

## Research on Modeling and Simulating of Discrete Event System Based on Petri Net

Xinglong Pan\*, Guo He, Chaojie Zhang, Tingfeng Ming, Xiaochuan Wang

College of Naval Architecture and Power, Naval University of Engineering,

717 Jiefang Street, Wuhan, Hubei, 430033, P. R. China

E-mail: paxilo52@tom.com

**Keywords:** discrete event system, Petri net, marine diesel engine, logical control system, simulation

**Abstract.** A framework of modeling and simulating a typical artificial system is proposed based on discrete event system and Petri net. Firstly, the system model is constructed based on discrete event system theory. Secondly, the model is described and analyzed by using Petri net. Then, the simulation procedures on Matlab platform are presented in detail. The proposed framework is applied to modeling and simulating a logical control system of a marine diesel engine. The simulation results indicate that this logical control system model can be constructed by the given framework and the proposed method is effective in simulating and analyzing this kind of artificial system.

### Introduction

Diesel engine is a typical artificial system, which is widely used in vehicle and watercraft and governed by operational rules designed by humans. The control system plays a vital role in guaranteeing the safety and the reliability of a diesel engine. A certain kind of marine diesel engine is the prime mover of the watercraft. The corresponding control system of this marine diesel engine is a logical control system, which is composed of variable relays and sensors and played a complex logical control. Because of the diversiform configuration and complex control logic of this system, it is difficulty to construct the model and analyze the work procedures of this kind of logical control system in traditional methods, such as differential equations and difference equations [1-3]. Discrete Event System (DES) is a dynamical system that evolves according to asynchronous occurrence of certain discrete changes, called events. It is widely used in the field of language and automata theory, supervisory control, Petri net theory, Markov chains and queuing theory, discrete-event simulation, and concurrent estimation techniques. It is can be used for modeling and describing this type of artificial system [3,4].

Generally, there are two kinds of modeling formalisms in describing and simulating a discrete event system: automata and Petri net. The two formalisms have in common that they represent the state transition function by specifying what the possible events are in each state of the system. But the lack of structure hierarchy in automata is an obvious shortcoming. Compared with automata, Petri net is conveniently described graphically. It is intuitive and captures a lot of structural information about a system [3-8].

In this paper, we will focus on modeling and simulating a discrete event system based on Petri net on Matlab platform. The rest of this paper is organized as follows. The general model of discrete event system and corresponding Petri net is introduced in Section 2. Section 3 provides a framework of simulating a Petri net based on Matlab platform and an example is given to illustrate the proposed method. Simulation results and conclusions are summarized in Section 4.

### General Model of a Typical Artificial System

**Discrete Event System (DES) Model.** Generally, a typical artificial system  $G$  can be modeled by discrete event system with [3-5]:

$$G = (\Sigma, Q, \delta). \quad (1)$$

Where,  $\Sigma$  is the set of events, which happened in  $G$ , and a certain event is denoted as  $\sigma$ .

$Q$  is the set of states, which indicates the possible states  $G$  may be stayed, and a certain state is denoted as  $q$ .

$\delta: \Sigma \times Q \rightarrow 2^Q$  is the state transition function, with  $\delta(\sigma, q)$  giving the set of next possible states if  $\sigma$  occurs at  $q$ .

For an event  $\sigma \in \Sigma$ , we define a partition:

$$\Omega_\sigma = \{Q(\sigma), Q - Q(\sigma)\}. \quad (2)$$

Where,  $Q(\sigma) = \{q \in Q : (\exists q' \in Q), q \in \delta(\sigma, q')\}$  is the possible set of states after the event  $\sigma$  occurred. It means that  $\Omega_\sigma$  partitions  $Q$  into two subsets: one with all the possible states after the occurrence of event  $\sigma$ , and the other with all the impossible states after event  $\sigma$  occurred.

**Petri Net Framework.** A Petri net structure is a weighted bipartite graph includes a finite set of places and transitions, the set of arcs from places to transitions and from transitions to places in the graph, and the weight function on the arcs. For the proposed discrete event system model in Eq. 1, the corresponding Petri net structure is as follow [6-8]:

$$CPN = (P, T; F, K, W, M). \quad (3)$$

Where,  $P$  is the finite set of places, the counterpart in DES model is states set  $Q$ . A certain place is denoted as  $p$ , and every place usually contains variable tokens. The number of tokens in a place indicates the information of the place.

$T$  is the finite set of transitions, the counterpart in DES model is events set  $\Sigma$ . When a transition  $t$  occurred, the states of the system are updated. The occurrence of a transition is called fire. The fire conditions for a certain transition can be defined as required.

$F \subseteq (P \times T) \cup (T \times P)$  is the set of arcs from places to transitions and from transitions to places in the graph, equal to the state transition function  $\delta$  in DES model.

$K: S \rightarrow Z^+ \cup \{\infty\}$  is the capacity function of the places. The number of tokens in a place should not exceed the capacity.

$W: F \rightarrow Z^+$  is the weighted function of the arcs.

Where,  $Z^+$  is the set of positive integer.

$M: P \rightarrow \{0, 1, 2, \dots\}, \forall p_i \in P, M(p_i) \leq K(p_i)$  is the marking set of places, the counterpart in DES is the partition  $\Omega$ , which indicates the token number in the places. Compared to  $\Omega$ , according to defining different marking sets, such as colored Petri net,  $M$  can describe many states partition in a Petri net, while  $\Omega$  partitions states set only into two subsets in DES: the possible states and impossible states after a certain event  $\sigma$ .

In describing a Petri net, it is convenient to use  $I(t_j)$  to represent the set of input places of transition  $t_j$ . Similarly, the set of output places from transition  $t_j$  is indicated as  $O(t_j)$ . Thus, we can define

$$I(t_j) = \{p_i \in P : (p_i, t_j) \in F\}, \text{ and } O(t_j) = \{p_i \in P : (t_j, p_i) \in F\}.$$

Similar notations  $I(p_i)$  and  $O(p_i)$  can be used to describe the input and output transitions for a given places  $p_i$ , which defined as

$$I(p_i) = \{t_j \in T : (t_j, p_i) \in F\}, \text{ and } O(p_i) = \{t_j \in T : (p_i, t_j) \in F\}.$$

There are many Petri net simulation tools, in this paper, we will use the General-Purpose Petri net simulator toolbox (GPenSIM toolbox) based on the Matlab platform, which is developed by Reggie Davidrajuh [9,10].

The procedures of simulating a Petri net graph are as follow:

Step 1: Define the static Petri net graph, including places, transitions, and arcs.

Step 2: Initialize the Petri net graph, including state markings, fire conditions and fire times.

Step 3: Start simulation.

Step 4: Output the simulation results.

Transition definition files will be called during simulation. Thus, if we want to inspect the simulation properties a transition definition file can be used as a probe to control and inspect the simulation processes.

### Example Analysis of a Marine Diesel Engine Control System

**Model Construction.** A certain kind of logical control system is used for controlling the work processes of a marine diesel engine, such as start-up, stopping, timing, supervision, alarming, and interlocking protection. Take the start-up process as an example, we will construct the discrete event system model and Petri net graph, and then simulate the starting procedure of this system on Matlab platform.

The control system of this marine diesel engine is a kind of logical control system, composed of variables relays and sensors. The states of these relays and sensors indicate the rotate speed of this marine diesel engine. Thus, we focus on controlling the rotate speed of the diesel engine and simulating the work schedules of the corresponding external sensors and relays in different rotate speed. It is obvious that the continuous rotate speed should be divided into different discrete parts based on the different work parameters of the external sensors and relays.

The discrete event system model of the diesel engine control system is constructed based on Eq. 1, the events and state spaces are as shown in Table 1 and Table 2, respectively.

Table 1 Events of the logical control system

events	meaning
$\sigma_1$	press the start-up button.
$\sigma_2$	the fuel pressure sensor worked, which means the rotate speed rising to a certain value ( $n_1$ ) and the fuel pressure is normal.
$\sigma_3$	the cooling water pressure sensor worked, which means the rotate speed rising to a certain value ( $n_2$ ) and the cooling water pressure is normal.
$\sigma_4$	the start-up relay power off, which means the processes of starting the diesel engine is completed and the start-up air is cut off.

The events in this system are defined as a button is pressed, relay and sensor are switch-on or switch-off.

Table 2 State spaces of the logical control system

state	name	value	meaning
$q_1$	state before start-up	1	enable start-up
		0	unable start-up
$q_2$	start-up air	1	supply start-up air
		0	cut off start-up air
$q_3$	lubricating oil pump	1	running
		0	stopping
$q_4$	state of rotate speed	1	rising

state	name	value	meaning
$q_5$	rotate speed 1	0	holding
		1	higher than $n_1$
		0	lower than $n_1$
$q_6$	rotate speed 2	1	higher than $n_2$
		0	lower than $n_2$
$q_7$	fuel pressure	1	normal value
		0	lower than normal value
$q_8$	cooling water pressure	1	normal value
		0	lower than normal value
$q_9$	result of start-up	1	completed
		0	uncompleted

Where, the sign  $n_1$  denotes the rotate speed when the fuel pressure rising to the normal value, which means the fuel pressure sensor have worked. The sign  $n_2$  denotes the rotate speed when the cooling water pressure rising to the normal value, which means the cooling water pressure sensor have worked, that is  $n_2 > n_1$ .

Then, we can construct the corresponding Petri net, as shown in Fig. 1.

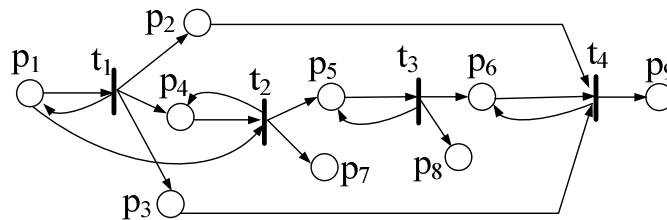


Fig. 1 The Petri net graph of the logical control system

Where,  $p_1 \sim p_9$  and  $t_1 \sim t_4$  are the same as  $q_1 \sim q_9$  and  $\sigma_1 \sim \sigma_4$  in the discrete event system model, respectively.

**Simulation.** After constructing the model of this marine diesel engine control system, we can simulate the work processes of this system on Matlab platform.

Firstly, we define the Petri net graph as follow:

$set\_of\_places = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\};$

$set\_of\_transitions = \{t_1, t_2, t_3, t_4\};$

$set\_of\_arcs = \{p_1, t_1, 1; p_1, t_2, 1; p_2, t_4, 1; p_3, t_4, 1; p_4, t_2, 1; p_5, t_3, 1; p_6, t_4, 1; t_1, p_1, 1; t_1, p_2, 1; t_1, p_3, 1; t_1, p_4, 1; t_2, p_4, 1; t_2, p_5, 1; t_2, p_7, 1; t_3, p_5, 1; t_3, p_6, 1; t_3, p_8, 1; t_4, p_6, 1; t_4, p_9, 1\}.$

Secondly, define the fire conditions of transitions.

To simulate the practical work procedures of this system, we define the fire conditions as Eq. 4.

$$M(p_j) = \begin{cases} 1, & p_j \in I(t_i) \\ 0, & p_j \in O(t_i) \end{cases} \quad i \in \{1, 2, \dots, 4\}, j \in \{1, 2, \dots, 9\} \quad (4)$$

Thirdly, initialize the Petri net graph.

The initial state markings  $M_0$  is defined as:  $M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ .

The fire times of transitions is defined as:  $fire\_times = \{t_1, 4; t_2, 2; t_3, 2; t_4, 2\}$ .

Then, start to simulate and output the simulation results. The results are shown in Fig. 2.

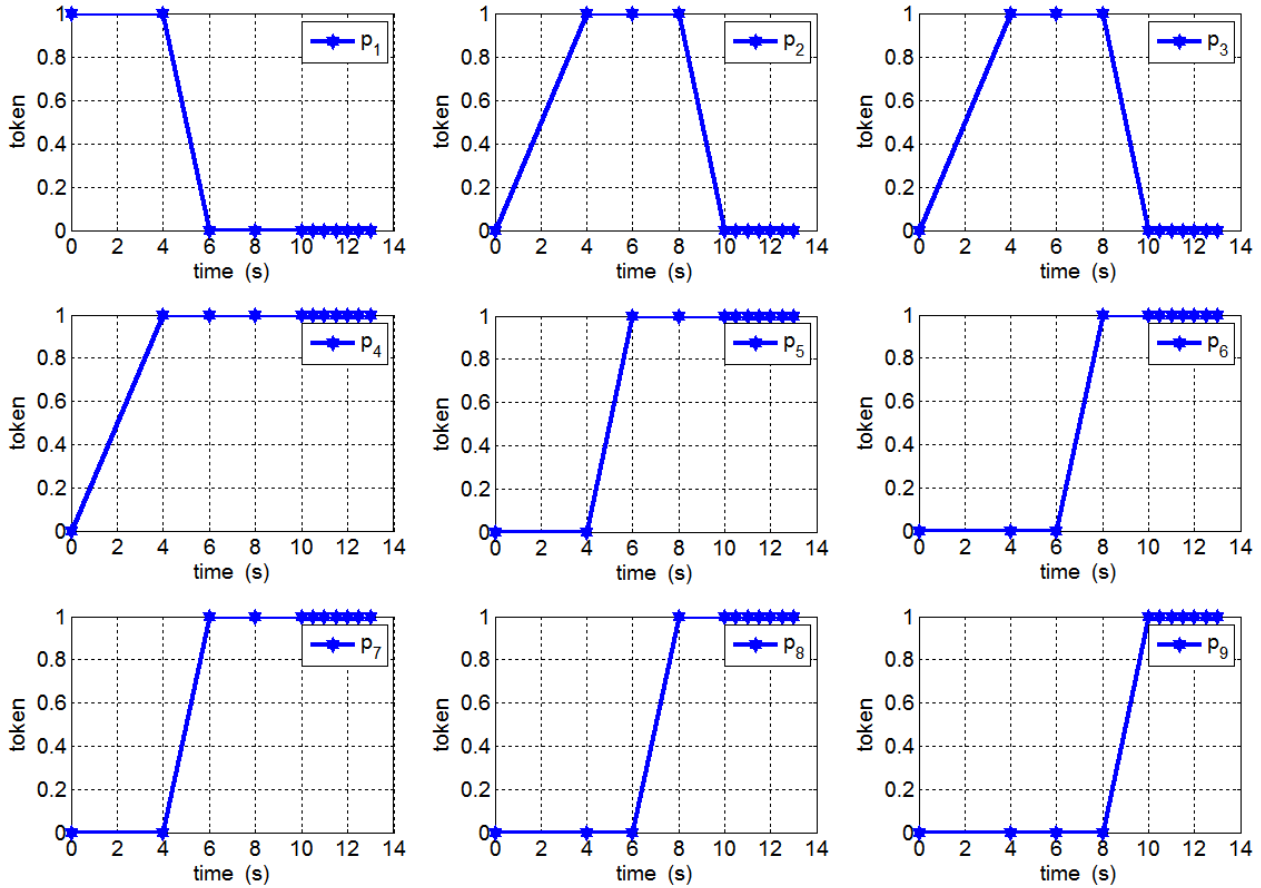


Fig. 2 Simulation results of the diesel engine control system

The simulation results indicate that when starting the simulation ( $p_1, 0 \text{ second}$ ) the number of token in  $p_1$  is 1, which means the initial state of the system is enable start-up. At time 4 second, the event  $t_1$  fired and the start-up button pressed. At this time,  $p_1$  has no token, while  $p_2$ ,  $p_3$ , and  $p_4$  have 1 token respectively. It means that the system is supplying start-up air ( $p_2$ ) and lubricating oil ( $p_3$ ), and rotate speed of the diesel engine is rising ( $p_4$ ). At time 6 second, event  $t_2$  fired and the fuel press sensor switch on. At the same time,  $p_5$ , and  $p_7$  contains 1 token, respectively. Which indicate that the rotate speed is higher than  $n_1$  ( $p_5$ ) and the fuel pressure has reached to normal value ( $p_7$ ). At time 8 second, event  $t_3$  fired and the cooling water press sensor switch on. At the same time,  $p_6$ , and  $p_8$  contains 1 token, respectively. Which means the rotate speed is higher than  $n_2$  ( $p_6$ ) and the cooling water pressure has reached to normal value ( $p_8$ ). At time 10 second, event  $t_4$  fired, which means start-up relay is power off. At the same time,  $p_9$  appears a token and the token in  $p_2$  and  $p_3$  disappear, which means the processes of starting the diesel engine is completed ( $p_9$ ) and start-up air is cut off ( $p_2$ ) and the lubricating oil pump is power off ( $p_3$ ).

## Conclusion

In this paper, we proposed a method of modeling discrete event system based on Petri net. Also, the simulation procedures on Matlab platform are given. This method is applied to analyzing and simulating a marine diesel engine control system. This system is a logical control system, which can not be modeled and described exactly by the conventional ordinary and partial differential equations. The experiment result indicates that this marine diesel engine control can be analyzed conveniently and exactly by using a discrete event system model and a Petri net graph on Matlab platform. It is concluded that the proposed framework is effective in modeling and simulating this kind of system.

---

## Acknowledgment

This paper is supported by the Innovation Research Foundation for Ph. D Candidates of Naval University of Engineering, P. R. China (No. HGBSJJ2011003). First Author is corresponding author.

## References

- [1] A. J. C. Sharkey, G. O. Chandroth, N. E. Sharkey, A Multi-Net System for the Fault Diagnosis of a Diesel Engine, *Neural Computing & Applications*, Vol.9, No.2 (2000), pp. 152-160.
- [2] L. Guzzella, C. H. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems*, Springer, New York, USA, (2009).
- [3] X. Pan, G. He, C. Zhang, Fault Diagnosis of Relay Control Circuit Based on Discrete Event System, *International IEEE Workshop on Intelligent Systems and Applications*, Wuhan, China, Vol.2, No.1 (2009), pp. 973-976.
- [4] C. G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems*, Springer-Verlag New York, Inc. Secaucus, NJ, USA, (2008).
- [5] L. Feng, Diagnosability of Discrete Event System and Its Application, *Discrete Event Dynamic Systems*, Vol.4, No.2 (1994), pp. 197-212.
- [6] M. V. Iordache, P. J. Antsaklis, *Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*, Birkhauser Boston, (2006).
- [7] B. Hruz, M. Zhou, *Modeling and Control of Discrete- Event Dynamical System: with Petri Nets and other Tool*, Springer, (2007).
- [8] X. Pan, G. He, Research on the Application of Discrete Event System Based on Controlled Petri Net, *2011 Third International Conference on Measuring Technology and Mechatronics Automation*, Shanghai, China, Vol.3 (2011), pp. 91-94.
- [9] R. Davidrajuh, GPenSIM: A New Petri Net Simulator, *Petri Net Applications*, InTech publisher (2010).
- [10] Information on <http://www.davidrajuh.net/gpensim/>.