evolution graph of a GBPN

When a compiled model is simulated, an output file is generated. For each date that drives the system's evolution, Simuleau writes the current date, the state of each place and transition, and a list of all upcoming events with their respective dates. The upcoming events that will be processed are listed at the end of each step.

Example 3. Thanks to Simuleau, we can easily construct the evolution graph (see Figure 10) of the GBPN model given in Figure 2.

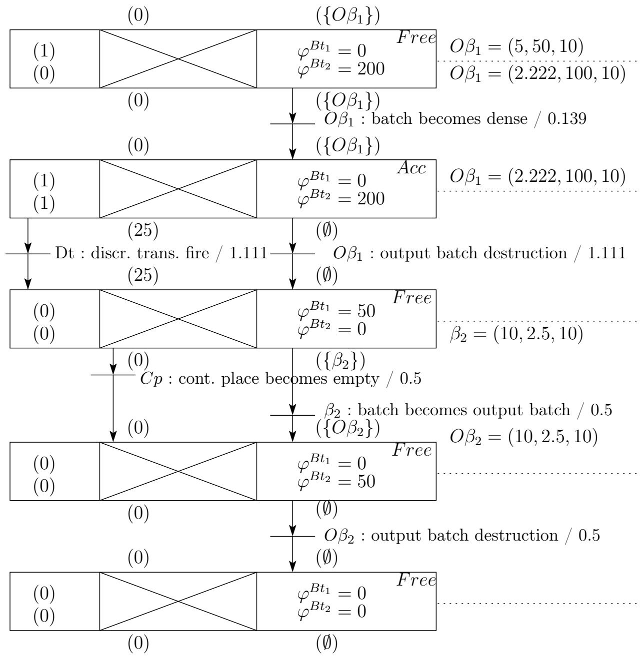

Fig. 10. Evolution graph of example of Figure 2

4. CONCLUSION

Simuleau is a tool for modeling and simulating some classes of PN models, such as timed discrete PNs, constant continuous PNs, hybrid PNs and, batches PN. The program implements various extensions of batches PNs, such as CBPN and TBPN. Several methods to compute the instantaneous firing flow are implemented to analyse the system evolution in different scenarios.

Since its first development in 1993, the tool is in constant evolution to implement new extensions or developed analysis methods. Future development plans include an independent graphical user interface for drawing batches PN models and analyzing the results, adding more features such as other extensions of batches PN, make Simuleau input file compatible with other PN tools and, finally, enlarge compatibility of Simuleau with several operating systems.

REFERENCES

Audry, N., Demongodin, I., and Prunet, F. (1994). Modelling of high throughput production lines by using generic models described in batches Petri nets. In *IEEE Int. Conf. on Robotics and Automation*, (1) 807–812.

Audry, N. and Prunet, F. (1994). Controlled batches Petri nets. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, 1849–1854.

Caradec, M. and Prunet, F. (1997). Coloured batches Petri nets. *IFAC/IFIP Conf. on Management and Control of Production and Logistics*, 30(19), 227–232.

David, R. and Alla, H. (2005). *Discrete, continuous, and hybrid Petri nets*, volume 1. Springer.

Demongodin, I. and Prunet, F. (1992). Extension of hybrid Petri nets to accumulation systems. In *Int. Symp. on Mathematical Modelling and Scientific Computing.*

Demongodin, I. (2001). Generalised batches Petri net: hybrid model for high speed systems with variable delays. *Discrete Event Dynamic Sys.*, 11(1-2), 137–162.

Demongodin, I. (2009). Modeling and analysis of transportation networks using batches Petri nets with controllable batch speed. In *Int. Conf. on Applications and Theory of Petri Nets*, 204–222.

Demongodin, I. and Giua, A. (2010). Linear programming techniques for analysis and control of batches Petri nets. *IFAC Proceedings Volumes*, 43(12), 1–6. 10th WODES.

Demongodin, I. and Giua, A. (2014). Dynamics and steady state analysis of controlled generalized batches Petri nets. *Nonlinear Analysis: Hybrid Systems*, 12, 33–49.

Durmus, M.S., Yildirim, U., and Soylemezk, M.T. (2012). Interlocking system design for ERTMS / ETCS: An approach with batches Petri nets. *IFAC Proceedings Volumes*, 45(29), 110–115. 11th WODES.

Gaddouri, R., Brenner, L., and Demongodin, I. (2016). Controlled Triangular Batches Petri Nets for hybrid mesoscopic modeling of traffic road networks under VSL control. In *IEEE Int. Conf. on Automation Science and Engineering (CASE)*, 427–432.

Giua, A. and Silva, M. (2017). Modeling, analysis and control of discrete event systems: a Petri net perspective. *IFAC-PapersOnLine*, 50(1), 1772–1783.

Hélouët, L. and Thébault, A. (2023). Mochy: A tool for the modeling of concurrent hybrid systems. In *Int. Conf. on Applications and Theory of Petri Nets and Concurrency*, 205–216. Springer.

Júlvez, J., Mahulea, C., and Vázquez, C.R. (2012). SimHPN: A matlab toolbox for simulation, analysis and design with hybrid Petri nets. *Nonlinear Analysis: Hybrid Systems*, 6(2), 806 – 817.

Kaakai, F., Hayet, S., and Moudni, A.E. (2005). Quantitative assessment of travellers connection times into a multimodal hub owing to batches Petri nets. In *17th IMACS World Congress Scientific Computation, Applied Mathematics and Simulation.*

Liu, R., Ammour, R., Brenner, L., and Demongodin, I. (2020a). Event-driven control for reaching a steady state in controlled generalized batches Petri nets. *IFAC-PapersOnLine*, 53(4), 180–186. 15th WODES.

Liu, R., Ammour, R., Brenner, L., and Demongodin, I. (2020b). On/off control trajectory computation for steady state reaching in batches Petri nets. In *VECOS 2020*, 84–99. Springer.

Liu, R., Ammour, R., Brenner, L., and Demongodin, I. (2023). On/off control for reaching a steady state attractive region in batches Petri nets. *IFAC-PapersOnLine*, 56(2), 9618–9623. 22nd IFAC World Congress.

Silva, M., Júlvez, J., Mahulea, C., and Vázquez, C. (2011). On fluidization of discrete event models: observation and control of continuous Petri nets. *Discrete Event Dynamic Systems*, 21(4), 427–497.

Svadova, M. and Hanzalek, Z. (2003). Modeling of systems with delays using hybrid Petri nets. *IFAC Proceedings Volumes*, 36(18), 177–182. Conf. on Control Sys. Design.

Wang, Y. and Zhou, C. (2004). Fluid-based simulation approach for high volume conveyor transportation systems. *Sys. Science and Sys. Eng.*, 13(3), 297–317.