

SOLVING DEADLOCK STATES IN MODEL OF RAILWAY STATION OPERATION USING COLOURED PETRI NETS

Michal Žarnay

University of Žilina, Faculty of Management Science and Informatics,

Address: Univerzitná 8215/1, Žilina, Slovak Republic, SK-01026

Phone: (+421-41) 513-4224, Fax: (+421-41) 565-1015, E-mail: michal.zarnay@fri.uniza.sk

Abstract: An ambition to avoid deadlock states occurring in simulation models of railway station operation requires an application of methods from resource allocation systems. Because of complexity of the underlying system, Banker's algorithm has been chosen. To test its applicability, a coloured Petri net model of a simple railway station operation was constructed and the algorithm implemented. This paper describes briefly the whole process and shows that the Banker's algorithm proved to be effective for deadlock avoidance in resource allocation system with parallel processing, flexible routing, repeated use of resources and use of professions for resource allocation.

Keywords: coloured Petri net, resource allocation system, transportation system, deadlock-avoidance policy, Banker's algorithm.

1. INTRODUCTION

In order to improve operation of transportation systems, models are made to study systems' properties and to test potential changes to them. For large and complex systems, it is done usually using simulation approach. A typical example of such an approach is simulation tool Villon[®], software tool suitable for simulation of large and complex transportation nodes like marshalling yards or logistic terminals. It offers tools for building a detailed model of real systems' operation, simulating it and analysing results. More on the simulation approach can be found in (Kavička et al., 2007).

When designing such large and complex simulation models, we face often problems with deadlock states. Deadlock state is a state of system, where two or more system processes are blocked in their execution because they wait for two or more resources, and the awaited resources are at the same time occupied by the processes included in the waiting list. The waiting processes thus block and are blocked. Unblocking this state is possible only by an exceptional operation. Difficulty of finding a solution to the problem in simulation models of complex systems is higher due to high number of involved resources and processes, and complex process behaviour. Intention to solve the problem led us to construct a simulation model of a small-scale system in the railway domain and

to implement a deadlock-solving method to it.

After analysis of properties of the given railway operation system, deadlock avoidance policy called *Banker's algorithm* has been chosen for the case. In order to implement and test the algorithm, a model of transportation system was needed, meeting both of the following criteria: having all necessary properties of the complex transportation system and producing a state space computable and analyzable within one day approximately for one configuration with current computer resources. It has been constructed in the form of *resource allocation system*, theory of which is widely used to study and solve deadlock states, and with help of *Petri net* formalism in order to use its ability for analysing models' state spaces, namely deadlock states and thus to evaluate efficiency of the deadlock avoidance algorithm. Designing and analysis of the model with algorithm has been realised with help of *CPN Tools*, worldwide used tool for hierarchical coloured Petri nets.

This paper focuses on brief description of the process. Further explanations can be found in (Žarnay, 2007). After a brief review of used terms in section 2, the construction of the model is described in section 3, followed by commenting on results in section 4 and concluding remarks in section 5.

2. PRELIMINARIES

Resource allocation system (RAS) is a system consisting of concurrently running processes that in certain stages, in order to get successfully completed, require exclusive use of certain number of system resources (Peterson, 1981). Resources are limited and re-usable as their allocation and de-allocation changes neither their character nor quantity. Based on its character, a process in the RAS can be sequential or non-sequential, i.e. some parts of the process can run concurrently. The resulting system is then either sequential (if all involved processes are sequential) or non-sequential (if at least one process in the system is non-sequential). Further a process can contain flexible routing, which means that in a certain moment, processing continues in one of available options and if correctly defined, taking any of them brings the process instance to the same final state. Finally, number and character of resources allocated at the same time distinguishes between single-unit RAS (every process is allowed to have only one resource unit allocated at a time), single-type RAS (at least one process can have more units of the same resource type at a time) or multiple-type RAS (at least one process has units of more than one resource type at a time). The outlined attributes create categories of RAS with varied complexity.

In literature, there is a wide range of methods for solving deadlock states: from deadlock prevention through deadlock avoidance to deadlock detection and recovery approaches. Some of them are available only for certain categories of RAS with restrictions on complexity of the system. More on the topic can be found e.g. in (Peterson, 1981) and (Tricas, 2003).

Petri net is a formalism used for modelling and analysis of systems with concurrent processes. It has graphical notation, precise mathematical language and analysis methods for specifying the system behaviour. Basic construction elements of Petri net are places, transitions, directed arcs and tokens. Places and transitions are two types of nodes in the net. Arcs link places with transitions and vice versa, while no pair of nodes of the same type can be connected. Tokens are elements that move in the created network between places through arcs and transitions. Principal difference between places (drawn as circles) and transitions (rectangles) lies in their relation to tokens (black dots). Places can be marked with tokens. The number of tokens and their distribution in places represent the state of the net. Transitions take

tokens from input places (i.e. places from which there are directed arcs to the transition) and provide tokens to output places (i.e. places to which there are directed arcs from the transition). This process is called firing - it performs an action in the net, changing its state. In this way, it is also possible to change the overall number of tokens in the net. More basics of Petri nets can be found in (Murata, 1989).

The basic formalism of Petri nets, called *Place/Transition* (P/T) Petri net, is often enriched or restricted to obtain enhancements or subclasses of Petri net. In this paper, we will deal with hierarchical *coloured Petri nets* (CPN), as they are introduced in (Jensen, 1997). Hierarchy allows dividing of a complex Petri net into modules called sub-pages that are interconnected through special kind of nodes (substitution transitions, fusion places). Colour is a symbolic term for value added into token in the Petri net, what distinguishes individual tokens (they are not black dots anymore). This requires additional specification for places, transitions and arcs, and thus allows construction of models with simpler net structure and added description, while keeping the same modelling power as would be with basic P/T Petri net.

CPN Tools [1] is a widely used tool for editing, simulation and analysis of hierarchical coloured Petri nets. Inscriptions are made in CPN ML, adjacent language to net structure in the CPN Tools. More on the tool can be found in (CPN Tools, online).

Banker's algorithm (BA) is a well-known deadlock-avoidance method. It answers requests for allocation of resources to processes in the system. Its working principle is based on creating an order of all running processes in the system, in which they can be all completed while using currently occupied and available resources. Algorithm thus uses information about current system state in allocation of resources and about requests for resources from individual processes in their execution stages. The intention of BA is to keep the system from unsafe states, i.e. states that lead to deadlock. Basic version of the algorithm does not allow also such states that are safe and thus restricts the state space of the system and its effectiveness inappropriately. That is why more elaborated versions of the algorithm have been developed. In this work three version have been used: basic version (labelled A), version working with partially ordered states, which may get a result for some states in a shorter time (B) and ver-

sion using more complicated calculation in certain states to recognize more safe states as ordered and thus allow larger state space (C). More on the topic can be read in (Lawley *et al.*, 1998).

3. CONSTRUCTION OF MODEL

3.1 Modelled system

Modelled system is a small railway station with 3 tracks for trains and 5 tracks for locomotives

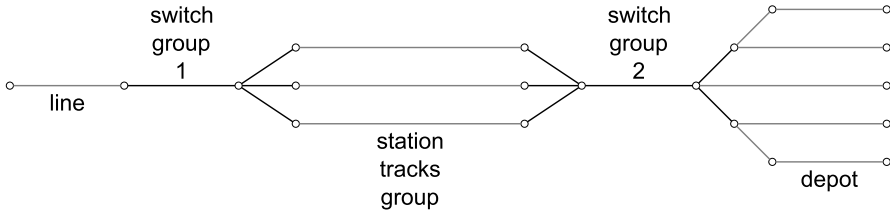


Fig. 1. Layout of the station in demonstration model with outlined types of resources.

The resources are divided into

- Tracks - 5 types of track or track group resources (*Line, Group1, Station, Group2, Depot*, see Figure 1) and
- Mobile resources - 1 type is for train locomotives (*Loco*) and 3 types are for personnel (*Coupler, Examiner* and *CoupEx* - crew member able to perform work of both coupler and examiner).

3.2 CPN model

Model is a hierarchical CPN of RAS with one non-sequential process type. It was created in CPN Tools. The process type subnet is divided into three pages. One page contains common part for both options of flexible routing of the process (Figure 3). Two other pages contain subnets for alternative routes of the process A and B (the latter is displayed at Figure 4).

On the figures, stages of the process description are represented by all places but 3: *Environment, Tracks* and *Mobile Resources*. Re-

sources could be placed into one place, but to make the model look clearer, they are divided to two: *Tracks* used for track (fixed) resources (*cTracksOcc* colour set) and *Mobile Resources* for mobile resources (locomotives and station personnel, *cMobileResourcesOcc* colour set). The place *Environment* represents place for trains outside of system (i.e. currently not executed processes). Altogether, the model contains 43 places, 5 fusion groups (using 18 fusion places), 31 ordinary and 3 substitution transitions.

3.3 Deadlock avoidance method

From analysis of properties of given railway operation system came out, that it falls into category of complex RAS with processes having the following properties: parallel processing, flexible routing, use of resources of multiple types at once and use of professions for handling of resources (Žarnay, 2007). For this combination of properties, most of methods available in literature (e.g. Li & Wonham, 1993 and 1994, Fanti *et al.*, 1997, Reveliotis *et*

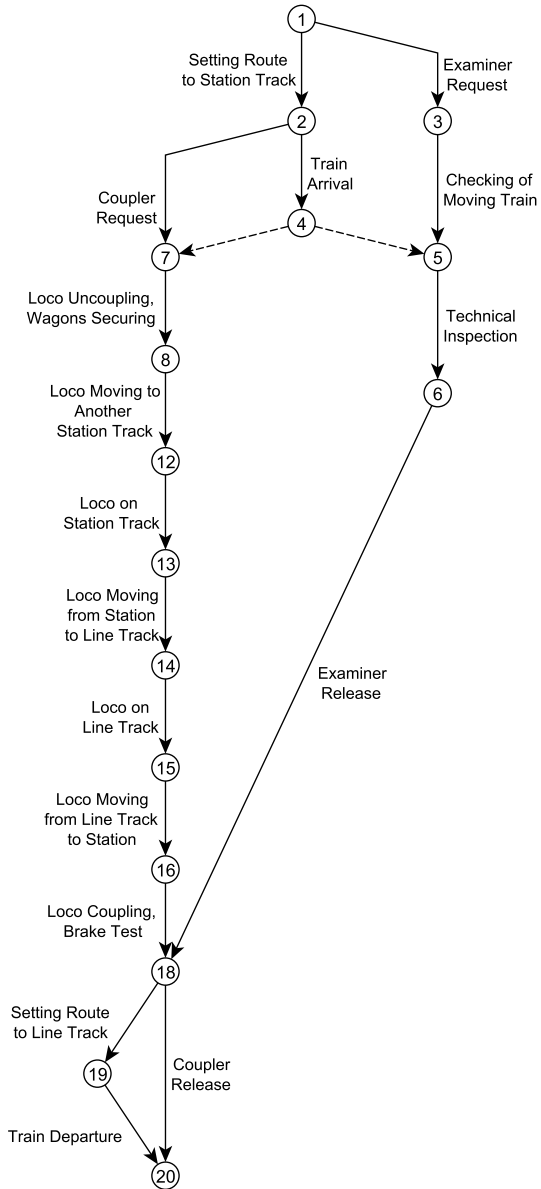


Fig. 2. Description of process with option B in the form of flowchart.

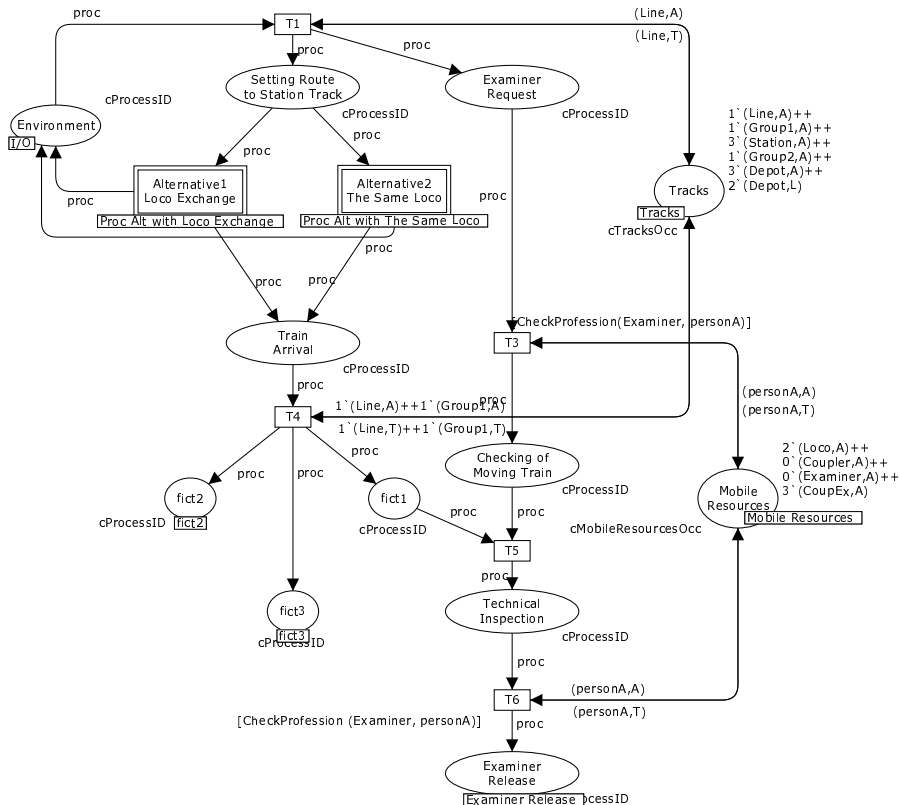


Fig. 3. CPN of common part of process before splitting to two options represented by substitution transitions.

al., 1997, Park *et al.*, 2001, Park & Reveliotis, 2001, Ezpeleta & Recalde, 2004, Reveliotis & Choi, 2006) are not suitable. That is why a deadlock avoidance policy not having restrictions in complexity of RAS, called *Banker's algorithm* has been chosen and applied for this case.

In order to apply it in the model, the BA had to be slightly adjusted to properties discussed in section 3. While the properties of parallel processing, flexible routing and relations between process stages and resources are implemented in the Petri net formalism, the use of professions for the allocation of resources influenced the BA's implementation itself.

The latter was solved by separating data structures for resources and for resources requests.

Resources are represented in such a form that all resource instances with the same attributes belong to one resource type, i.e. one colour in the CPN. On the other hand, resource requests respond to professions (attributes of resources), i.e. other colours in CPN. In addition, relevant mapping functions were integrated in the BA.

The BA (3 versions) and supporting routines are implemented as a set of 35 functions in CPN ML. It uses data stored in tokens in two dedicated places reachable throughout the model as fusion places. Data consist of information about current system state that is updated every time, when the system state changes. The BA is called every time, when a resource request is made. It answers the question, whether the resource can be allocated

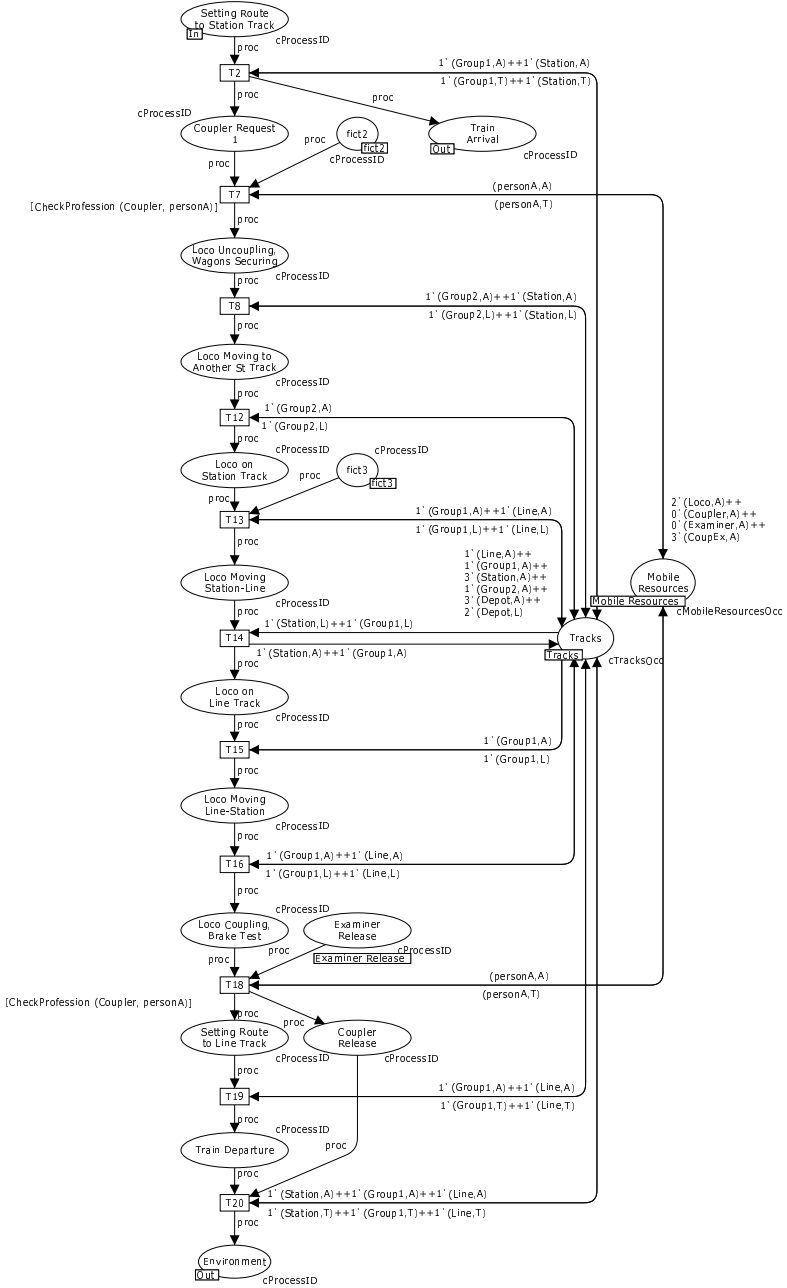


Fig. 4. CPN of the separated option B from process description.

to requesting process in the current state of the system or not. If not, the process awaits a new state, when the request will be considered again. The behaviour is achieved by connecting the algorithm to guards of relevant transitions (16 out of 31 transitions), where resources are requested. Result of algorithm placed in transition's guard decides, whether the transition in question will be allowed to fire or not.

4. RESULTS

24 model configurations were prepared and tested. They differed in:

- Personnel configuration – 2 crews of personnel differing in attributes for 3 members (i.e. 2

versions of this attribute)

- Number of trains (i.e. processes) in the model: 2, 3 and 4 - minimum 2 trains to have at least one opportunity for deadlock situation in the course of simulation and no more than 4 trains because of limited size of the modelled station (i.e. 3 versions of attribute)
- Deadlock avoidance algorithm – without BA and 3 versions of the BA: A, B and C (i.e. 4 versions of this attribute).

For every prepared configuration, state space has been analysed. From the analysis, 3 criteria are relevant for this work: number of deadlock states, state space size and calculation time. Results are summarized in tables 1-3.

Deadlock States	–	A	B	C
2 trains + crew I	1	0	0	0
3 trains + crew I	306	0	0	0
4 trains + crew I		0	0	
2 trains + crew II	6	0	0	0
3 trains + crew II	760	0	0	0
4 trains + crew II		0	0	0

Tab. 1. State space analysis results – number of deadlock states.

State Space Size	–	A	B	C
2 trains + crew I	18002	7986	7986	11142
3 trains + crew I	179408	23362	23362	83920
4 trains + crew I		46327	46327	
2 trains + crew II	20798	9978	9978	12172
3 trains + crew II	228500	29338	29338	35920
4 trains + crew II		58279	58279	71443

Tab. 2. State space analysis results – number of all states in state space.

Calculation Time [s]	–	A	B	C
2 trains + crew I	163	61	59	126
3 trains + crew I	13252	497	491	6323
4 trains + crew I		1937	1843	
2 trains + crew II	230	98	94	144
3 trains + crew II	23963	797	773	1185
4 trains + crew II		3052	3270	4962

Tab. 3. State space analysis results – time duration of state space calculation (for orientation).

Table 1 contains the most important result of the whole work. It shows that when no deadlock avoidance algorithm is used (second column with highlighted cells), the deadlock states may occur in the system run. When the BA has been used, all deadlock states were eliminated (columns A, B, C).

Three empty cells represent configurations, when the state space calculation has not been finished by the tool successfully. In these cases, simulation of the CPN model has been carried out. It finished as expected. In the configurations without BA, simulation finished always in a deadlock state, usually after less than ten thousand steps. In the case with the C version of BA, simulation has reached the set limit of 1 million of steps in all attempts.

Table 2 provides results for comparing size of calculated state space in the configurations. It shows that versions A and B of the BA reduce the state space to the same level. Version C allows much more states, thanks to more complex calculation.

Table 3 gives an estimate of time for calculation of state space. It reflects time of BA individual calculations, assuming that state space related calculations have invariant time length. In this perspective, BA version A is for most configurations slightly slower than version B, while they both produce the same results in number of accepted states. The expected shorter calculation of version B for resource requests in some system states is thus confirmed. Time length of calculations of version C to the time length of version A is relatively higher than is the corresponding ratio in number of accepted states. This confirms to certain extent higher complexity of the version C of BA. However, time values should be taken as informative only, since the state space calculations may have been influenced (although with minimal probability) by other processes potentially running in the test system outside of the CPN Tools.

Results have been calculated on 3 GHz Intel® Pentium® D CPU with 1 GB of memory.

5. CONCLUSION

In this paper, a coloured Petri net model for resource allocation model of railway station operation has been introduced and its use for testing Banker's algorithm for deadlock avoidance briefly described. It is an important step on the way towards solving of deadlock states in simulation

models of complex transportation systems' operation like those constructed with help of the tool Villon®.

The formalism of hierarchical coloured Petri nets showed to be very suitable for solving of this problem. Petri nets natively provide the (for this task) necessary state space analysis for systems of reasonable size. At the same time, construction of Petri net model with help of small number of building elements is rather easy and smooth (especially for experienced users), what is even reinforced by use of colours and hierarchy in the chosen subclass of high-level Petri nets. The environment of the CPN Tools, in addition, allowed implementing the Banker's algorithm and integrating it to the CPN model – thanks to the CPN ML. Flexibility of the approach has been utilized also by definition of test configurations, where changes in model parameters have been done easily and fast.

The Banker's algorithm proved to be an effective deadlock-avoidance method for such a complex system. Testing of 3 versions proved the assumptions: the most elaborated version C restricts the system state space less than basic version A and basic version B with a small refinement for certain system situations. However, version C is more complex and takes longer to compute in some system situations. Difference between versions A and B in shorter calculation for some system situations in version B has been also proved in most compared configurations. Should the effectiveness of the simulation model control be a priority, then the version C of the algorithm is the most suitable. If the time length of simulation runs is a more important concern than effectiveness of the model, then the version B is advised.

Next steps in the research lead in two directions. One of them is a study of application of other deadlock-avoidance methods for transportation systems modelled in Petri nets. The other and more important direction is application of the Banker's algorithm in the simulation tool Villon®. Here, the challenge of large number of processes and resources by the application of BA must be solved first. When that is succeeded, construction of simulation models in Villon® can become smoother and the space for simulation experiments will be significantly enlarged.

ACKNOWLEDGEMENT

This paper has been supported by the grant of the Scientific Grant Agency VEGA 1/4057/07 in the

Slovak Republic and the research project MŠM 0021627505 - Theory of transport systems in the Czech Republic.

REFERENCES

- CPN Tools home page. [online].
[Cited 30 April 2008] Available at:
<<http://www.daimi.au.dk/CPNTools/>>.
- Ezpeleta, J. and Recalde, L. (2004). A deadlock avoidance approach for non-sequential resource allocation systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 1, 93-101.
- Jensen, K. (1997). *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts*. Monographs in Theoretical Computer Science, Springer-Verlag.
- Fanti, M. P., Maione, B., Mascolo, S. and Turchiano, B. (1997). Event-based feedback control for deadlock avoidance in flexible production systems. *IEEE Transactions on Robotics and Automation*, 3, 347-363.
- Kavička, A., Klíma, V. and Adamko, N. (2007). Simulations of transportation logistic systems utilizing agent-based architecture. *International Journal of Simulation Modelling*, 13-24.
- Lawley, M. A., Reveliotis, S. A. and Ferreira, P. M. (1998). The application and evaluation of Banker's algorithm for deadlock-free buffer space allocation in flexible manufacturing systems. *The International Journal of Flexible Manufacturing Systems*, 73-100.
- Li, Y. and Wonham, W. M. (1993). Control of Vector Discrete-Event Systems – Part I: The Base Model. *IEEE Transactions on Automatic Control*, 8, 1214-1227.
- Li, Y. and Wonham, W. M. (1994). Control of Vector Discrete-Event Systems – Part II: Controller Synthesis. *IEEE Transactions on Automatic Control*, 3, 512-531.
- Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 4, 541-580.
- Park, J. and Reveliotis, S. A. (2001). Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Transactions on Automatic Control*, 10, 1572-1583.
- Peterson, J. L. (1981). *Operating System Concepts*. Addison-Wesley.
- Reveliotis, S. A., Lawley, M. A. and Ferreira, P. M. (1997). Polynomial-complexity deadlock avoidance policies for sequential resource allocation systems. *IEEE Transactions on Automatic Control*, 1344-1357.
- Park, J., Reveliotis, S. A., Lawley, M. A. and Ferreira, P. M. (2001). A correction to the RUN DAP for conjunctive RAS presented in 'Polynomial complexity deadlock avoidance policies for sequential resource allocation systems'. *IEEE Transactions on Automatic Control*, 672.
- Reveliotis, S. A. and Choi, J. Y. (2006). Designing reversibility-enforcing supervisors of polynomial complexity for bounded Petri nets through the theory of regions. *27th International Conference on the Application and Theory of Petri Nets and Other Models of Concurrency*, LNCS 4024 (S. Donatelli and P. S. Thiagarajan, Eds.), Springer, 322-341.
- Tricas, F. (2003). *Deadlock Analysis, Prevention and Avoidance in Sequential Resource Allocation Systems*. Ph.D. Thesis. Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza.
- Žarnay, M. (2007). *Systém na podporu rozhodovania pre riadenie dopravných procesov* [Decision-Support System for Control of Transportation Processes]. PhD Thesis. Faculty of Management Science and Informatics, University of Žilina.