

# Cyclic Steady State Refinement: Multimodal Processes Perspective

Grzegorz Bocewicz, Peter Nielsen, Zbigniew Banaszak, Vinh Dang

► **To cite this version:**

Grzegorz Bocewicz, Peter Nielsen, Zbigniew Banaszak, Vinh Dang. Cyclic Steady State Refinement: Multimodal Processes Perspective. International Conference on Advances in Production Management Systems (APMS), Sep 2011, Stavanger, Norway. pp.18-26, 10.1007/978-3-642-33980-6\_3. hal-01524185

**HAL Id: hal-01524185**

**<https://hal.inria.fr/hal-01524185>**

Submitted on 17 May 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Cyclic Steady State Refinement: Multimodal Processes Perspective

Grzegorz Bocewicz<sup>1</sup>, Peter Nielsen<sup>2</sup>, Zbigniew A. Banaszak<sup>3</sup>, Vinh Quang Dang<sup>2</sup>

<sup>1</sup> Koszalin University of Technology, Dept. of Computer Science and Management, Koszalin, Poland, e-mail: bocewicz@ie.tu.koszalin.pl

<sup>2</sup> Dept. of Mechanical and Manufacturing Engineering, Aalborg University, Denmark, e-mail: peter@m-tech.aau.dk; vinhise@m-tech.aau.dk

<sup>3</sup> Warsaw University of Technology, Faculty of Management, Dept. of Business Informatics, Warsaw, Poland, e-mail: Z.Banaszak@wz.pw.edu.pl

**Abstract.** The cyclic scheduling problem modeled in terms of Cyclic Concurrent Process Systems is considered. The problem can be seen as a kind of Diophantine problem, hence its solvability, i.e. schedulability, plays a pivotal role in many supply-chain problems. In contradiction to the traditionally offered solutions the approach proposed allows one to take into account such behavioral features as transient periods and deadlocks occurrence. So, the contribution's aim is the modeling framework enabling an evaluation of cyclic scheduling problems solvability, i.e., the declarative approach to reachability problems regarding cyclic steady states determination as well as conditions guaranteeing assumed performance of multimodal processes executed within a concurrent cyclic processes environment.

**Keywords:** cyclic processes, multimodal process, state space, periodicity, dispatching rules.

## 1 Introduction

Operations in cyclic processes are executed along sequences that repeat an indefinite number of times. In everyday practice they arise in different application domains such as manufacturing as well as service domains (covering such areas as workforce scheduling, timetabling, and reservations [4], [6], [8]). Such systems belong to a class of systems of concurrently flowing cyclic processes (SCCP) [1], [2], [7], and relevant cyclic scheduling problems belong to a class of NP-hard ones [5]. Subway or train traffic can be considered as an example of such kind of systems.

Subway trains following particular metro lines can be treated as cyclic processes passing, the sequence of stations, allows one to state a question concerning a minimization of the total passenger travel time. So, if passengers travel between two distinguished locations in the transportation network for which no direct connection exists, i.e., transfers become inevitable, the relevant scheduling problem can be stated in the following way. Given is a set of metro lines, i.e. the set of trains' routings. Some lines may share common stations. Given is also a headway time, i.e., the fixed interval

between the trips of a line sometimes called the period time. The question considered is: Which transportation route between two designated terminal stations in the transportation network provides the shortest travel time subject to assumed constraints? So, the best transportation route of the multimodal process, i.e. sharing different lines, is sought. This type of system finds many analogies in manufacturing, in the form of routing between fixed manufacturing lines, e.g. routing and scheduling of AGVs within an Automated Storage and Retrieval System environment.

Many models and methods have been proposed to solve the cyclic scheduling problem. Among them, the mathematical programming approach (usually IP and MIP), max-plus algebra [7], constraint logic programming [2], [3], [9] evolutionary algorithms and Petri nets [1], [8] frameworks belong to the more frequently used. Most of these methods are oriented towards finding a minimal cycle or maximal throughput while assuming deadlock-free process flow. Approaches trying to estimate the SCCP cycle time from local cyclic processes structure and the synchronization mechanism used (i.e. rendezvous or mutual exclusion instances) are quite unique [1].

In that context our main contribution is to propose a new modeling framework enabling to evaluate the cyclic steady state of a given SCCP on the base of the assumed processes topology, dispatching rules employed, and an initial state. So, the objective of the presented research is to provide the observations useful in the course of multimodal processes routing and scheduling in systems composed of concurrently flowing cyclic processes interacting between oneself through mutual exclusion protocol. The goal is to provide the conditions useful for routing and scheduling of SCCP so as to be effective in the course of multimodal processes control.

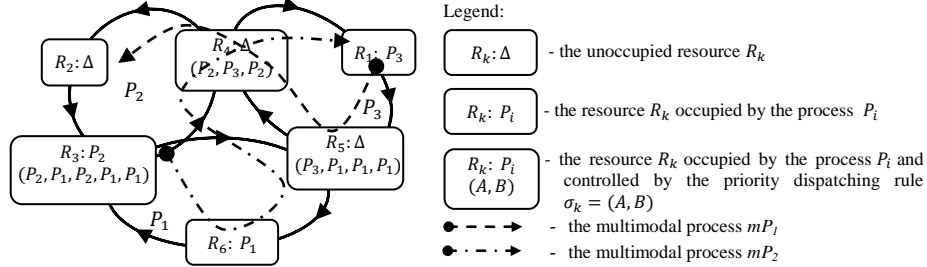
In order to achieve it, the paper presents an introduction to the SCCP, and then to the concept of state space of considered systems. Consequently, the terms of a cyclic steady state and the corresponding space of cyclic steady states are introduced as well as conditions linking them with multimodal processes scheduling are presented.

## 2 Concept of Multimodal Processes System

### 2.1 Systems of Concurrent Cyclic Processes

Consider the digraph shown in Fig. 1. The distinguished are three cycles specifying routes of cyclic processes  $P_1$ ,  $P_2$  and  $P_3$ , respectively. Each process route, specified by the sequence of resources passed on among its execution, can interact with other processes through so-called system common resources. So, the process routes are specified as follows:  $p_1 = (R_6, R_3, R_5)$ ,  $p_2 = (R_2, R_3, R_4)$ ,  $p_3 = (R_1, R_5, R_4)$ , where the resources  $R_3$ ,  $R_4$ ,  $R_5$ , are shared resources, since each one is used by at least two processes, and the resources  $R_1$ ,  $R_2$ ,  $R_6$ , are non-shared because each one is exclusively used by only one process. Processes sharing common resources interact with each other on the base of a mutual exclusion protocol. Possible resources conflicts are resolved with the help of assumed priority rules determining the order in which processes make their access to common shared resources (for instance, in case of resource  $R_4$ ,  $\sigma_4 = (P_2, P_3)$  – the priority dispatching rule determines the order in which

processes can access to the shared resource  $R_4$ , i.e. at first to the process  $P_2$ , then to the process  $P_3$ , next to  $P_2$  and once again to  $P_3$ , and so on).



**Fig. 1.** Process routes structure of SCCP owning three processes

In general case, each process  $P_i$  (where:  $P_i \in P = \{P_1, P_2, \dots, P_n\}$ ,  $n$  – a number of processes) executes periodically a sequence of operations performed on resources creating the given process route  $p_i = (R_{j_1}, R_{j_2}, \dots, R_{j_{lr(i)}})$ ,  $j_k \in \{1, 2, \dots, m\}$ , where  $lr(i)$  – a length of cyclic process route,  $m$  – the number of resources, and  $R_{j_k} \in R = \{R_1, R_2, \dots, R_m\}$ .

The time  $t_{i,j} \in \mathbb{N}$ , of operation executed on  $R_j$  along  $P_i$ , is defined in the domain of uniform time units ( $\mathbb{N}$  – set of natural numbers). So, the sequence  $T_i = (t_{i,j_1}, t_{i,j_2}, \dots, t_{i,j_{lr(i)}})$  describes the operation times required by  $P_i$ . To each common shared resource  $R_i \in R$  the priority dispatching rule  $\sigma_i = (P_{j_1}, P_{j_2}, \dots, P_{j_{lp(i)}})$ ,  $j_k \in \{1, 2, \dots, n\}$ ,  $P_{j_k} \in P$  is assigned, where  $lp(i) > 1$ ,  $lp(i)$  is a number of processes dispatched by  $\sigma_i$ . In that context a *SCCP* can be defined as follows [3]:

$$SC_l = (R, P, \Pi, T, \theta), \quad (1)$$

where:  $R = \{R_1, R_2, \dots, R_m\}$  – the set of resources,  
 $P = \{P_1, P_2, \dots, P_n\}$  – the set of local processes,  
 $\Pi = \{p_1, p_2, \dots, p_n\}$  – the set of local process routes,  
 $T = \{T_1, T_2, \dots, T_n\}$  – the set of local process routes operations times,  
 $\theta = \{\sigma_1, \sigma_2, \dots, \sigma_m\}$  – the set of dispatching priority rules.

Let us assume the all operation times are equal to a unit operation time (noted as: u.o.t.),  $\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, lr(i)\}, (crd_j T_i = 1 \text{ u.t.})$ . In the case of the *SCCP* considered the following constraints imposed on processes interaction are assumed:

- The new process operation may begin only if current operation has been completed and the resource designed to this operation is not occupied.
- The new process operation can be suspended only if designed resource is occupied.
- Processes suspended cannot be released and processes are non-preempted.

The main question concerns of a *SCCP* cyclic steady state behavior and a way this state depends on direction of local process routes as well as on priority rules, and an initial state, i.e. initial process allocation to the system resources. Assuming such a steady there exists the next question regarding of travel time along assumed multimodal process route linking distinguished resources plays a primary role.

## 2.2 States Space

Consider the  $k$ -th state  $S^k$  (2) composed of the sequence of processes allocation  $A^k$ , the sequence of semaphores (encompassing the rights guaranteeing processes' access to a resource)  $Z^k$ , and the sequence of semaphore indices  $Q^k$ :

$$S^k = (A^k, Z^k, Q^k), \quad (2)$$

where:  $A^k = (a_1^k, a_2^k, \dots, a_m^k)$  – **the processes allocation** ( $m$  – a number of CCPS resources),  $a_i^k \in P \cup \{\Delta\}$  ( $P$  – a set of processes:  $P = \{P_1, P_2, \dots, P_n\}$ ) means the process is allotted to the  $i$ -th resource  $R_i$  in the  $k$ -th state,  $a_i^k = P_g$  means, the  $i$ -th resource  $R_i$  is occupied by the process  $P_g$ , and  $a_i^k = \Delta$  – the  $i$ -th resource  $R_i$  is unoccupied.

$Z^k = (z_1^k, z_2^k, \dots, z_m^k)$  – **the sequence of semaphores** corresponding to the  $k$ -th state, where  $z_i^k \in P$  means the name of the process (specified in the  $i$ -th dispatching rule  $\sigma_i$ , allocated to the  $i$ -th resource) allowed to occupy the  $i$ -th resource  $R_i$ .  $Q^k = (q_1^k, q_2^k, \dots, q_m^k)$  – **the sequence of semaphore indices** corresponding to the  $k$ -th state, where  $q_i^k$  means the position of the semaphore  $z_i^k$  in the priority dispatching rule  $\sigma_i$ :  $z_i^k = \text{crd}_{(q_i^k)} \sigma_i$ ,  $q_i^k \in \mathbb{N}$  ( $\text{crd}_i D = d_i$ , for  $D = (d_1, \dots, d_i, \dots, d_w)$ ).

**The state  $S^k = (A^k, Z^k, Q^k)$  is feasible** only if for any of its  $a_i^k$  co-ordinate in  $A^k = (a_1^k, a_2^k, \dots, a_m^k)$ , the following conditions hold:

- i)  $\forall_{i \in \{1, 2, \dots, n\}} \exists!_{j \in \{1, 2, \dots, m\}} (P_i = \text{crd}_j A^k), \quad (3)$
- ii)  $\forall_{i \in \{1, 2, \dots, m\}} (\text{crd}_i A^k \in P \cup \{\Delta\}), \quad (4)$
- iii) if the values of the semaphore  $Z^k$  and **the sequence of semaphore indices**  $Q^k$  result from allocation (3), (4) (i.e., semaphores determining busy resources show the processes allotted to them, while indexes show semaphore values).

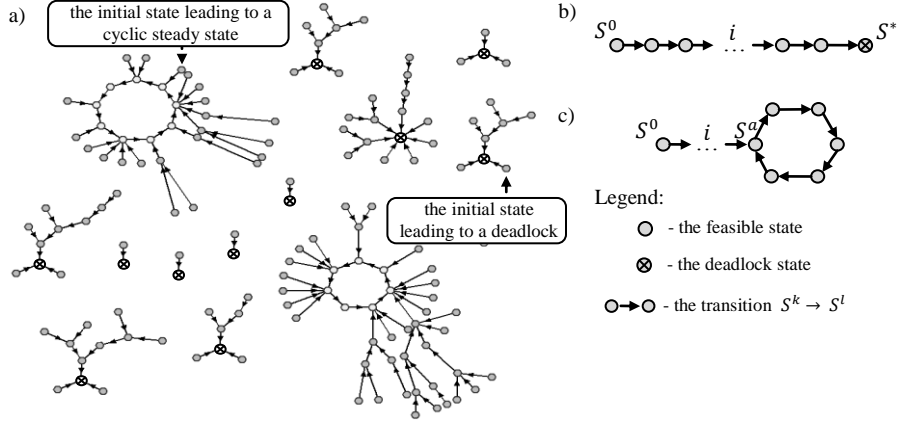
The set of all feasible states is called **a state space**  $\mathbb{S}$ . The states  $S^k \in \mathbb{S}$  and  $S^l \in \mathbb{S}$  can be linked determining transitions among states, e.g.  $S^k \rightarrow S^l$  means the state  $S^l$  follows the state  $S^k$ . In general case, the states can be linked via other states, e.g.  $S^r, S^w$  what leads to the following sequence of transitions:  $S^k \rightarrow S^r \rightarrow S^w \rightarrow S^l$ ,  $S^k \xrightarrow{i} S^l$  in short, where:  $i$  – means the number of states  $S^r, S^w$  linking  $S^k, S^l$ , e.g., in case consider  $S^k \xrightarrow{2} S^l$ , and  $S^k \xrightarrow{0} S^r$ . a transition  $S^k \rightarrow S^l$ , can be represented as the following function:

$$S^l = \delta(S^k), \quad (5)$$

where:  $\delta$  – is a transition function  $\delta: \mathbb{S} \rightarrow \mathbb{S}$  [3].

**The deadlock state**  $S^k$  (denoted in the rest of the paper by  $S^*$ ) is defined as the state such that does not exist any feasible state  $S^l \in \mathbb{S}$ , following the transition  $S^k \rightarrow S^l$ . So, the deadlock state means the all processes in the CCPS are suspended.

In a state space following from a given SCCP model (1), and the next state function (5) one can easily distinguish two kinds of reachability digraphs (see Fig. 2a). The properties specifying particular kinds of possible behaviors are as follows:



**Fig. 2.** Illustration of the state space  $\mathbb{S}$  structure a), the string-like digraph ending with a deadlock state b), and the string-like digraph ending with a state belonging to a cyclic steady state c)

### Property 1

Consider the SCCP model (2) and an initial state  $S^0 \in \mathbb{S}$ . The reachability digraph, generated from an initial state  $S^0$ , is either a string-like digraph ending with a deadlock state, i.e. (Fig. 2b):  $S^0 \xrightarrow{i} S^*$ ,  $i \geq 0$ , or a string-like digraph ending with a state belonging to the cyclic steady state (Fig. 2c):  $S^0 \xrightarrow{i} S^a \xrightarrow{Tc-1} S^a$ ,  $i \geq 0$ , where  $Tc$  – a number of states creating the cycle.

### Property 2

Consider the sets of initial states ending with the same the  $k$ -th deadlock state  $SD_k \subset \mathbb{S}$ , and initial states ending with states belonging to the  $i$ -th cyclic steady state  $SC_i \subset \mathbb{S}$ , respectively. Assume a state space generated by the SCCP model (1) and the next-state function (5). The following conditions hold:

- i)  $\forall S^i, S^j \in \mathbb{S}^*, i \neq j [SD_i \cap SD_j = \emptyset]$ , where:  $\mathbb{S}^*$  – the set of deadlock states,
- ii)  $\forall S^l, S^k \in \mathbb{S}^t, l \neq k [SC_l \cap SC_k = \emptyset]$ , where:  $\mathbb{S}^t$  – the set of cyclic steady states.

That means the following questions can be considered:

- What are the sufficient conditions guaranteeing the state space is free of deadlock states and states leading to the deadlocks?
- What are the conditions guaranteeing transitions among assumed cyclic steady states?
- What is the period of the cyclic state of multimodal processes performed in the SCCP executed in a given cyclic steady state?

## 2.3 The multimodal processes

Let  $\{P_i | i \in \{1, \dots, n\}\}$  be the set of cyclic processes determined by the set of process routes  $\{p_i | i \in \{1, \dots, n\}\}$ . Consider the set of multimodal processes  $MP = \{mP_1, mP_2, \dots, mP_u\}$ , where  $u$  – is a number of multimodal processes. Due to our in-

formal definition, each multimodal process  $mP_i$  is specified by the transportation route  $mp_i$  which is a sequence of sections of local cyclic process routes:

$$mp_i = (mpr_j(a_j, b_j), mpr_k(a_k, b_k), \dots, mpr_h(a_h, b_h)), \quad (5)$$

where:  $mpr_j(a, b) = (crd_a p_j, crd_{a+1} p_j, \dots, crd_b p_j)$ ,  $crd_i D = d_i$ , for  $D = (d_1, d_2, \dots, d_i, \dots, d_w)$ ,  $\forall a \in \{1, 2, \dots, lr(i)\}$ ,  $\forall j \in \{1, 2, \dots, n\}$ ,  $crd_a p_j \in R$ .

By analogy to local cyclic processes the sequence  $mT_i = (mt_{i,j_1}, mt_{i,j_2}, \dots, mt_{i,j_{lm(i)}})$ ,  $mt_{i,j_k} \in \mathbb{N}$ , describes operation times required by operations executed along  $mP_i$  (where:  $lm(i)$  is the length of the  $i$ -th multimodal route  $mP_i$ ). In that context a *SCCP* can be defined as a pair [3]:

$$SC = (SC_l, SC_m), \quad (6)$$

where:  $SC_l = (R, P, \Pi, T, \Theta)$  – is specified by the (1),

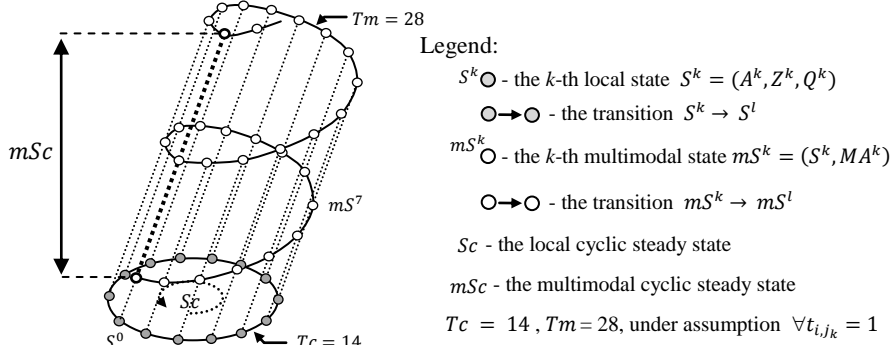
$SC_m = (MP, M\Pi, MT)$  – characterizes the *SCCP* behavior, i.e.

$MP = \{mP_1, mP_2, \dots, mP_u\}$  – the set of multimodal process,  $M\Pi =$

$\{mp_1, mp_2, \dots, mp_u\}$  – the set of multimodal process routes,  $MT =$

$\{mT_1, mT_2, \dots, mT_u\}$  – the set of multimodal process routes operations times.

Since, the multimodal processes execution depends on a steady state of the *SCCP* considered their periodicity depends on the current  $Tc$  of  $Sc$  (see Fig. 3). In turn that means that initial states and sets of dispatching rules can be seen as control variables allowing one to “adjust” multimodal processes schedule.



**Fig. 3.** Graphical illustration of the relationship between the cyclic steady state  $Sc$  of *SCCP* and the multimodal cyclic steady state  $mSc$

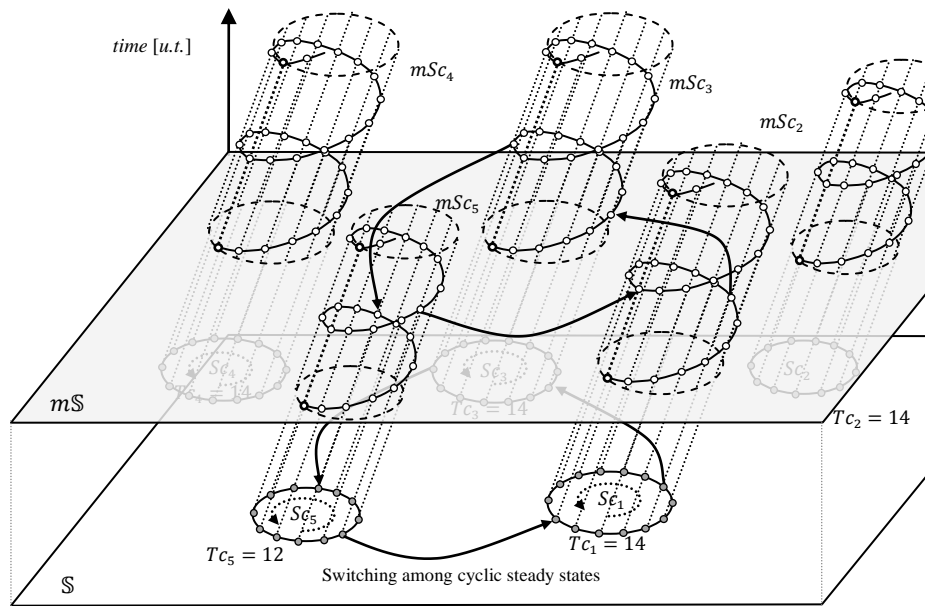
### 3 Illustrative example

The approach proposed is based on the system of concurrently flowing cyclic processes concept assuming its cyclic steady state behavior guaranteed by the given sets of dispatching rules and initial states. So, the multimodal processes scheduling, that can be seen as processes composed of parts of local cyclic processes, lead to the

two fundamental questions. Does there exist a control procedure (i.e. a set of dispatching rules and an initial state) guaranteeing an assumed steady cyclic state subject to SCCP's structure constraints? Does there exist a SCCP's structure such that an assumed steady cyclic state (e.g. following requirements caused by multimodal processes at hand) can be achieved? In other words, an approach provides a framework enabling to take into account both the forward and backward way of cyclic scheduling problem formulation. Moreover, the question regarding possible switching among both the local and multimodal steady state states can be considered. The illustration of the local and multimodal cyclic state spaces prototyping is shown in the Table 1.

**Table 1.** The local and multimodal cyclic state spaces prototyping for SCCP from Fig. 1.

Structure	Cyclic Steady State	Period of the Cyclic Steady State [u.o.t]	Period of the multimodal process [u.o.t]	Cyclic Steady State	Period of the Cyclic Steady State [u.o.t]	Period of the multimodal process [u.o.t]
SCCP (Fig. 1)	$Sc_1$	14	28	$Sc_6$	14	28
	$Sc_2$	14	28	$Sc_7$	12	24
	$Sc_3$	14	28	$Sc_8$	12	24
	$Sc_4$	14	28	$Sc_9$	12	24
	$Sc_5$	12	24	$Sc_{10}$	12	24



**Fig. 4.** Illustration of cyclic steady state spaces ( $Sc$  and  $mSc$ ) for SCCP from Fig. 1

Because of the discrete structure of SCCP topology and dispatching rules as well as the discrete event nature of processes execution the considered problems of  $mSc$  and  $Sc$  scheduling (including switching among cyclic steady states) can be modeled in terms of Constraint Satisfaction Problem (CSP) [2], [3], and consequently implemented in declarative languages environment – OzMozart system. The illustration of pos-



sible cyclic schedules taking into account possible switching among cyclic steady states is shown in Fig. 4.

## 4 Concluding remarks

In contradiction to the traditionally offered solutions the approach presented allows one to take into account such behavioral features as transient periods and deadlock occurrence. So, the novelty of the modeling framework offered lies in the declarative approach to reachability problems enabling multimodal cyclic process evaluation as well as in CSP-based evaluation of possible switching among cyclic steady states from both the local and multimodal cyclic state spaces.

The approach presented leads to solutions allowing the designer to compose elementary systems in such a way as to obtain the final SCCPSs' scheduling system with required quantitative and qualitative behavioral features while employing the sufficient conditions provided. So, we are looking for a method allowing one to replace the exhaustive search for the admissible control by a step-by-step structural design guaranteeing the required system behavior.

## REFERENCES

1. Alpan G., Jafari M.A., Dynamic analysis of timed Petri nets: a case of two processes and a shared resource. *IEEE Trans. on Robotics and Automation* 1997; Vol.13, No.3: 338-346
2. Bocewicz G., Banaszak Z., Wójcik R., Design of admissible schedules for AGV systems with constraints: a logic-algebraic approach. In: *Agent and Multi-Agent Systems: Technologies and Applications*, Nguyen N.T., Grzech A., et al. (Eds.), *Lecture Notes in Artificial Intelligence LNAI*, Vol. 4496, Springer-Verlag, Berlin-Heidelberg, 2008, 578-587
3. Bocewicz G., Wójcik R., Banaszak Z., Cyclic Steady State Refinement. In: *International Symposium on Distributed Computing and Artificial Intelligence*, (Eds.) Abraham, A., Corchado, J.M., Rodríguez González, S., de Paz Santana, J.F. *Series: Advances in Intelligent and Soft Computing*, Vol. 91, Springer, 191-198.
4. Fournier, O., Lopez, P., Lan Sun Luk, J.-D., Cyclic scheduling following the social behavior of ant colonies. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2002, 450-454
5. Levner E., Kats V., Alcaide D. Pablo L., T.C.E. Cheng, Complexity of cyclic scheduling problems: A state-of-the-art survey. *Computers and Industrial Engineering*, Vol. 59, Issue 2, 2010, 352-361
6. Liebchen, C., Möhring, R.H., A case study in periodic timetabling. In: *Electronic Notes in Theoretical Computer Science*, Vol. 66 (6), 2002, 21-34
7. Polak M., Majdzik P., Banaszak Z., Wójcik R., The performance evaluation tool for automated prototyping of concurrent cyclic processes. *Fundamenta Informaticae*. In: *ISO Press, Editor-in-Chief, A. Skowron*, Vol. 60, No.1-4, 2004, 269-289
8. Song J.-S., Lee T.-E., Petri net modeling and scheduling for cyclic job shops with blocking. In: *Computers & Industrial Engineering*, Vol.34, No.2, 1998, 281-295
9. Wójcik R., Constraint programming approach to designing conflict-free schedules for repetitive manufacturing processes. *Digital enterprise technology. Perspectives and future challenges*. Ed. by Pedro F. Cunha, Paul G. Maropoulos. Springer, 2007, 267-274