Queueing production system, Petri net

Oleh KUZMIN[*]*, Petro BAZYLEVYCH*[**]

A HIGH-LEVEL PETRI NET MODEL OF QUEUEING PRODUCTION SYSTEM

Abstract

In the paper a high-level Petri model for widespread queueing networks in production systems with many customers, several queues with independent production buffers and different service nodes is presented. Requests can be either single- or polytypic. Such parameters of system as productivity, queue length and waiting time are evaluated.

1. INTRODUCTION

The production systems with two groups of participants: customers of requests (orders, tasks) on the one hand and service nodes from the other are widely used. There can be many customers and service nodes. There can be also many types of requests, they could demand different runtime and need involvement of various profile of service nodes for their realization. This is the queueing system. Such systems are, for example, in flexible manufacturing, traffic engineering, shops, offices, banks, libraries, distributed computer nets, Internet, communication networks and in many others areas. For modeling these systems high level Petri nets that are helpful for qualitative and quantitative analysis can be used. The needs if customers concerning the waiting time are not always satisfied by service nodes' capabilities of existing systems. In this connection some customers are dissatisfied with the period of request realization. The proper modeling that would create an opportunity to predict such situations and to reduce the runtime is essential here.

Targeted opportunities to do this have timed and coloured Petri nets, which are created for simulation systems that have explicit causal relationships [1-9]. Directed bipartite graph is used to describe them. This graph has two groups of vertices: places (position, execution conditions of the event, if), represented

[*] Lviv Polytechnic National University, S.Bandera Str, 13, Lviv, 79013, Ukraine, tel.+380322582673, e-mail:okuzmin@lp.edu.ua

[**] Lviv Polytechnic National University, S.Bandera Str, 13, Lviv, 79013, Ukraine, tel.+380504300050, e-mail:petbazyl1@yahoo.com

by circles, and transitions (events) that are denoted by shelves (bars), as well as directed arcs, each connects only vertices of different types, i.e. place and transition or vice versa. The execution conditions are indicated by token (label) drawn as the black dot in the suitable place. The token in appropriate conditions is removed after the event is completed. Place may reflect the presence of request, material, information, etc., necessary to perform the appropriate operations. The transitions describe the different types of procedures (operations). Petri net allows us to trace the sequence of all operations that occur in the production system, to detect loops, deadlocks, resource requirements for implementation of the production system functions.

The high-level Petri nets are introduced to extend the modeling capabilities with additional options: "colours" and time. "Colours" identify the type of token that describes the properties of the resources required to perform the operation, and time determines the event's duration. It is possible to build a "tree reachability" which specifies the sequence of all possible changes in the system, to detect recurrence deadlocks in which the system may come with its imperfect design, to predict the occurrence of undesirable situations and to exclude the restructuring of the system. Coloured and timed Petri nets are used for scheduling as well as for solving many other problems that occur in production systems [10-15].

It is desirable to create the model of production systems that have many customers with different types of tasks (requests, orders), a number of service nodes with various functional properties and multiple queues. The research goal is to create a description of these types of processes, which allows estimation the service waiting time, the sequences of operations needed to determine the required composition of service nodes and other functions. Such model can be used to create the proper software for practical application.

2. PETRI NET MODEL

Let us consider the possibility of using timed coloured Petri nets to create a model of the production system of type CQS: "Customers (sources of different-type of requests, orders) – Queues – Services (a group of various profiles service nodes)" (Fig. 1). A formal definition of the CQS system is as follows:

$$\text{CQS} = (C, Q, S, Stg), \tag{1}$$

where: $C = \{C1, C2, \ldots, Ck\}$ – the set of customers;
$Q = \{Q1, Q2, \ldots, Qn\}$ – the set of queues;
$S = \{S1, S2, \ldots, Sm\}$ – the set of service nodes;
Stg – Storage.

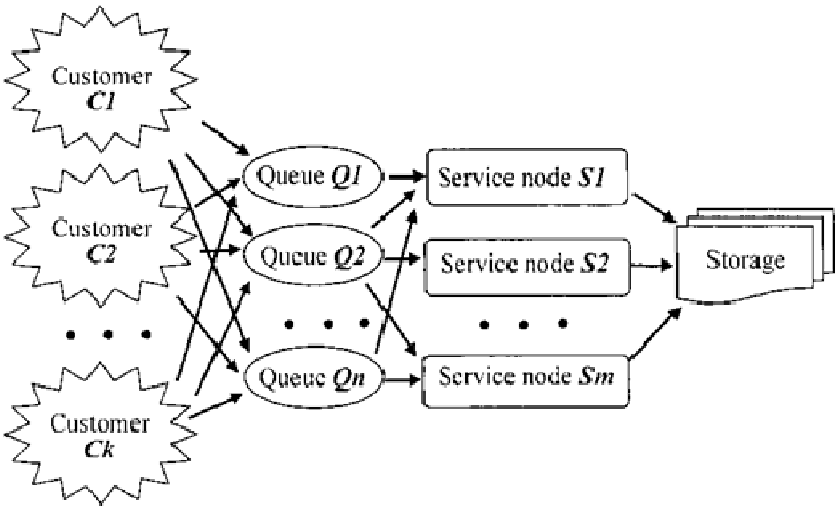

Fig. 1. Schematic representation of the CQS production system [source: own study]

The timed coloured Petri net of this system (Fig. 2) is represented by:

- $C = \{C1, C2, \ldots, Ck\}$ - the set of shelves that represent the customers.
- p_0 – the initial place that imitate the admission of rom customers.
- $T\{\{t_{11}, \ldots, t_{1w}\};\{t_{21}, \ldots, t_{2w}\}; \ldots; \{t_{n1}, \ldots, t_{nw}\}\}$ – the family transitions' sets, each of which represents one node of queue; n is the number of queues; w is the number of nodes in every queue. For simplicity, we accept that the numbers of nodes in all queues are the same. However, they can be different.
- $P\{\{p_{11}, \ldots, p_{1w}\};\{p_{21}, \ldots, p_{2w}\}; \ldots; \{p_{n1}, \ldots, p_{nw}\}\};$
- $P^*\{\{p_{11}^*, \ldots, p_{1w}^*\};\{p_{21}^*, \ldots, p_{2w}^*\}; \ldots; \{p_n^*, \ldots, p_{nw}^*\}\}$ – the two families of sets of places each of which represents one node (independent production buffer) of the queue.
- $T\{t_{l11}, t_{l12}, t_{l13}; t_{l2}; \ldots, t_{lm1}, t_{lm2}\}$ – the set of loading transitions of service nodes as well as for simplicity the time for loading. The number of these transitions for every service node depends of the number of job types that can perform this node. For simplicity, first node in this case has tree such types, second – one and the $-m^{\text{th}}$ – two. The number of service nodes in m.
- $T\{t_{s1}, t_{s2}, \ldots, t_{sm}\}$ – the set of direct service transitions as well as they times.
- $T\{t_{u1}, t_{u2}, \ldots, t_{um}\}$ – the set of unloading transitions as well as they times.

- $P\{\{p_{s11}, p_{s12}, p_{s1}^*\};\{p_{s21}, p_{s22}, p_{s2}^*\}; \ldots; \{p_{sm1}, p_{sm2}, p_{sm}^*\}\}$ – the family of sets of tree places in each service node that represent acceptance of request, it completion and creation the condition for next request taking after completion unloading pre-request.
- p_{stg} represents the storage of executed requests.

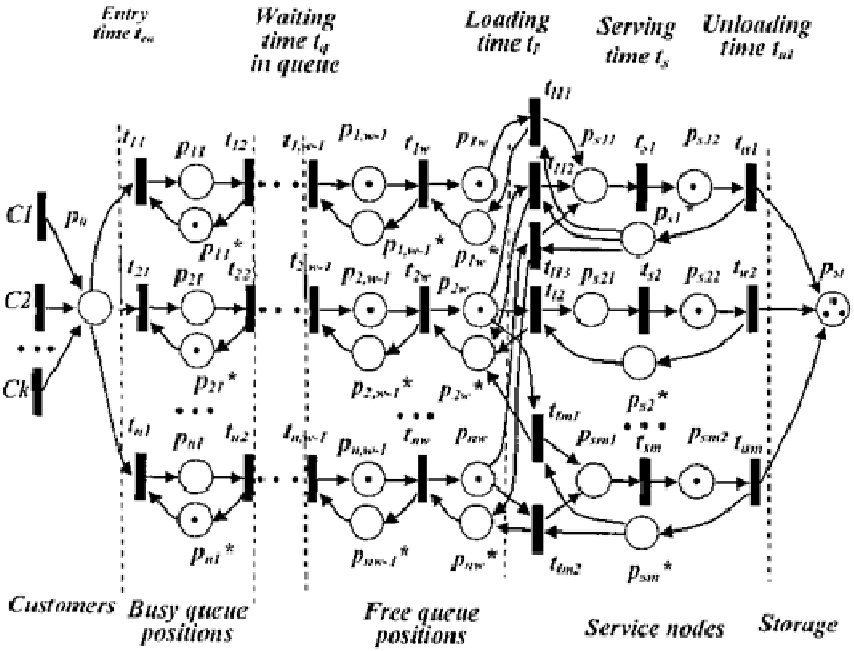

Fig. 2. The timed coloured Petri Net of CQS production system [source: own study]

Every source *C1, C2,..., Ck* generates some types of request that go to the place p_0. The requests cause the appearance of tokens. Each token has three parameters: type φ_i, time of entering the queue $t_{e,i}$ and service time $T_{srv,i}$ of going through all transactions. The first two parameters are fixed during the execution of the request. So they describe the "colour" of token. Time changes during of all transactions: waiting in the queue $t_{q,i}$, loading for serving t_l, direct serving $t_{ds},_i$ and unloading $t_{u,i}$. Every transaction has the parameter – time t_i. In addition, every first transaction in the queues $t_{11}, t_{21},\ldots, t_{n1}$ and every loading for service transaction $t_{l,ij}$ $(i=1,\ldots,m;\ j$ is the number of job types that can perform appropriate service node) have the type ψ_j, $\psi_j = \varphi_i$.

So, the finishing moment of served request appearance in the storage (complication of the work) includes the time of appearance of request $t_{e,i}$ from customer and full service time $T_{srv,i}$:

$$T_{stg} = t_{e,i} + T_{srv,i}, \tag{2}$$

where

$$T_{srv,i} = t_{q,i} + t_{l,i} + t_{ds,i} + t_{u,i}. \tag{3}$$

Check the details of serving process. Each request from sources *Ci* is labelled by entering time $t_{e,i}$ and its type φ_i *(i = 1,…, n).* After formation of the request from customer it falls into the place p_o. Receipt of every request is recorded by a dot in this place. The request has the properties that are described by its token: φ_i and $t_{e,i}$. Request from each source comes in the appropriate queue that matches its type and has an attribute that contains information about the entry time. For every type of request φ_i is formed one queue. So, the number n of queues must be the same as the number of request types. Every queue has the number of nodes that are determined as a maximum possible number of requests that could be generated by the system of all customers *C1,…, Ck.* This amount must be determined at the design stage. At the Figure 2 this number is the same w. Every node of queues is shown by two places p_{ij} and p_{ij}^* *(i=1,…,n; j=1,…,w)* and one transition.

Moving in queue, the request enters the last in queue place p_{iw} and will be accepted for serving when one of the service nodes that can perform this type of request is free. Such possibility will be signalized by a token appearance in position $p_{s,j}$. The system has several various functional service nodes. Each of them takes to implement proposal from queue that match his profile. In the event of conflict the request with an earlier entering time is accepted for serving. Service nodes are depicted by tree transitions: loading with time t_l, direct serving with time t_{ds} and unloading with time t_u, as well as by two places $p_{s,ij}$ *(i=1,…,m; j=1,2)* and one $p_{s,i}$* *(i=1,…,m).* The request is carried out in three steps: loading, the direct serving and unloading. After serving the result is saved in the storage p_{stg} that is reflected by appearance of the token as well as by the token in the place $p_{s,i}$* that create condition for acceptation the new request for serving. Storage collects the results which will be taken out by the customers.

Important characteristics of production system are its productivity, service time and the length of queue. They should be determined at the design stage. Queues are formed when the total productivity of all service nodes is less than the total productivity of all sources of requests. Queues are not formed when these productivities are the same or the productivity of service nods is greater and requests have come evenly and in the same amounts. The moments of request reception can be random, irregular. Not in all cases we have such stability, and at certain intervals of time the total volume of requests can be great comparatively with the production systems possibility. Productivity of pro-

duction system may also vary. For example, reducing the number of service nodes, some machines out of request etc. These will cause the creation of queues. Therefore, the requirements to designer are creating such production system that provides an efficient it functioning in the worst cases. It is necessary to properly evaluate the performance of productive resources – a number of service nodes with desired characteristics, and to prevent the formation of queues longer for the desired value.

We introduce the following notation:

- ω_c^{min}, ω_c, ω_c^{max}, $i = 1,\dots, k$ - respectively the minimum, average and maximum amounts of requests that can be produced by of *i-st* customer per unit time, i.e. it productivity;
- ω_{sj}^{min}, ω_{sj}, ω_{sj}^{max}, $j= 1,\dots,m$ - respectively the minimum, average and maximum amounts of products that can be served by *j-st* node per unit of time, i.e. it productivity.

We define the following characteristics of the production system:

- Ω_c^{min}, Ω_c, Ω_c^{max} – respectively the minimum, average and maximum total amounts of requests that can be produced by all sources per unit time, i.e. their total productivity where
- $\Omega_c^{min} = \sum\omega_{ci}^{min}$, $\Omega_c = \sum\omega_{ci}$, $\Omega_c^{max} = \sum\omega_{ci}^{max}$, $i =1,\dots, k$;
- Ω_s^{min}, Ω_s, Ω_s^{max} – respectively the minimum, average and maximum total volume of products that can be served by the whole system per unit of time - it productivity, where: $\Omega_s^{min} = \sum\omega_{si}^{min}$, $\Omega_s = \sum\omega_{si}$, $\Omega_s^{max} = = \sum\omega_{si}^{max}$, $i =1,\dots, m$.

The conditions of absence of queue for some period of time at a constant formation of requests is $\Omega_s \geq \Omega_c$. If this condition is not satisfied at the times t_i, i.e. $\Omega_s(t_i) < \Omega_c(t_i)$, then the queue is created. The value of generated queue's length will be determined by the duration ΔT of this time.

The worst case is when the customers generate the maximum amount of requests with the performance of Ω_c^{max}, and the production system has the smallest possibility, i.e. its productivity is Ω_s^{min}. Then for interval time ΔT the amount of requests that wait for direct serving is:

$$\Delta R = (\Omega_c^{max} - \Omega_s^{min})\,\Delta T. \tag{4}$$

The number of necessary queue nodes (independent production buffers) to avoid the accumulation of requests in customers for the n queues must be:

$$w_n = \Delta R / n\Delta r, \tag{5}$$

where: Δr – average amount of one request.

Value ΔT must be defined as the maximum allowable waiting time for designing stage of the production system. For banks, cash desks, offices it should be negligible, for industrial systems – according to needs of the customers.

It is also important to determine the waiting time of the request r in the queue from the moment t_i of it appearance to beginning it direct serving:

$$t_q(r) = \Delta R(t) / \Omega_s(t), \tag{6}$$

where: $\Delta R(t)$ – the total amount of all requests of given type, that are waiting for serving at the moment t_i in the appropriate queue,
$\Omega_s(t)$ – the productivity of the system for waiting time which we consider as a constant.

3. CONCLUSIONS

The timed coloured Petri nets are widely used for simulating of production systems by asynchronous discrete models. Such models create conditions for the analysis of their structure, evaluated the waiting time and other characteristics that may be helpful on the stages of designing of new and reengineering of existing systems to improve their efficiency. Such nets are specifically beneficial for systems of the type "Customers – Queues – Services" that are common in various areas. Defined in the paper indicators of performance, including productivity, the waiting time and the length of queues must be considered on design stage.

REFERENCES

[1] PETERSON J.: *Petri net theory and the modelling of systems.* Prentice-Hall, NJ:, 1981.

[2] JENSEN K., ROZENBERG G.: *High-level Petri nets: theory and application.* Springer-Verlag, London, 1991.

[3] DICESARE F., HARHALAKIS G., PROTH J.M., SILVA M., VERNADAT F.B.: *Practice of Petri Nets in Manufacturing,* Chapman and Hall, London, 1993.

[4] JENSEN K., KRISTENSEN L.: *Coloured Petri Nets: Modelling and Validation of Concurrent Systems.* Springer; 2009.

[5] MENG CHU ZHOU; KURAPATI VENKATESH.: *Modeling, simulation, and control of flexible manufacturing systems: a Petri net approach*: Singapore ; River Edge, NJ, World Scientific, 1998.

[6] MANS R.S., RUSSELL N.C., VAN DER AALST W.M.P., MOLEMAN A.J., BAKKER P.J.M.: *Augmenting a Workflow Management System with Planning Facilities using Colored Petri Nets.* Proceedings of the Ninth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, October 2008, Department of Computer Science, University of Aarhus, PB-588, 143-162.

[7] RUSSELL, NICK C., VAN DER AALST W.M.P., TER HOFSTEDE, ARTHUR H. M.: *Designing a workflow system using coloured petri nets.* Lecture Notes in Computer Science: Transactions on Petri Nets and Other Models of Concurrency III, 5800, 2009, pp. 1-24.
[8] BOŻEK A.: *Using Timed Coloured Petri Nets for Modelling, Simulation and Scheduling of Production Systems.* Production Scheduling, Rodrigo Righi (Ed.), 2012, pp. 207-230.
[9] HE X. AND MURATA T.: *High-Level Petri Nets-Extension, Analysis and Applications.* The Electrical Engineering Handbook edited by W.K. Chen, Elsevier Academic Press, Burlington, MA, 2005, pp. 459-475.
[10] TAE-EOG LEE.: *A review of scheduling theory and magnitude for semiconductor manufacturing cluster tools*. Proc 2008 Winter simulation conference, IEEE, pp. 2127-2222.
[11] LOHMANN N., VERBEEK H.M.W., DIJKMAN R.: *Petri Net Transformations for Business Processes. A Survey.* LNCS ToPNoC, II (5460), March 2009. Special Issue on Concurrency in Process-Aware Information Systems. Springer-Verlag. Berlin. Heidelberg. 2009. pp. 46-63.
[12] RÉNE D., HASSANE ALLA.: *Discrete, continuous, and hybrid Petri Nets*, Springer, Berlin; London, 2010.
[13] GONCA TUNCEL, G. MIRAC BAYHAN.: *Applications of Petri nets in production scheduling: a review*. The International Journal of Advanced Manufacturing Technology, Volume 34, Issue 7-8, 2007, pp. 762-773.
[14] ZHANG H., GU M, SONG X.: *Modelling technique and Analysis of Real-life Job Shop Scheduling Problems by Petri Nets*. 41st Annual Simulation Symposium, 2008.
[15] ZUBEREK W.M.: *Petri nets in hierarchical modeling of manufacturing systems.* Proc. IFAC Conf. on Control System Design (CSD'2000), Special Session on Petri Nets in Design, Modeling and Simulation of Control Systems, Bratislava, 2000, pp. 287-292.