# Robotic Sorting Line Model using Coloured Petri Net

Dominik Stursa[1[0000-0002-2324-162X]], Libor Havlicek[2[0000-0001-6874-5971]], Libor Kupka[3[0000-0001-7750-2824]]

Faculty of Electrical Engineering and Informatics, University of Pardubice, Studentska 95, Pardubice 53210, Czech Republic
[1] dominik.stursa@upce.cz
[2] libor.havlicek@upce.cz
[3] libor.kupka@upce.cz

**Abstract.** With the increasing availability of robots and image processing systems, the automated robotic line implementations are still extensively increasing. As well, the complexity of these systems increases, which increases the demands on the determination of their function and possible analysis. For these purposes the complex robotic systems are modeled. Here, the Coloured Petri Nets were used to model the robotic sorting line. Specifically, the CPN Tools software was used to model creation, simulation, analysis and to made experiments. State space analysis was done to revelation of possible unwanted deadlocks. Furthermore, the sorting line model was used to found out the maximum number of objects that can be sorted in a defined time. As the robot manipulation time directly affects maximum number of objects in one batch, the timing procedure was created to declare robot operation dependency on types of manipulated object. The state space analysis showed that there are no unwanted deadlocks in the system. The model has been verified and declared as correct. Optimal composition and order of objects was successfully found by timed coloured Petri nets experiments.

**Keywords:** Colored Petri Net, Sorting Line, Modeling.

## 1 Introduction

With the development of robotic technologies along with computer vision approaches, automated lines are being implemented more often. Such systems are more involved in industrial applications what leads to control system improvements. With the growing number of involved components, the complexity of the control system also increases. Therefore, the state analysis and system behavior description need to defined as it simplifies the design of such system.

Modeling with event systems enjoys growing popularity because of their ability to adequately represent the world around us, especially in fields of business processes [1], production systems [2], networked control systems [3], etc. Therefore, a software for system modeling is often used to analyze system behavior. The scope of this article is the design of robotic sorting line model.

The event model of the robotic sorting line can help with process understanding, especially if the system is comprehensive and with its complexity the possibilities of states analysis is extensively increasing. With the complexity is also growing the possibility of the unwanted system deadlocks.

Authors' decides to use the coloured Petri net as a modeling tool, because it provides a mathematical apparatus for the possible solution and analysis of the state space of any model. Furthermore, the construction of the coloured Petri nets is capable enough to prevent unplanned deadlocks as is indicated in [4] or [5].

### 1.1 Motivation

The main motivation for using of the Coloured Petri Nets (CPN) is the easy creation of the state model for complex system, which can be simulated, analyzed, verified and visualized. The CPN concepts are precisely defined and clearly described in the literature [6] and [7], what makes a CPN modeling more accessible.

Furthermore, the CPN models can be mathematically described, which is allowing complex model state space analysis [8]. Moreover, there exist a lot of software for CPN model creation as is JARP [9], CPN Tools [10]. Possibilities of accurate system description by modeling is increased with use of CPN or their further improvements.

## 2 Problem definition

The general automated robotic sorting systems are consisted of a several parts executing more or less important tasks. As first, the object identification must be implemented for distinguish objects that need to be sorted. Next, the control system must evaluate and calculate object positions at moment in which robot will try to pick and place object to the correct spot. As other, the selection strategy must be also defined to provide fast and accurate sorting. The robotic sorting line is captured on Fig. 1.

Sorting line could consist of a lot of different configurations. At first, the number of object types can be various. With increasing number of object types, the demands on the end effector of the robotic manipulator also increase. This is due to the reduced probability of being able to grab an object by only one effector with increasing differences in the shape of the objects. Furthermore, the number of conveyor belts can also increase when combining multiple production technologies [11].

The number of required components of robotic sorting line increases considerably with the complexity of the sorting line. Definition of the specific problem that we dealt with is stated in the next section.

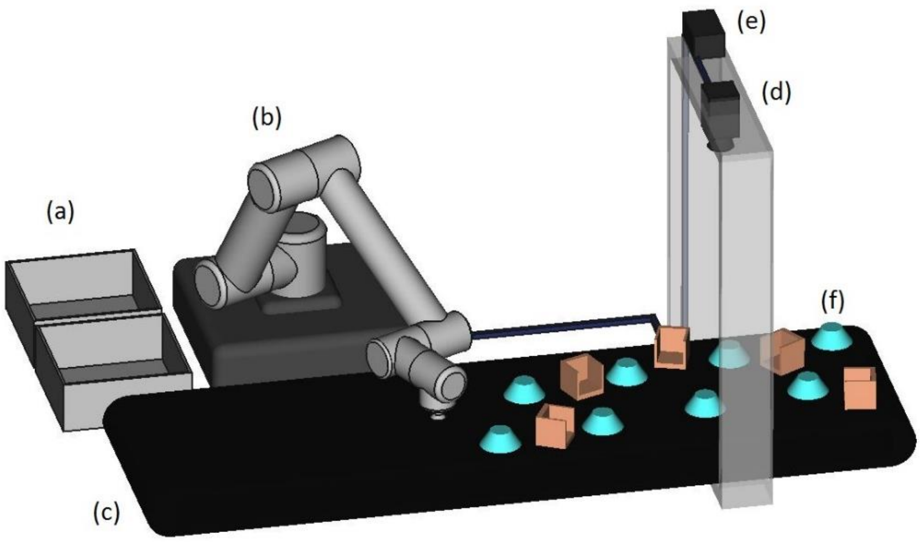


**Fig. 1.** Robotic sorting line illustration. Illustration contains two storage boxes (a), robotic manipulator (b), conveyor belt (c), camera sensing system (d), control system (e) and two types of object (f).

### 2.1 Problem statement

Problem of general sorting line is not considered here, because its solution leads to a very long duration of calculations. As such, problem was limited to the specific composition of the robotic sorting line. Specifically, the composition using two conveyor belts and three robotic manipulators were used for five object types sorting.

There, these five types of objects are produced in one production. The reason for this two-belt composition is the parallel production of components of different types in two different plants. Each of robot is equipped with a different end effector and therefore can only manipulate some object types. This particular robotic sorting line composition is illustrated in Fig. 2.

The function of presented system can be described as follows. Each conveyor belt has a dispenser on the starting side, that dispenses batches of an object mixture. When the batch is released, the conveyor starts to move and transports objects mixture under the scanner, where each object in the whole batch is classified and localized. The batch is moving on the conveyor belt toward robots, which are gradually relocating objects to its boxes till no object remains on the belt. The next batch is released after successful sorting of objects into corresponding boxes from the previous batch. Presented case of the robotic sorting line was used for CPN model creation, which is described in next section.

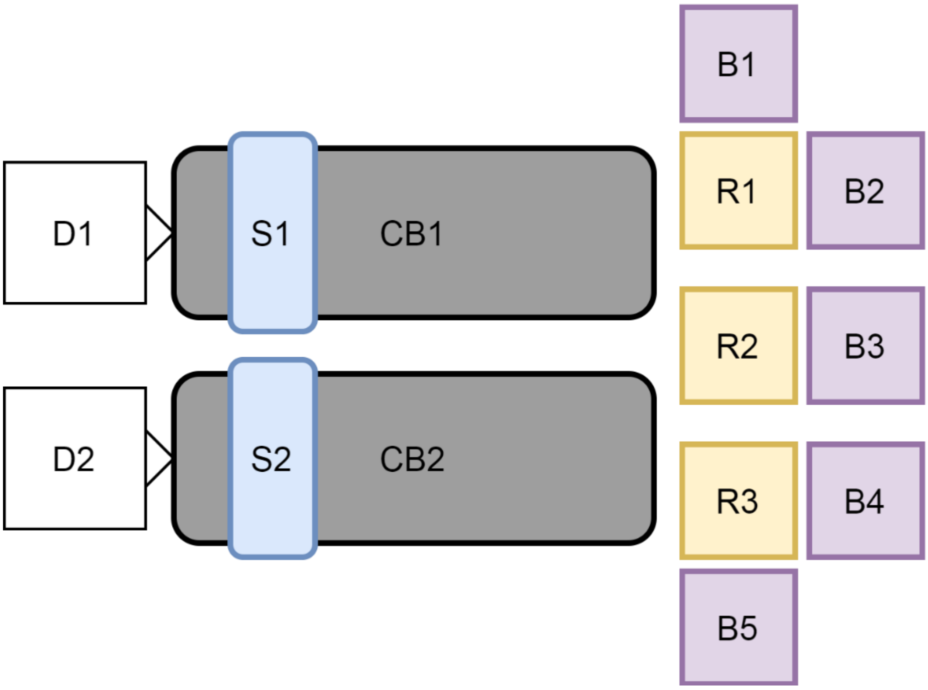


**Fig. 2.** Specific robotic sorting line layout.

### 2.2 Coloured Petri net model creation

Coloured Petri net construction is a multistage process, where at least the places, transitions, edges, variables, guards and colors must be defined. Each place is defined by specific color type and can contain tokens in the same color as its type. Edges can transfer these tokens within defined conditions. When each input connection contains proper tokens, the transitions can fire (be triggered) whatever type of tokens through any output connection. The transfer of tokens is controlled by restrictive conditions on edges and transitions, which makes the correct token flow through the model.

As a modeling software was used CPN Tools [10], which allows a variety of automated simulation and analysis options. For purposes of our model was needed do define specific color types and variables for possible transition of colors through model arcs. The definition of the color sets and variables is written in following CPN Tools program code:

```
Standard declaration():
      colset OBJECT = with O1| O2| O3| O4| O5;
      colset OBJECTList = list OBJECT;
      colset INT, NO = int;
      colset BATCH = product NO * OBJECTList;
      colset PROCESS = product INT * OBJECT;
      colset BOOL = bool;
```

```
       var obj: OBJECT;
       var olist: OBJECTList;
       var n: NO;
       var e: INT;
       var b: BOOL;

       fun em(olist) =
           if lenght olist = 0
             then true
           else false;
end.
```

It can be seen from the CPN Tools standard declaration program that seven colors, five variables, and one conditional function have been defined. As these declarations are made, then the model could be presented as its on Fig. 3.

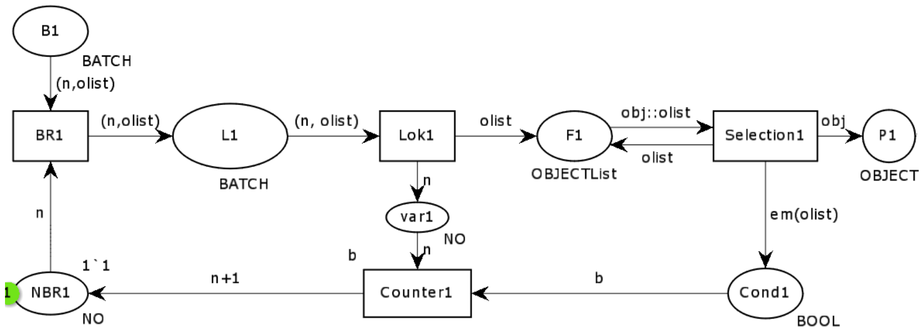


**Fig. 3.** CPN tools model of particular line behavior and object consumption.

The model presents the particular line behavior and object consumption process. At first, the place marked as B1 is defined as color set of BATCH type. From program declarations is obvious, that this color is product of colors NO and OBJECTList. It means that, the tokens in this place are numbered batches of objects. The color NO stands for number, which correspondents to the order of the batches. The color OBJECTList stands for batch, which is ordered set of object mixture.

The place marked as NBR1 serves as the counting token holder, which is conditioning the passing of the correct batch through the transition BR1. The number and batch tuple are then moved to the locator, which separates the number and batch. The batch is then shifted as the list of objects to the part where the gradual selection of objects is performed. This this is ensured by relation between elements F1 and Selection1. Objects are then handed over through the place P1 for manipulation by the robots. If the object list is empty after the selection, then the counter is incremented, which executes the next batch processing.

As such, the purpose of this model is to select batch, separate objects and shift them for further processing with robots. This particular model is ended by place P1, which is the first place of robot function model shown in Fig. 4.

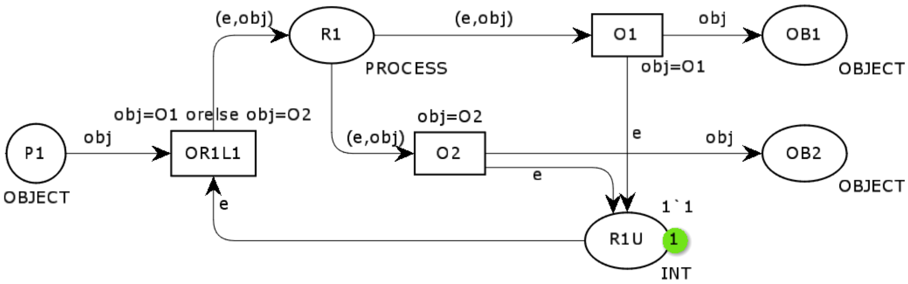


**Fig. 4.** CPN tools model of particular robot function for object placing into boxes.

On this particular model is showed only function of the first robot, which can manipulate O1 and O2 types of objects. The place P1 is passing objects of any type to next transitions. In this case it is the transition OR1L1. The place R1U contains token, which is signalizing that, the robot is available for manipulation task. Accordingly, to the actual robot state, the OR1L1 transition can fire. If so, the place R1 get token composed of number and object colors tuple. From place R1 leads connection to transition O1 and O2. When the type of object is O1, then the transition O1 fires object token to place OB1, which indicates that the object is placed in the box, and number token to place R1U, which shows the release of the robot resource. In case of the second branch, the workflow works the same, but with object type O2.

As the robotic line was created from more similar parts defined in previous models, the entire sorting line model with two lines (model on Fig.3.) and three robots (model on Fig. 4.) is assembled into a whole, which is captured on Fig. 5.

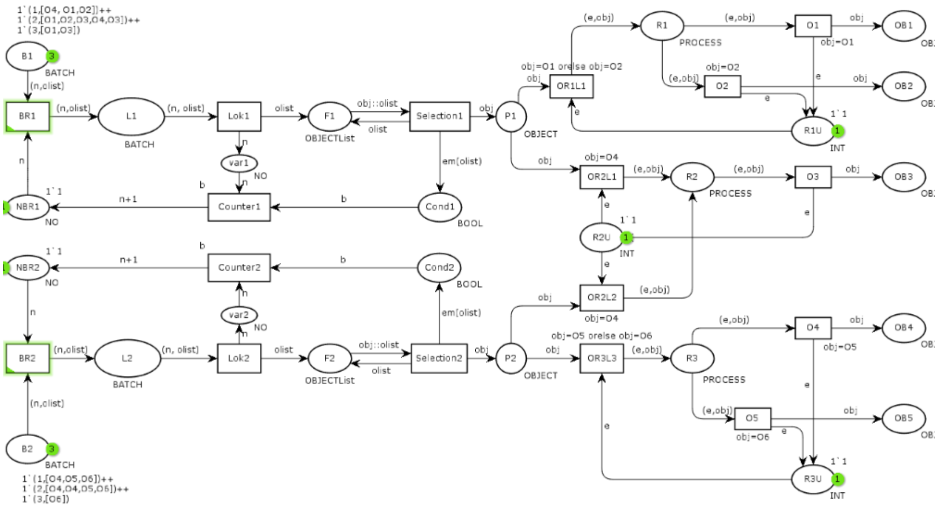


**Fig. 5.** CPN tools model of entire robotic sorting line

Function of complete system model should be described as follows. On each of line is processed selection of objects from batch. Each manipulated object takes its robot resource and after successful manipulation resource is released, which will allow the robot to be reused. If all objects in particular batch are placed to its boxes, then the next batch is released.

The model was extended by adding the time to color sets definitions and is necessarily described in following section.

### 2.3 Timed coloured Petri net model

For further analysis of the dependency between batch processing time and composition of objects and their counts in the mixture, the coloured Petri net model was extended with timing. The places, where robots are manipulating objects got the time consumption stamp, which adds some duration to the operations as it works like in the real process. The duration time of each batch processing can be used for calculations for optimal queuing of objects in a batch.

In addition to coloured Petri net model, the check was placed on the placement of all objects in their boxes, which could trigger the release of another batch.

### 2.4 Experimental procedure

The experimental procedure was composed from three different experiments. As first, the experiments with model proper function were done by repeated simulations with randomly generated initial batches. As second, the state space analysis was done for the same simulations as in previously mentioned experiment. Proper model function and state space analysis was tested on 100 simulation experiments.

Finally, the experiments with timing were performed. The time consumption for each object type was defined. After that, one specific batch composition, with only a given number of objects, was selected for repeated simulations in which the order of the taken objects changed. One batch composition was tested with 1000 object order changes. Total processing time was compared for each of these 1000 object orders, and best order composition was then marked as optimal. Totally, the 50 batches compositions were tested. A simplified illustration is given for a better idea on Fig. 6.

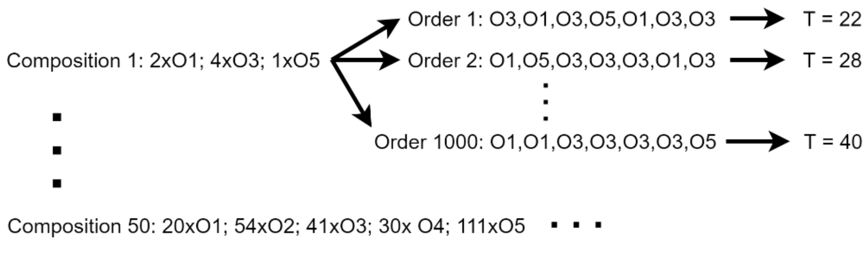


**Fig. 6.** Simplified illustration of experiments with batch orders and time consumption

## 3 Results

The model proper function was tested at first. Each of the 100 simulation experiments ended up in a state where all the objects from the batches were sorted in the appropriate boxes. The result of this experiment can be considered successful. The proposed model of the colored petri net was functional according to the requirements and therefore, can be considered as correct.

The CNP Tools provides a state space analyzing tool, which can be used to investigate the behavioral properties of coloured Petri net model. This tool was used to proper analyzation of every state space of a model for each of 100 simulations. Due to the increasing complexity of state space caused by tokens counts, the state space generation for some models lasted several tens of minutes. As an example, an average state space model contained around 15 thousand of nodes, 69 thousand of arcs between these nodes and the state space generation takes 112 seconds.

The analysis also showed, that the number of home markings is the same as the number of dead markings, which can be interpreted in two ways. First, this finding confirms the results of the first experiments. Second, it clearly indicates that, there were no unwanted deadlocks. Furthermore, there were no infinite occurrence sequences, which means that no infinite loops were indicated. The results of second experiment procedure also confirms the correctness of the proposed model.

The timed coloured Petri net used for time consumption measurements provided the resulting time required to sort each specific batch. From the last experiments were verified that, the objects order directly affects time consumption, so the correct sorting of objects for removal by robots should be ensured for each robotic sorting line.

## 4 Discussion

The coloured Petri net model and timed coloured Petri net model of specific robotic sorting line was created. The model was composed of different parts, including two lines and three robots.

With the experiments, the correctness of the model function and proper model behavior was verified. The state space analysis confirmed that, there are no unwanted deadlocks in the model.

Optimal composition and order of objects was successfully found by timed coloured Petri nets experiments. As such, this possible application of coloured Petri nets would be confirmed.

In future work, the modeling of this kind of a system should be done on more general basis, which can help with generalized mathematical analysis of the state space. Moreover, the specific parts of model should be created as a hierarchical model, which allows easier and more natural model extension.

## Acknowledgments

The work has been supported by Funds of University of Pardubice (by project “SGS2021” No: SGS_2021_019), Czech Republic. This support is very gratefully acknowledged.

## References

1. Liu, C., Pei, Y., Cheng, L., Zeng, Q., Duan, H. Sampling business process event logs using graph-based ranking model, (2020) Concurrency Computation
2. Chen, W., Liu, H., Qi, E. Discrete event-driven model predictive control for real-time work-in-process optimization in serial production systems, (2020) Journal of Manufacturing Systems, 55, pp. 132-142.
3. Zhang, X.-M., Han, Q.-L., Ge, X., Ding, D., Ding, L., Yue, D., Peng, C. Networked control systems: A survey of trends and techniques, (2020) IEEE/CAA Journal of Automatica Sinica, 7 (1), art. no. 8766208, pp. 1-17.
4. Huang, B., Zhou, M., Wang, C., Abusorrah, A., Al-Turki, Y. Deadlock-free Supervisor Design for Robotic Manufacturing Cells with Uncontrollable and Unobservable Events, (2021) IEEE/CAA Journal of Automatica Sinica, 8 (3), art. no. 9106876, pp. 597-605.
5. Pomares-Angelino, R., Lopez-Mellado, E. Automated Modelling of Deadlock-free Petri Nets Using Duplicated Transition Labels, (2020) 2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2020, art. no. 9299169
6. K. Jensen and L.M. Kristensen. Coloured Petri Nets — Modeling and Validation of Concurrent Systems. Springer-Verlag Berlin, 2009.
7. W.M.P. van der Aalst and C. Stahl. Modeling Business Processes — A Petri Net-Oriented Approach. The MIT Press, 2011.
8. Christensen, S., Petrucci, L.: Modular state space analysis of coloured petri nets, (1995) Lecture Notes in Computer Science, 935, pp. 201-217.
9. JARP Project, http://jarp.sourceforge.net/, last accessed 2021/02/12
10. CPN Tools – A tool for editing, simulating, and analyzing Colored Petri nets https://cpntools.org/, last accessed 2021/02/12
11. Hino, R., Harada, M.: Development of a transportation system with multiple conveyor belts: Conditions for transportation and scheduling (2012) Seimitsu Kogaku Kaishi/Journal of the Japan Society for Precision Engineering, 78 (12), pp. 1105-1111