RESEARCH PAPERS
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

2019 Volume 27, Number 45

SUPERVISORS OF PETRI NETS

Jana FLOCHOVÁ[1], Tomáš LOJAN[1]

[1]SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA
INSTITUTE OF APPLIED INFORMATICS, AUTOMATION AND MECHATRONICS
ULICA JÁNA BOTTU Č. 2781/25, 917 24 TRNAVA, SLOVAK REPUBLIC
e-mail: jana.flochova@stuba.sk, tomas.lojan@gmail.com
Received 19 August 2019, accepted 12 September 2019, published 29 November 2019

Abstract

The design and operation of modern industrial systems require modeling and analysis in order to select the optimal design alternative and operational policy. Discrete event system models are encountered in a variety of fields, for example computers, communication networks, manufacturing systems, sensors or actuators, faults diagnosis, robotics and traffic. The paper describes principles and methods of supervisory control of discrete event systems initiated by Ramadge and Wonham. Three supervisory control methods based on the Petri net models are introduced, and the key features of the Petri tool software application for the supervisory control of discrete event systems modeled by Petri nets are highlighted.

Key words

Discrete event systems, supervisory control, Petri nets, reachability tree, P-invariant

INTRODUCTION

The growth in the complexity of modem industrial systems poses numerous problems to their developers. The design and operation of such systems require modeling and analysis in order to select the optimal design alternative, and operational policy. Discrete event system models (DES) [7] describe the behavior of a system that evolves in accordance with the sudden and asynchronous occurrence of events. Such systems are encountered in a variety of fields, e.g. computers, communication networks, manufacturing, process control, faults diagnosis, robotics and traffic. The problem of supervisory control of DES was presented in [18-19] and has been studied extensively since then [1-2], [4-6], [8], [10-16], [20-26], [28-29] etc. For a given DES, it is of interest to synthesize a supervisor that prevents the occurrence of undesirable states and that guarantees that certain termination states are reached under conditions where some state-to-state transitions can neither be prevented by any action of the supervisor nor observed in the supervisory control system. The regulatory actions of supervisors illustrated in Figure 1 are based on the plant observations resulting in feedback control.

In the introductory papers [18-19], DES (a plant to be controlled) is modeled as an automaton $G = (Q,\Sigma,\delta,q_0,Q_m)$ that generates a formal language over a finite alphabet Σ, whose elements label the automaton transitions or events. Σ^* denotes the set of all finite strings of elements of Σ, including the empty string. The events labeled by an element in a fixed subset Σ_c of Σ are declared to be controllable, that means they can be disabled by an external controller or supervisor. The events labeled by an element in a fixed subset Σ_u of Σ are declared to be uncontrollable; they cannot be disabled, e.g. machine breakdown. The plant observation is given by the events labeled by an element in a fixed subset Σ_o of Σ whose occurrence can be seen by an external sensor while the rest of events including faults are unobservable silent. The supervisory controller specification does not contradict the behavioral specifications of the plant model, and achieves the maximal permissiveness of the closed loop system within the specification (all events which do not contradict the specification are allowed to happen).

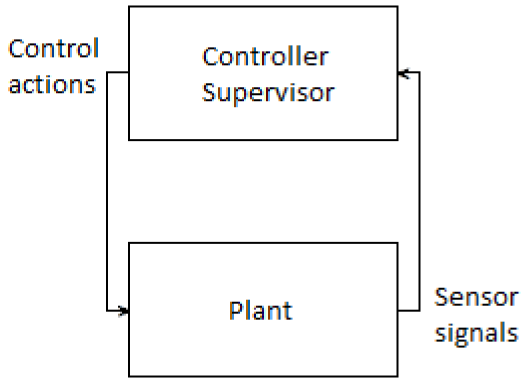

Figure 1 *DES control*

PETRI NETS

We consider DES modeled by Petri nets [3], [17]. Petri nets, as graphical and mathematical tools, provide a uniform environment for modeling, formal analysis and design of DES. Synchronization and concurrency of processes and events can be easily modeled within the framework. The nets are better suited than final automata for representation of systems with repetitive structures and flows. In addition, they provide an appealing graphical representation that makes it possible to visualize the state-flow of a system and to quickly see the dependencies of one part on another. Many reduction and decomposition techniques have been developed for Petri nets; there are procedures for verifying their behavioral and structural properties. Petri nets are directed bipartite graphs in which two sets of nodes are called places and transitions. Places hold tokens which move from one set of places to the other owing to the firing of transitions. The firing represents the occurrence of events in the DES context. Tokens distribution indicates the net state or its marking.

The controlled Petri net is a septuplet:

$$S = (P,T,F,W,C,B,M_0)\ , \qquad (1)$$

where:

1. P and T are the non-empty sets of state places and transitions
2. $F\subset (P\times T)\cup(T\times P)$ is the set of directed arcs connecting state places and transitions
3. C is the finite set of control places, most one per transition
4. $B \subset (C\times T)\cup(T\times C)$ is the set of oriented arcs associated control places with transitions
5. M_0 is the initial marking.

The structure of a Petri net is described by (P,T,$\boldsymbol{D}^+$,$\boldsymbol{D}^-$), where P and T represents the vertices of the graph, and $\boldsymbol{D}^+$ and $\boldsymbol{D}^-$ are integer matrices with nonnegative elements representing the flow relation between the two vertex types. Elements indicate how many tokens will be added to ($\boldsymbol{D}^+$) or removed from ($\boldsymbol{D}^-$) the places after firing a specific transition. The state evolution of a Petri net is given by:

$$M_{i+1} = M_0 + (\boldsymbol{D}^+ - \boldsymbol{D}^-)x_i \tag{2}$$

where x_i is the firing count vector which indicates how many times each transition fires.
A natural solution of the equation $\boldsymbol{D}^T x = \boldsymbol{0}$ is called a place invariant (or P-invariant) of Petri net. The state space of Petri nets is described by the reachability tree.

PETRI NET SUPERVISORS

A general Petri net supervisory control problem can be stated as follows. Given the specifications for a closed loop plant and a Petri net model of the open loop design, a supervisory feedback control law disables as few controllable transitions as possible and guarantees that the extra safety properties expressed by the specifications (such as boundedness, liveness and reversibility) are satisfied. In a manufacturing example, these three properties imply the absence of overflows and deadlocks, and ensure repeated execution of critical tasks and successful completion of production cycles.

Petri net supervisors based on P-invariants

The method of the University of Notre Dame [16], [26] synthesizes the supervisor that consists only of places and arcs. The main synthesis technique is based on the idea that specifications representing desired plant behavior could be enforced by making them invariants of the closed-loop system. The size of the controller is proportional to the number of constraints which must be satisfied. The supervisors designed in this way are computed very efficiently by a single matrix multiplication. The method may be applied to the systems whose constraints are expressed as inequalities or logic expressions involving elements of the marking and/or the firing vectors. It does not solve directly either the problem of uncontrollable transitions or the problem of required accessibility of places. These drawbacks can be mitigated partially by transforming the systems specifications to include all uncontrollably reachable states.

Petri net supervisors of Ghent University

A detailed description of the method is included in [21], [22]. The authors constructed a controlled Petri net model of the system to be controlled, treated the forbidden state problem for the class of discrete event systems and general forbidden sets, allowing the specification of a lower and/or upper bounds on the number of tokens in some places. They characterized forbidden markings through general constraint sets obtained from unions and/or intersections of simpler constraint sets expressing that some places of the net cannot contain more than a certain number of tokens. The method uses the integer linear algebraic algorithms for describing the set of states reachable from the present state under certain control settings. If the Petri net satisfies certain structural conditions, then a simple linear algebraic algorithm exists for enumerating all the markings reachable from a given state, when the control law blocks certain transitions. The method provides a necessary condition for the maximal permissivity and allows approximating the worst-case uncontrollable behavior of the original plant Petri net without doing any reachability analysis.

Petri net supervisors based on reachability tree analysis

The method is based on the reachability tree analysis algorithm [5-6], [8] and requires input files with nodes and edges of the reachability tree and a set of forbidden (not-allowed) edges. Two algorithms are included in the method.

Algorithm 1. Analysis without guaranteeing the return to marked nodes:

- Forbidden nodes in the reachability tree are labeled as inadmissible nodes *IA_nodes*.
- Deadlocks in the reachability tree are labeled as inadmissible nodes *IA_nodes*.
- Repeat while at least a new inadmissible node or a new not allowed edge (*NA_edge*) is found
    - If the edge of the reachability three is uncontrollable and its successor is *IA_node,* the predecessor node is labeled as *IA_node*.
    - If the edge of the RT is controllable and its successor is *IA_node,* the edge is not allowed and it is labeled as *NA_edge*.
    - If all the edges going out of a node are labeled as *NA_edges,* the node is labeled as *IA_node*.

Algorithm 2. Backward analysis ensuring the return to marked states (nodes):

Repeat all steps of the algorithm 1 plus the following analysis.

- Begin in marked states. Analyze backwards from which admissible nodes the marked states can be reached. The nodes from which one cannot reach the marked states using allowed edges are converted into forbidden inadmissible nodes.

The flow diagram of the algorithm 1 is shown in Figure 2. The interesting aspect of the reachability tree analysis approach that makes it potentially useful for control synthesis for large plants is the fact that the reachability tree algorithm can be applied modularly [9]. Under certain conditions, it will be possible to generate a maximally permissive supervisory controller using this modular approach.

PETRI NET APPLICATION PETRI TOOL

Although there are many existing tools for supporting design and analysis of Petri nets [27], a few tools are available for the Petri nets supervisory control synthesis. The Java application Petri Tool of the Slovak University of Technology shown in Figures 3-4 is a software application for the design, analysis and control synthesis of DES based on the Petri net models. The application is equipped with the following modules: intuitive graphical user interface, design and analysis of Petri net models, supervisory control synthesis based on [22] and [26]. Two types of places (process and control places); and three types of transitions (controllable, uncontrollable and unobservable transitions) are supported. Control places are filled with yellow color and arcs leading to and from a control place are sketched with dotted lines. After drawing a Petri net model, the incidence matrix is automatically built from the net topology. Modules of the applications solve the following problems:

- Petri net design
    - Inserting new places, transitions and arcs
    - Associating capacities and tokens with places
    - Associating controllability and observability attributes with transitions
- Edit functions
    - Labeling, selecting, copying, and pasting elements of a Petri net
    - Selecting and pasting subsets of a Petri nets
- Supervisory control problem solving
    - Adding new constraints
    - Deleting existing constraints
    - Two supervisory control methods [22] and [26].

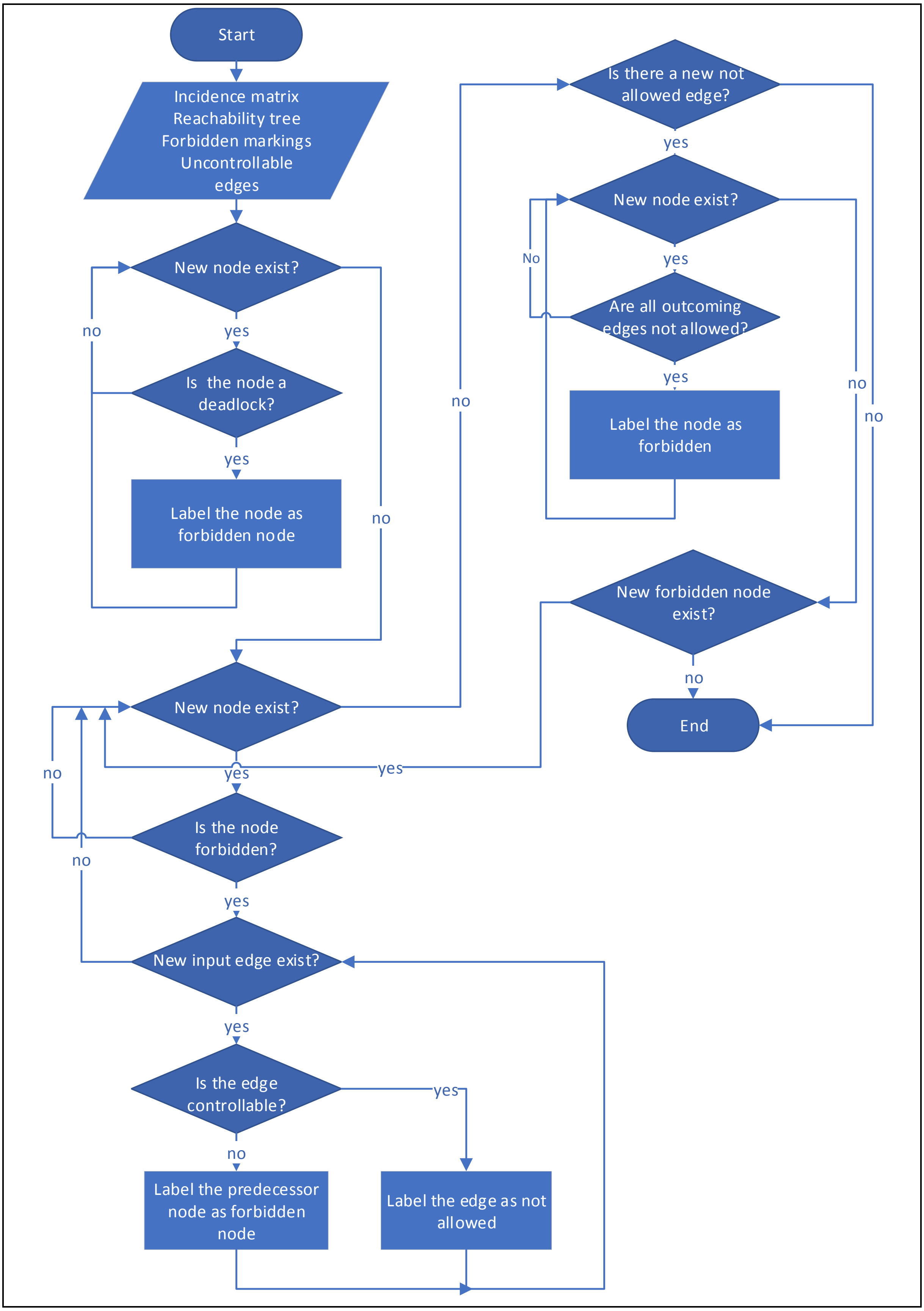

Figure 2 *Flow diagram of the supervisory control algorithm 1*

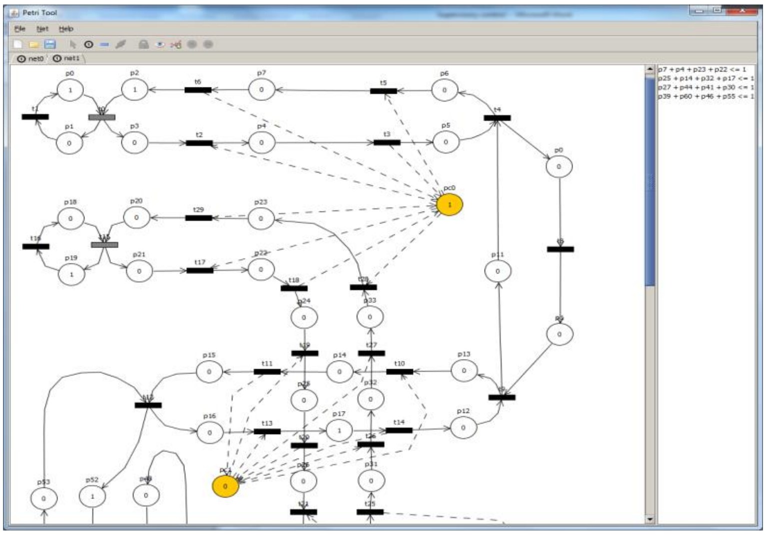

Figure 3 *Petri tool – the main window*

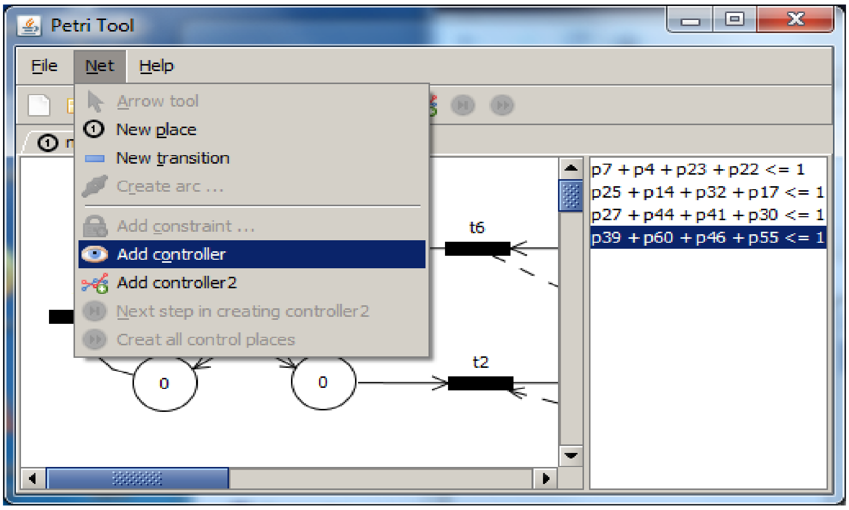

Figure 4 *Net design and control synthesis menu*

ATTAINED RESULTS

The paper provides a comparison of supervisory control methods for DES modelled by Petri nets

- Methods of supervisory control based on reachability tree were described with details
- The function of the Java application Petri tool were introduced.

DISCUSSION ON RESULTS

Methods [26] and [22] provide simple algebraic solutions and can be solved by matrix multiplications. Method [26] requires constraints expressed as logic expressions involving elements of the marking and the firing vectors. Method [22] can be used for more general constraint types and for systems with uncontrollable and unobservable transitions. Both methods are computationally efficient and they both were included in the Java application Petri tool.

Many problems of marked graphs are solvable in polynomial time. Almost no problems, including reachability tree construction of Petri net classes substantially larger than marked graphs, are solvable in polynomial time. However, real industrial systems are bounded and do not have infinite capacities, which is due to fact that in industrial plants there are always limited resources, limited capacities and limited production parameters. Reachability tree of a bounded Petri net can be designed modularly [9]. Moreover, the reachability tree in the method [5], [8] is enumerated offline, and the online supervisory control requires only lists of the allowed and forbidden edges in each particular state.

CONCLUSION

It is often necessary to supervise the behavior of DES in order to meet safety or performance criteria. DES supervisors are used to ensure that the behavior of the plant does not violate a set of constraints under a variety of operating conditions. The regulatory actions are based on observations of the plant resulting in feedback control. Two supervisory control methods based on algebraic representation of Petri net models were briefly introduced in the paper, two algorithms based on reachability tree analysis were described with details and the key features of the software application Petri tool for the supervisory control of DES modeled by Petri nets were highlighted. The application of Petri tool may be included in a master university curriculum dealing with DES control. The novelty of the paper consists in a detailed description of the third algorithms in the section Petri net supervisors and in introducing the application of Petri tool.

Acknowledgement

This publication has been written thanks to support of the Operational Program Research and Innovation for the project Research of advanced methods of intelligent information processing, ITMS code 313011T570 co-financed by the European Regional Development Fund.

References

[1] BOEL, R. K. 2002. Adaptive supervisory control. In: *Synthesis and Control of Discrete Event Systems* (Caillaud et al., Ed.), Kluwer Academic Publishers, Boston. pp. 115-124. ISBN 0-7923-7639-0.
[2] CASSANDRAS, C.G., LAFORTUNE, S. 2008. *Introduction to Discrete Event Systems*. 2nd ed. Springer Science+Business Media, LLC, 769 p., ISBN-13: 978-0-387-33332-8, e-ISBN-13: 978-0-387-68612-7.
[3] ČEŠKA, M. 1994. *Petriho sítě (Petri nets)*. Brno: CERM Academic Publishing House, 94 p. ISBN 80-85867-35-4.
[4] FLOCHOVÁ, J., HRÚZ, B. 1996. Supervisory control for discrete event dynamic systems based on Petri nets. In: *Proceeding of International conference on Process control.* Czech Republic, Horní Bečva, 2, pp. 80-83.
[5] FLOCHOVÁ, J., LIPTÁK, R., BACHRATÝ, P. 2003. An on line course for supervisory control teaching. In: *Proceeding of 6th IFAC Symposium on Advances in Control education ACE 2003*. 6th Symposium on Advances in Control Education: Finland, Oulu, pp. 198-203. Elsevier Science, ltd., 2004. ISBN-10: 9780080435596; ISBN-13: 978-0080435596.
[6] A. GIUA, F. DICESARE. 1994. Petri Net Structural Analysis for Supervisory Control. *IEEE Trans. on Robotics and Automation*, **10**(2), 185-195. ISSN 1042-296X.
[7] HO, Y.(ed.) 1982. *Discrete Event Dynamic Systems: Analyzing complexity and performance in the Modern World.* A Selected Preprint Volume. The Institute of Electrical and Electronics Engineers, Inc., New York.
[8] HRÚZ, B., FLOCHOVÁ, J. 1999. The supervisory control design based on the Petri net reachability graph. *Journal of Electrical Engineering*, **50**(11-12), 380-385. ISSN 1335-3632.

[9] HUDÁK, Š. 1994. DE-compositional reachability Analysis. *Electrotechnics Journal*, **45(**11), 424-431.
[10] KOMENDA Jan, MASOPUST Tomáš, VAN SCHUPPEN Jan H. 2012. Supervisory Control Synthesis of Discrete-Event Systems using a Coordination Scheme. *Automatica,* **48**(2), 247-254. ISSN 0005-1098.
[11] KOMENDA Jan, MASOPUST Tomáš, VAN SCHUPPEN Jan H. 2015. Coordination Control of Discrete-Event Systems Revisited. *Discrete Event Dynamic Systems,* **25** (1), 65-94. ISBN 0924-6703 (Print) 1573-7594 (Online).
[12] KUSHI, N., TAKAI, S. 2018. Synthesis of similarity enforcing supervisors for nondeterministic discrete event systems. *IEEE Transaction on Automatic Control*, **63**(5), 1457-1464. ISSN 0018-9286.
[13] LI, Y., WONHAM, W.M. 1993. Control of Discrete-Event Systems I-The based Model. *IEEE Transaction on Automatic Control*, **38**(8), 1214-1227. ISSN 0018-9286.
[14] LI, Y., WONHAM, W.M. 1994. Control of Discrete-Event Systems II-Controller Synthesis. *IEEE Transaction on Automatic Control*, **39**(3)**,** 512-531. ISSN 0018-9286.
[15] LIN, F., WONHAM, W.M. 1988. Decentralized Supervisory Control of Discrete-Event Systems. *Information Science*, **44**, 199-224. ISSN: 0020-0255.
[16] MOODY J.O., ANTSAKLIS, P.J. 2000. Petri Net Supervisors for DES with Uncontrollable and Unobservable Transitions. *IEEE Transaction on Automatic Control,* **45**(3), 462-476. ISSN 0018-9286.
[17] MURATA, T. 1998. Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, **77**(4), 541-580. ISSN 0018-9219.
[18] RAMADGE, P., WONHAM, W.M. 1987. Supervisory control of a class of discrete event processes, *SIAM J. Control and optimatization.* **25**(1), 206-230. ISSN 0363-0129 (print).
[19] RAMADGE, P.J., WONHAM, W.M. 1998. The Control of Discrete Event Systems. *Proceedings of the IEEE*, **77**(1), 81-98. ISSN 0018-9219.
[20] SREENIVAS, R.S. 1997. On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by Petri nets. *IEEE Transaction on Automatic Control*, **42**(7), 928-945. ISSN 0018-9286.
[21] STREMERSCH, G., BOEL, R.K. 2001. Decomposition of supervisory control problem for Petri nets. *IEEE Transaction on Automatic Control*, **46**(9), 1490-1496. ISSN 0018-9286.
[22] STREMERSCH, G. 2001. *Supervision of Petri nets*. Kluwer Academic Publishers, 2001. 200 p. ISBN 0-7923-7486-X.
[23] TAKAI, S. 2018. Synthesis of bisimilarity enforcing supervisors for nondeterministic discrete event systems. In: *Proceeding of 14th Int. Workshop Discrete Event Systems*. Workshop Discrete Event Systems: Italy, Sorrento Coast, pp. 1-6. ISBN 9781510867772.
[24] TAKAI, S. 2018. Maximally permissive supervisory control of nondeterministic discrete event systems with nondeterministic specification. In: *Proceeding of 57th IEEE Conference on Decision and Control*. 57th Conference on Decision and Control: USA, Florida, Miami Beach, pp. 3975-3980. ISBN 978-1-5386-1395-5.
[25] WONG. K.C., VAN SCHUPPEN, J. 1996. Decentralized Supervisory Control of DES with Communication, In: *Preprints of International Workshop on Discrete Event Systems WODES'1996*. Workshop on Discrete Event Systems: UK, Scotland, Edinburg, pp.284-289. ISBN 0 85296 664 4.
[26] YAMALIDOU, K., MOODY, J. LEMMON, M., ANTSAKLIS, P. 1996. Feedback Control of Petri Nets Based on Place Invariants. *Automatica*. **32**(1), 15-28. ISSN 0005-1098.
[27] http://www.informatik.uni-hamburg.de/TGI/PetriNets/index.php, Petri Nets World. [Online].[Accessed: 07-2019]
[28] ZHOU, C., KUMAR, R. AND JIANG, S. 2006. Control of nondeterministic discrete event system for bisimulation equivalence. *IEEE Transaction on Automatic Control,* **51**(5), pp 754-765. ISSN 0018-9286.
[29] ZHOU, C., KUMAR, R. 2011. Bisimilarity enforcement for discrete event system using deterministic control. *IEEE Transaction on Automatic Control,* **56**(12), 2986-2991. ISSN 0018-9286.

ORCID

Jana Flochová 0000-0001-5107-6337