36th European Modeling & Simulation Symposium, 001
21th International Multidisciplinary Modeling & Simulation Multiconference

2724-0029 © 2024 The Authors.
doi: 10.46354/i3m.2024.emss.001

Transformation of BPMN models of business processes into models using colored Petri nets

David Le[1,*], Antonín Kavička[1]

[1]University of Pardubice, Studentská 95, Pardubice, 532 10, Czech Republic

[*]Corresponding author. Email address: david.le@student.upce.cz

Abstract

Modeling business processes followed by their simulations is crucial in running efficient and effective businesses. However, the whole analysis process can sometimes be very complex and time-consuming. A new methodology called *PetriBPMN* has been proposed to simplify this process. The methodology describes the entire process of how to automatically transform initial models defined in reduced formalism *BPMN-light* into CPN (Colored Petri net) models. CPNs enhance *BPMN-light* models by incorporating color annotations, enabling the representation and simulation of complex system states and resource allocation. The transformation process involves mapping BPMN elements to CPN constructs while preserving the semantics of the original model. This approach provides a bridge between the intuitive visual representation of BPMN and the analytical power of CPNs, enabling a more thorough understanding of process dynamics. Real-world case studies were processed to evaluate the practical usability and speed of comprehension facilitated by the new reduced *BPMN-light* formalism, which minimizes the learning curve by using a minimal set of core elements in BPMN specification. As a result, the *BPMN-light* converter web application following the *PetriBPMN* methodology was created to construct, edit, and transform *BPMN-light* files. This enables people in the industry to convert multiple XML files of the *BPMN-light* diagram into XML files of the CPN model easily and fully automatically.

Keywords: Business Process Model and Notation; Colored Petri Nets; process modeling; simulation

1. Introduction

In the rapidly evolving area of modern business, staying competitive requires a constant commitment to process enhancement. Business process modeling has emerged as a vital tool for effective management and adaptation. As processes become increasingly complex, adopting a systematic approach to identify areas for enhancement is crucial.

Streamlining processes through the application of business process modeling not only reduces operational costs but also significantly improves resource allocation and time management.

By simulating modeled processes, organizations can test various scenarios and thoroughly analyze the impact of potential changes before actual implementation in the real world. This predictive approach allows for a good understanding of possible outcomes, enabling more informed decision-making and strategy development. This method allows businesses to identify optimal solutions and adjustments, minimizing risks and maximizing efficiency.

This paper presents a new straightforward and comprehensive methodology called *PetriBPMN* for modeling initial business processes using a user-friendly formalism, specifically the Business Process Model and Notation (BPMN), which is then automatically transformed into Colored Petri Nets (CPNs) for better suitability in conducting simulations and analyses. The primary objective is to devise a simple technique for modeling and analyzing business processes that share resources via simulation experiments and evaluations. To aid individuals unfamiliar with the whole BPMN specification, a simplified version, termed *BPMN-light*, has been designed to facilitate the modeling of input business processes. Colored Petri nets are employed for both static and dynamic process analysis. (Le, 2023)

2. State of the art

BPMN is widely used for its intuitive and comprehensive graphical notation, making it excellent for modeling business processes. However, BPMN lacks the capability for direct simulation of these models, necessitating the use of an alternative modeling formalism for simulation purposes. CPNs are favored for their ability to model and analyze complex systems with concurrency and synchronization.

Despite the complementary strengths of BPMN and CPN, the lack of an automated conversion process means that transforming a BPMN model into a CPN model requires manual intervention, which can be time-consuming and error-prone.

This paper addresses this gap by proposing a methodology for automatically converting simplified BPMN models into CPN models, facilitating the seamless creation of simulations and enhancing the efficiency and accuracy of business process analysis. (Von Rosing et al., 2015), (Jensen & Kristensen, 2009)

3. Reduction of BPMN specification

BPMN is an extensive and complex standard that covers many process modeling issues. A minimalist set of symbols and elements was introduced to create an elementary graphical language for modeling business and business processes, even for users with no or little experience in BPMN. This simplifies the creation of input process models, which are then easily understandable by all participants.

The reduced BPMN-light set comprises only the most fundamental elements for describing the process flow and representing data and physical resources in competitive process modeling, where shared resources are a factor. ("Business Process Model and Notation (BPMN)", 2013)

List of supported BPMN elements in BPMN-light:

- Empty Start Event
- Empty End Event
- Undefined Task
- Data Store
- Exclusive/Inclusive/Parallel Gateway
- Sequence Flow
- Data Flow
- Text Annotation
- Group

4. PetriBPMN methodology

The newly proposed methodology *PetriBPMN* describes the whole process of transformation of business process models using reduced elements set *BPMN-light*. The methodology consists of four main phases:

1. Modeling of input models of business processes in *BPMN-light* formalism.
2. Deserialization of *BPMN-light* models in XML format to *BPMN-light-graph* models
3. Transformation of *BPMN-light-graph* models into *CPN-graph* representation
4. Serialization of *CPN-graph* model into output *CPN-xml* model.

Each phase will be described in detail in the following subsections. These four phases, with their respective sub-steps, are depicted in Figure 1 below.

4.1. Modeling of input models

The methodology begins with creating a BPMN model of the process using the Camunda Modeler editor. While the editor offers modeling capabilities, some features are not supported in the new reduced BPMN-light set. Validation rules are, therefore, in place to ensure that the model meets all necessary criteria, allowing for immediate detection of any deficiencies or errors during the modeling phase prior to transformation into a Petri net. The output of this phase is a validated XML file of the process model in the *BPMN-light-xml* format, making it compatible for transformation purposes.

4.1.1. Validation rules

Developers of the Camunda Modeler have implemented a set of basic validation rules to check for common issues and ensure compliance with recommended rules for creating clean models. For this project, a subset of these rules was adopted. The names of the rules are mostly self-describing. Complete documentation of these rules can be found in the official documentation.

List of adopted validation rules:

- end-event-required
- fake-join
- no-complex-gateway
- no-duplicate-sequence-flows
- no-gateway-join-fork
- no-implicit-split
- single-blank-start-event
- start-event-required
- superfluous-gateway

Additional validation rules had to be created for modeling processes that meet the requirements of the new BPMN-light methodology.

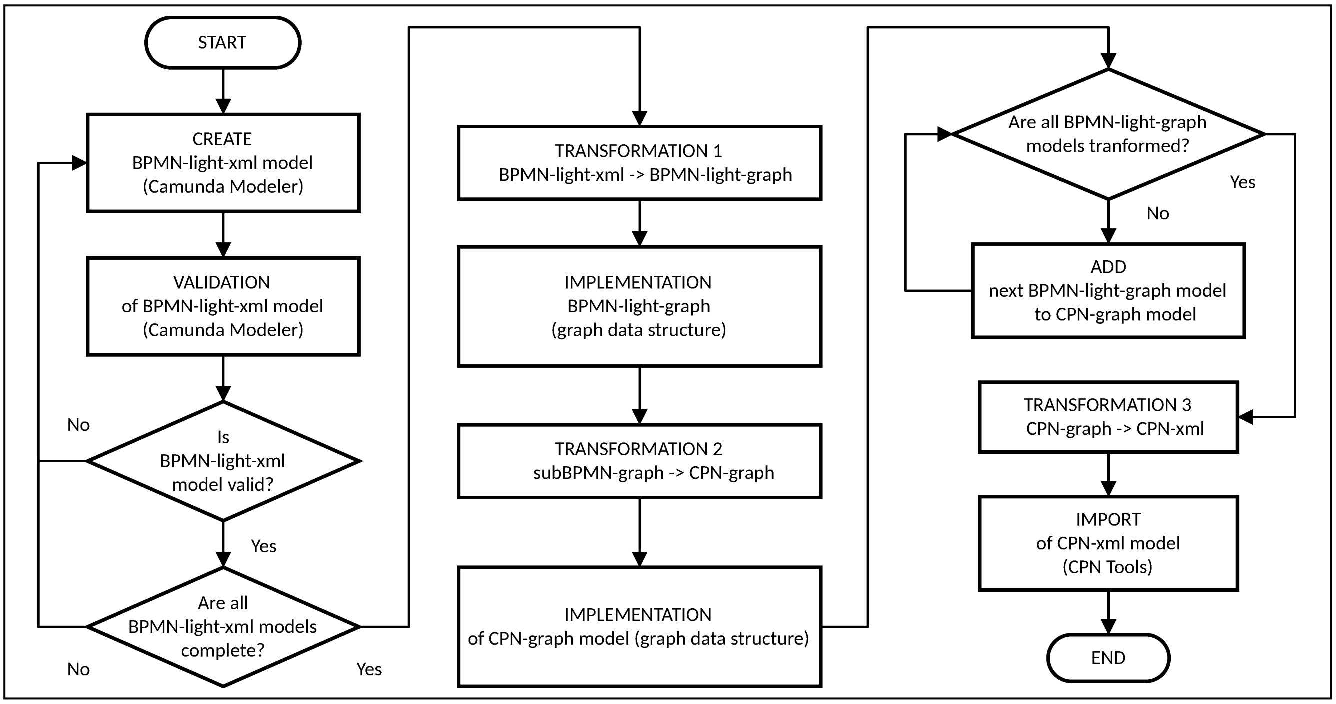

Figure 1. Flow chart representing the newly proposed methodology of *PetriBPMN*.

The newly implemented rules can be divided into two categories. The first category of rules checks that the model does not contain any unsupported elements but only elements supported by the reduced BPMN-light set mentioned before. The second, more important category of rules checks the semantics and the presence of additional information necessary for the PetriBPMN methodology, particularly in the transformation phase. This includes checking for forking conditions on sequence flows and resource quantity allocation on data associations.

List of semantic and additional validation rules:

- label-required
- no-disconnected
- forking-conditions
- no-duplicate-flownode-label
- quantity-on-data-associations

4.2. Deserialization of BPMN-light models

The second step involves deserializing the exported *BPMN-light-xml* model from the previous step for further processing. This XML file adheres to BPMN format standards, enabling partially automated deserialization without custom methods.

The process yields a set of classes that do not yet represent the graph data structure. Therefore, a traversal of elements is necessary to build this structure, resulting in a *BPMN-light-graph* model representing the initial process model.

4.3. Transformation into CPN-graph structure

The transformation between two formalisms is the most extensive and most challenging phase implementation-wise. Ensuring the correct transformation of elements with additional annotations and descriptions is crucial. This process was mostly taken from (Dechsupa et al., 2018) and (Ramadan et al., 2011).

Before transforming each BPMN element into its counterpart in CPN, there is a need to find gateway pairs. This step is essential for mapping the inclusive OR gateway and sequence flows representing correct splitting, merging, and synchronization behavior. This process was taken from a source (Dechsupa et al., 2018), which deals with model partitioning using gateway pairs.

PetriBPMN methodology also supports transforming multiple input BPMN processes into one hierarchical CPN model. This enables users to merge many models that share resources throughout processes. These shared resources are transformed into fusion places representing one shared resource pool between processes.

Transforming more than one *BPMN-light* model into the *CPN-graph* model results in the colored Petri net, where each page represents one input model.

4.4. Serialization of transformed model

The last step is to serialize the transformed *CPN-graph* model into the *CPN-xml* model, an XML file compatible with the CPN editor.

CPN Tools was chosen as a target of transformation and serialization because of its suitability as a simulation tool. It supports working with colored and timed Petri nets and offers advanced analytical features such as reachability analysis, state space exploration, and model simulation.

Serialization to a file was implemented similarly to the BPMN process of deserialization. CPN Tools also has an XSD specification defining its XML file structure. The *CPN-graph* structure is transformed, mapped to XML format, and then saved to the output file that can be directly imported to the CPN Tools.

5. Case study

To explain and demonstrate the use of a newly created methodology, two models are being introduced: vehicle repair shop and gas station. The vehicle repair shop and gas station work closely together as the gas station provides washing facilities for its customers and locally serviced cars. This means the washing facilities are shared between the repair shop and the gas station.

By modeling this collaborative setup, examining the efficiency, resource allocation, and potential bottlenecks in such an integrated model is possible.

The first model is the company car service (Figure 2), which has workshops for repairing various vehicles and providing extra services such as applying advertising stickers on the car body or cleaning the vehicle's interior.

The process starts with the vehicle's arrival at the workshop. The vehicle is then moved to the appropriate workshop based on its type - trucks go to the service technician, cars with faults are inspected and repaired, and buses are not serviced. After that, the vehicle is washed, and the arrival report is written. Finally, the vehicle can leave the service or opt for additional services like adhesive coating and interior cleaning.

The second model is a petrol station model (Figure 3), which provides refueling of conventional fossil fuels such as petrol, diesel, or LPG. It also offers body washing services by providing automated washing lines.

The process, in this case, is very trivial. When the vehicle arrives, a decision is made as to whether the vehicle is refueled with LPG or not. When refueling with LPG, the vehicle must arrive at the LPG dispenser. Gasoline refueling stands are used to refuel petrol and diesel. After refueling, body washing can be used, which takes place in the washing line. After refueling and possible body wash, the vehicle leaves the pump station area.

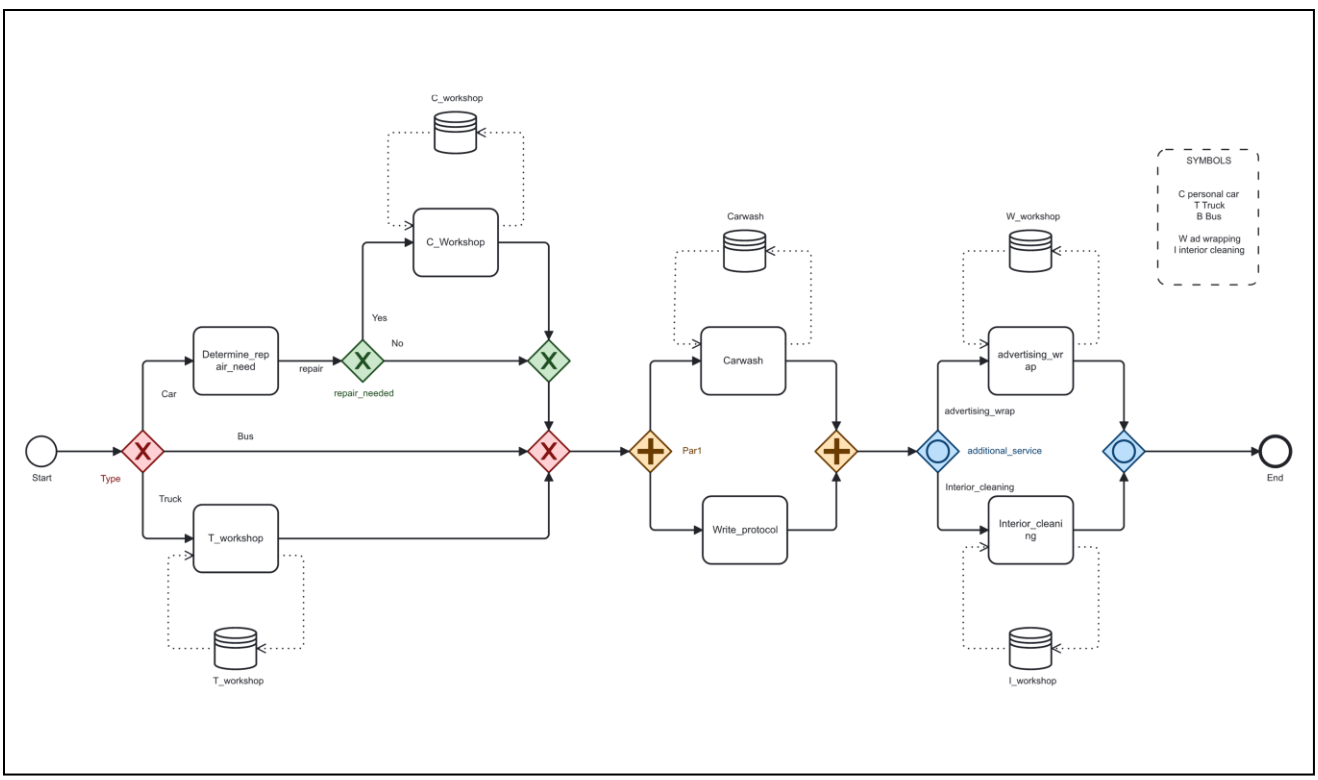

Figure 2. BPMN model of the repair shop

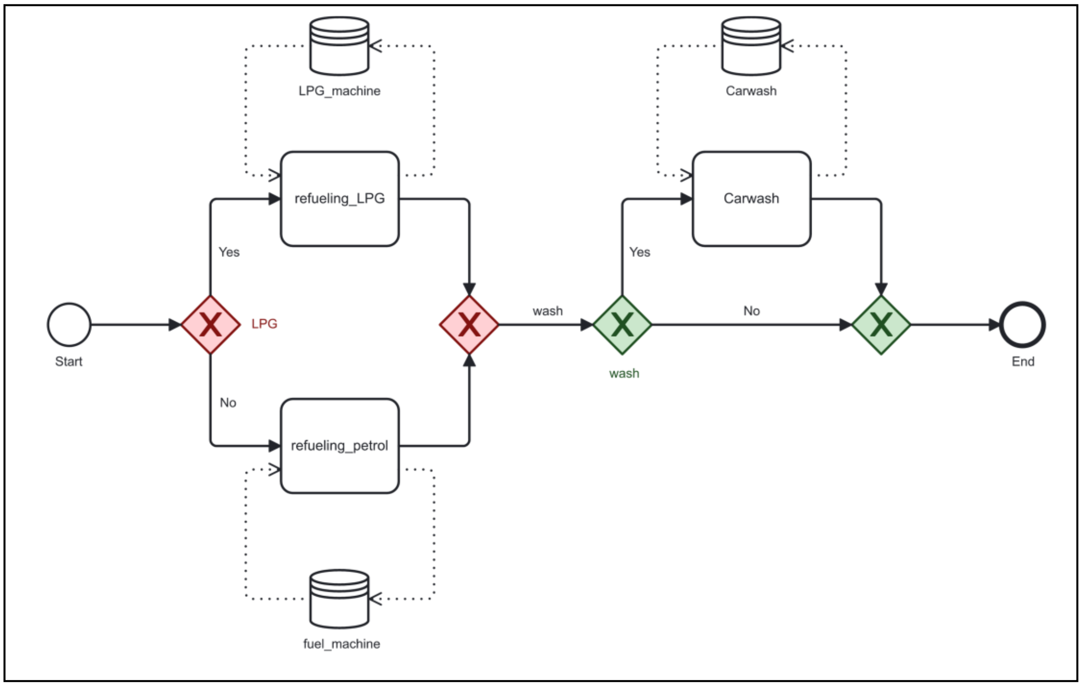

Figure 3. BPMN model of the gas station

5.1. BPMN-light model transformation

After modeling both processes using BPMN-light formalism, the next step was to convert them into colored Petri net models for further simulations and analysis.

This conversion was achieved using a newly developed web application that can take one or more BPMN-light compatible models and produce a single transformed CPN model. This fully automated process results in a comprehensive Colored Petri net structure that accurately represents the input models as Petri nets, including functional transitions, places, arcs, and arc inscriptions with all necessary declarations.

5.1.1. Adding time component to the model

The resulting Colored Petri net is untimed and does not contain time information. This model can be used only for statistical analysis such as state space, reachability, or liveness analysis. Supplementing the process model with a time component to describe dynamic systems is necessary. By introducing timed colored Petri nets, it is possible to perform full-scale simulations and analyses to explore additional characteristics such as wait times, queue lengths, etc.

Before conducting any simulations on the transformed model, timed color set declarations, functions, and arc inscriptions had to be added to the model. This involved incorporating time delays into transitions to represent the duration of the underlying activity. Also, some places were modified to support timed nets.

By adding a time element into the model, the behavior of the net is also changed. The main difference is in transition enabledness. (Jensen & Kristensen, 2009)

Secondly, by having a timestamp on the token itself, it is necessary to consider the ordering of the tokens at places. If a place contains more than one token that can be used for free variable binding (it is color-enabled and timestamp-enabled), any of these tokens can be used for binding. However, this behavior is not desirable in this case.

Arriving tokens should be ordered by their timestamps to form the FIFO (first in, first out) queue. This will enable the model to prioritize token selection for binding, which will adhere to a correct queuing system for this model.

5.2. Parametrization of simulation experiments

Three simulation scenarios were created: Sc01, Sc02, and Sc03, which applied different numbers of Carwash stations. Table 1 shows the global parameters of simulations.

All scenarios share the same event probability settings in Table 2, (transition) delays in Table 2, and entity arrival distribution settings in Table 3.

Table 1. Global parametrizations of simulation experiments

Value Name	Value
Number of replications	500
System open time	12 h (720 min.)

System open time is the window between open time and closing time for the last entity to arrive in the system.

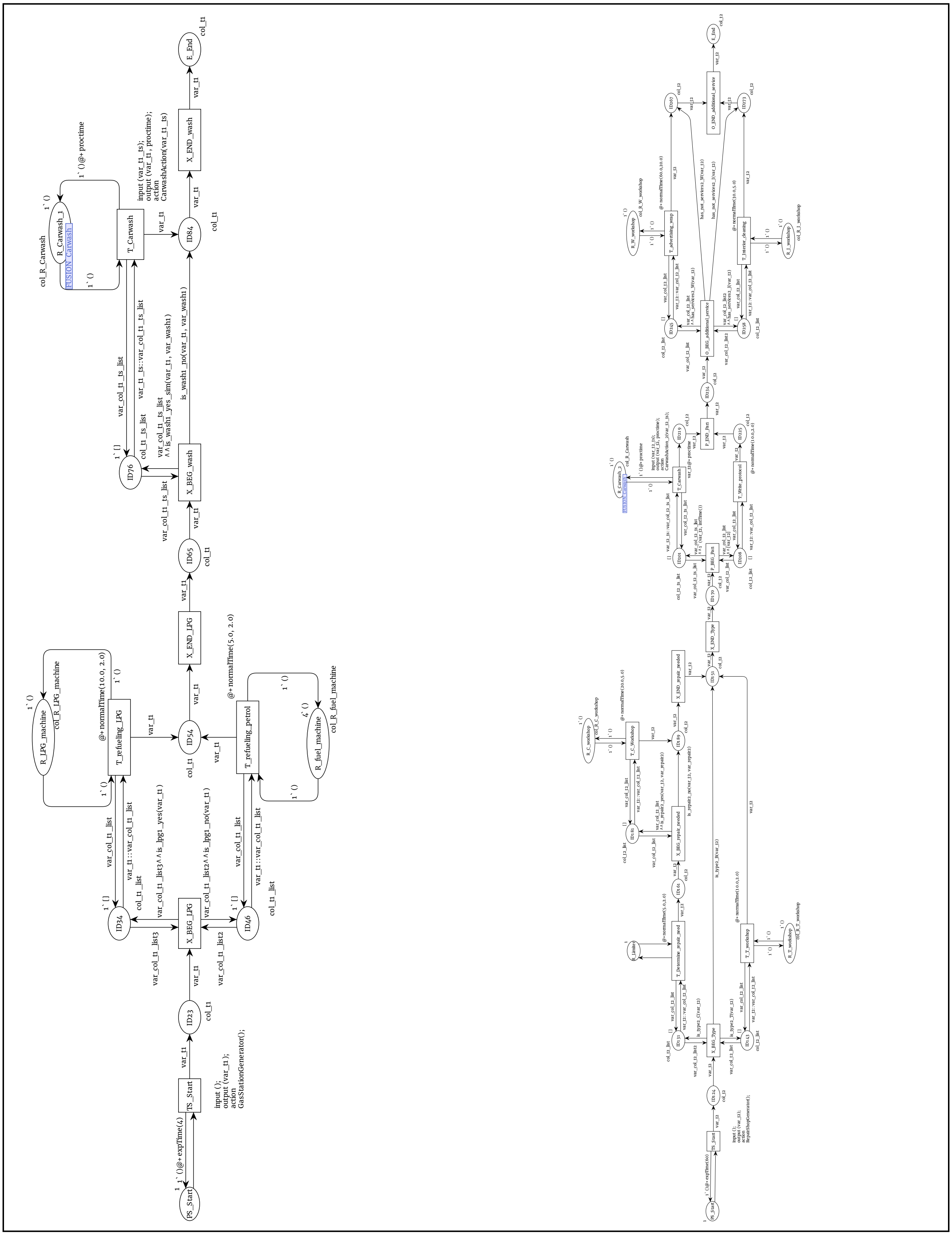

Figure 4. Transformed models of Gas station and Repair shop in CPN formalism with time component and FIFO queue system

Table 2. Event and randomization probability settings

Value Name	Probability
LPG refueling	0.2
Petrol Refueling	0.8
Vehicle type - Car	0.7
Vehicle type - Truck	0.2
Vehicle type - Bus	0.1
Carwash service at repair shop	1.0
Carwash service at gas station	0.5
Additional services – advertisement wrapping	0.2
Additional services – interior cleaning	0.4

Probability values are from the range of 0.0 to 1.0

Table 3. Interarrival times and transition delay distribution settings

Name	Distribution type	Parameters
Gas station arrivals	exponential	$\lambda = 4$
Repair shop arrivals	exponential	$\lambda = 60$
T_Refueling_petrol	normal	$\mu = 5, \sigma^2 = 2$
T_Refueling_LPG	normal	$\mu = 10, \sigma^2 = 2$
T_Carwash	normal	$\mu = 12, \sigma^2 = 4$
T_T_Workshop	normal	$\mu = 10, \sigma^2 = 2$
T_C_Workshop	normal	$\mu = 20, \sigma^2 = 5$
T_Determine_repair_need	normal	$\mu = 5, \sigma^2 = 2$
T_Write_protocol	normal	$\mu = 10, \sigma^2 = 2$
T_Interior_cleaning	normal	$\mu = 30, \sigma^2 = 5$
T_Advertising_wrap	normal	$\mu = 60, \sigma^2 = 20$

Rows with the prefix "T_" mean transitions visible in Figure 4

The exponential distribution function was used to generate input flow to simulate the time between vehicle arrivals in the system. For transition delays, the normal distribution with different parameters was used. Both distributions generate times in minutes.

Table 4. Shareable resource capacity setup common for all scenarios

Resource Name	Value
R_LPG_machine	1
R_fuel_machine	4
R_T_workshop	1
R_C_workshop	1
R_W_workshop	1
R_I_workshop	1
R_I_workshop	1

6. Results and Discussion

The simulations were done using a transformed model shown in Figure 4. This model was set up with values as in Table 1, Table 2, and Table 3. The model was rerun between each scenario with the different number of Carwash stations depicted in fusion places R_Carwash_1 and R_Carwash_2. Other resources were kept the same, as shown in Table 4.

The most important characteristics and indicators were "mean wait time" in the queue at the shared carwash stations and their "utilization".

Two monitors were created in CPN Tools to capture needed information and statistical data, one for each model sharing the carwash resource.

The monitors captured the wait time for each token in place, representing the queue before the carwash station(s) and the processing time needed to complete the task of the token at the carwash station itself.

The results of the simulations for each experiment scenario are described in Table 5. This table contains the "mean" and "margin of error" calculated for the confidence interval with confidence of 95%.

Table 5. Results of the simulation experiments for each scenario

Scenario	SC01	SC02	SC03
Number of Carwash stations	1	2	3
Mean wait time (Gas station model)	168,81 min	13,19 min	1,56 min
MOE of wait time (Gas station model)	3,10 min	0,85 min	0,09 min
Mean wait time (Repair shop model)	25,30 min	6,22 min	1,21 min
MOE of wait time (Repair shop model)	1,15 min	0,27 min	0,08 min
Mean of utilization per carwash station	99,77 %	83,94 %	57,15 %
MOE of utilization per carwash station	0,10 %	0,65 %	0,46 %

MOE – margin of error in same units as Mean. The resulting confidence interval is calculated as [Mean – MOE; Mean + MOE]

Based on the above results, the setup of scenario SC01 is not recommended due to the high waiting time, which indicates a growing queue over time.

The results are significantly better in scenarios SC02 and SC03, indicating improved system stability and usability. The wait time in scenario SC03 is very low, though the Carwash station's utilization is only about 50%. When deciding between these scenarios, it is crucial to weigh the importance of these performance indicators according to the specific priorities and goals of the system.

7. Conclusions

The *PetriBPMN* methodology effectively bridges the gap between BPMN's intuitive visual modeling and the analytical capabilities of CPNs, allowing people in the field to analyze and create models in simplified *BPMN-light* formalism and easily convert and transform these models into CPNs automatically without manual transformation.

Through this transformation, creating a more detailed and dynamic analysis of business processes is possible, facilitating improved decision-making and optimization.

However, as mentioned above, it is currently impossible to create full simulations without modifying the transformed CPN model. It would also be desirable to create an automated solution for creating timed CPNs with an automated scenario creation. These scenarios could then be tested and simulated to find an optimized model without manually setting the parameters of the model.

Acknowledgments

The Internal Grant Agency of the University of Pardubice supported the research within the project SGS_2024_011.

References

Business Process Model and Notation (BPMN). (2013). Object Management Group. https://www.omg.org/spec/BPMN/2.0.2

Dechsupa, C., Vatanawood, W., & Thongtak, A. (2018). Transformation of the BPMN Design Model into a Colored Petri Net Using the Partitioning Approach. IEEE Access, 6, 38421-38436. https://doi.org/10.1109/ACCESS.2018.2853669

Jensen, K., & Kristensen, L. M. (2009). Coloured Petri Nets: Modelling and Validation of Concurrent Systems (1st ed.). Springer Berlin Heidelberg. https://doi.org/10.1007/b95112

Ramadan, M., Elmongui, H., & Mansour, R. (2011). BPMN Formalisation using Coloured Petri Nets. ResearchGate. Retrieved May 9, 2023, from https://www.researchgate.net/publication/268364122_BPMN_Formalisation_using_Coloured_Petri_Nets

Von Rosing, M., Scheer, A. -W., & Von Scheel, H. (2015). The complete business process handbook: body of knowledge from process modeling to BPM. Morgan Kaufmann.

Le, D. (2023). Transformace BPMN modelů technologických procesů do modelů využívajících barvené Petriho sítě [Diploma thesis, University of Pardubice, Faculty of Electrical Engineering]. https://dk.upce.cz/handle/10195/81566