Modeling and Analysis of ETC Control System with Colored Petri Net and Dynamic Slicing

WANGYANG YU, Key Laboratory of Modern Teaching Technology, Ministry of Education, China; The School of Computer Science, Shaanxi Normal University, China
JINMING KONG*, Puyang Vocational and Technical College, China; Puyang Institute of Technology, Henan University, China
ZHIJUN DING, The Department of Computer Science and Engineering, Tongji University, China
XIAOJUN ZHAI, The School of Computer Science and Electronic Engineering, University of Essex, UK
ZHIQIANG LI and QI GUO, The School of Computer Science, Shaanxi Normal University, China

Nowadays, an Electronic Toll Collection (ETC) control system in highways has been widely adopted to smoothen traffic flow. However, as it is a complex business interaction system, there are inevitably flaws in its control logic process, such as the problem of vehicle fee evasion. Even we find that there are more than one way for vehicles to evade fees. This shows that it is difficult to ensure the completeness of its design. Therefore, it is necessary to adopt a novel formal method to model and analyze its design, detect flaws and modify it. In this paper, a Colored Petri net (CPN) is introduced to establish its model. To analyze and modify the system model more efficiently, a dynamic slicing method of CPN is proposed. First, a static slice is obtained from the static slicing criterion by backtracking. Second, considering all binding elements that can be enabled under the initial marking, a forward slice is obtained from the dynamic slicing criterion by traversing. Third, the dynamic slicing of CPN is obtained by taking the intersection of both slices. The proposed dynamic slicing method of CPN can be used to formalize and verify the behavior properties of an ETC control system, and the flaws can be detected effectively. As a case study, the flaw about a vehicle that has not completed the payment following the previous vehicle to pass the railing is detected by the proposed method.

CCS Concepts: • **Theory of computation → formal languages**.

Additional Key Words and Phrases: Colored Petri net, ETC control system, dynamic slicing, formalized analysis

ACM Reference Format:
Wangyang Yu, Jinming Kong, Zhijun Ding, Xiaojun Zhai, Zhiqiang Li, and Qi Guo. . Modeling and Analysis of ETC Control System with Colored Petri Net and Dynamic Slicing. , ( ), 28 pages. https://doi.org/XXXXXXX.XXXXXXX

---

[*] Corresponding author

Authors' addresses: Wangyang Yu, ywy191@snnu.edu.cn, Key Laboratory of Modern Teaching Technology, Ministry of Education , Xi'an, Shaanxi, 710062, China; and The School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi, China, 710119; Jinming Kong, kjm@snnu.edu.cn, Puyang Vocational and Technical College, Puyang, Henan, 457000, China; and Puyang Institute of Technology, Henan University, Puyang, Henan, China, 457000; Zhijun Ding, The Department of Computer Science and Engineering, Tongji University, Shanghai, China, 201804, zhijun_ding@hotmail.com; Xiaojun Zhai, The School of Computer Science and Electronic Engineering, University of Essex, Essex, UK, xzhai@essex.ac.uk; Zhiqiang Li, lizq@snnu.edu.cn; Qi Guo, gqojbk@163.com, The School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi, China, 710062.

1 INTRODUCTION

ETC control systems are currently the most advanced toll collection way in highways in the world. It refers to, when a vehicle arrives at an electronic toll station in highways, they recognize the vehicle utilizing an on-board unit (OBU) installed in the vehicle in advance, then write the related information about entrance address into the IC card in OBU, and finally automatically deduct tolls from a pre-bound bank account or IC at the exit in highways [1–3]. They have brought great convenience to drivers.

As a distributed system, an ETC control system is complex and loosely coupled. The complex integration process among multi-participants and the complex operation process of its each sub-system pose much difficult to ensure the completeness of its control logic design. A wrong design may cause the logical flaws. For example, a vehicle can explore them via some special behaviours, and thus avoid paying fees [4]. As a result, sometimes the charging work cannot be completed according to a certain regulation [5–7].

Several studies have shown that the special behaviors of vehicle fee evasion include reversing IC card, counterfeiting special vehicle, storming the toll lane, tampering with OBU information, and so on [4, 6–9]. As for storming the railing, there are two means: the vehicle that has not completed the payment passes the railing by following the previous vehicle and the vehicle is recognized by the roadside units (RSU) before the previous vehicle [4]. In view of these various means of evading fees, it is an urgent matter to propose useful solutions, and it is also the issue that most scholars are currently addressing.

Scholars have never stopped exploring the solution to vehicle fee evasion. The improvement of ETC technology has been a research focus [10–15]. Scholars are committed to improving key technologies used to realize ETC control systems, such as automatic vehicle identification, identity classification, transaction processing, and license plate recognition [10]. Lu et al. [12] discussed the design of a GPS-based ETC system in detail, which resolved the vehicle traveling multi-path identification problem by calculating the traveling distance; Randriamasy et al. [13] proposed a method of high-precision positioning and connection of vehicles through ITS-G5 technology to perform ETC transaction services; Ref. [14] improved an ETC system based on Radio frequency identification system based on the current issue at the ETC system; Jain et al. [15] studied a unique identity based automated toll collection system using RFID and image processing techniques to offer a hassle-free, cashless, secure and reliable payment at the toll center.

As recent research trends, the methods of data mining and big data analysis are used to predict vehicle fee evasion behavior and avoid financial losses to a certain extent [16–24]. Specifically, Qu et al. [17] predicted an evasion behavior about changing the IC card using the C4.5 decision tree algorithm. Their work is only limited to predict the evasion behavior about changing the IC card only; Ref. [18] used K-means clustering and Density-Based Spatial Clustering of Applications with Noise to mine highway toll-evading vehicles; Lin et al. [19] used logistic regression analysis to establish a toll evasion behavior prediction model to predict the likelihood of each toll card. But its ability to identify non-overtime toll evasion behaviors is insufficient; Wang et al. [20] summarized the methods of vehicle fee evasion on expressways in Henan Province, and then used K-means and logistic regression algorithm to establish a behavior recognition model, which can be reasonably predicted to a certain extent, but it is easy to fall into the local optimum.

The above-mentioned studies either improve key technologies of ETC or predict the future behavior of vehicles to establish preventive measures. Regrettably, the previous studies rarely concern its entire control logic process that completes the charging work by controlling the state of the railing. The logic flaws may still exist in an ETC control system.

Formal methods can be used to discover flaws in the logical design of a system and have been proven to be effective in reducing design flaws and improving system reliability [25]. Yet it is rare to model and analyze the control process of an ETC control system with formal methods.

There are many kinds of formal methods, including automata, transition systems, and Petri nets [26–31]. Petri nets have been widely used in detecting flaws in e-commerce transaction process [29, 32, 33]. In this paper, we introduce CPN to model and analyze the control process of an ETC control system. CPN can graphically express the concurrency, sequence, conflict, and synchronization relations for a system, and has a set of analysis methods. CPN can be used to model, simulate and analyze various complex distributed concurrent systems, especially control systems [34, 35]. Thus, CPN is suitable to depict the logical relationships among various objects in an ETC control system.

The formal analysis methods of CPN include state space analysis method and invariant method that refers to the use of place/transition invariants to prove a property [36]. When a system model is relatively complex and large, the method of enumerating states becomes impractical, and state-space explosion makes formal verification difficult, and there is often a part of the reachable state space that does not need to be generated. Therefore, it is necessary and meaningful to propose a more efficient formal analysis method for CPN.

Recently, the slicing idea has been applied to Petri nets as a formal analysis method [37–46]. The slicing methods are divided into two categories: static and dynamic ones. Both are subdivided into: backward and forward slicing [47]. Static slicing only takes the set of places of interest as a slicing criterion and does not consider the initial marking of a net. Its basic idea is to trace back the relevant parts of the model according to the slicing criterion [38–41]. Later, some scholars discovered that it was also useful to consider the initial marking. Llorens et al. [42] firstly proposed two dynamic slicing methods for a Petri net. Later, Yu et al. [43] improved the dynamic slicing method based on Ref. [42], which firstly constructed the structural dependency graph by backtracking according to the slicing criterion. These slicing algorithms in [37–44] are all for low-level Petri nets, and now as an evolution of low-level Petri nets, slicing methods for high-level Petri nets are also being further explored. Khan et al. [45] propose slicing algorithms for Algebraic Petri net to improve the model checking and testing of systems modeled using Algebraic Petri net. P Chariyathitipong et al. [46] propose an alternative dynamic slicing algorithm written as a metric temporal logic (MTL) formula to reduce the size of the Time Petri net model by considering specific criteria. However, as far as we know, as a high-level Petri net, CPN has alternative net elements and complex definitions. So there is no effective slicing method for CPN now.

In summary, regarding modeling and analysis of ETC control systems at the design phase, this paper introduces CPN to model its control logic process. We attempt to introduce the slicing idea into CPN and propose a dynamic slicing method to analyze whether there are flaws in an ETC control system design. We aim to make the following contributions:

1) We propose a formal modeling method based on CPN for the control logic process of an ETC control system, and take a case study which derived from real scenes as an example to illustrate the modeling process.

2) We propose a dynamic slicing method for CPN. This method can reduce the size of the CPN model and reduce the size of state spaces of the model, which can be used to detect the flaws in ETC control logic process.

3) We illustrate how to analyze whether there are logic flaws in the ETC control system model, i.e., faulty states for vehicle fee evasion. As a case study, we detect whether there is a flaw that the vehicle that has not completed the payment passing the railing by following the previous vehicle in the ETC control system, and give a specific modification plan.

The rest of this paper is organized as follows: Section II describes the business process of an ETC control system; Section III introduces the modeling method and gives the CPN model of an ETC control system; the dynamic slicing method is proposed in Section IV. Section V includes a case study and discussion. Finally, Section VI concludes this paper.

2 BUSINESS PROCESS OF ETC CONTROL SYSTEM

The business interaction process between an ETC control system and a vehicle is complicated. In this section, we present an ETC control system derived from real scenarios. We only focus on its most important logic process.

When a vehicle enters an ETC lane, the business process of an ETC control system is as follows:

1) When a vehicle enters the detection coil, a detection coil detects it. Then a roadside unit (RSU) is started to communicate with its OBU and reads the information to make a valid judgment, and then reads the relevant information in the IC card inserted in OBU to judge whether the card is valid or not, finally judges whether the vehicle information stored in the two is consistent. If all the above conditions are met, the transaction is continued; otherwise, an alarm is sent out and the vehicle is switched to a manual processing mode.

2) After completing the transaction, the computer system sends an upward command to a railing controller. At the same time, the vehicle enters the capture coil, a license plate recognition system is triggered to recognize whether the license plate information is legal or not. If it is not, the vehicle is transferred to a manual processing mode.

3) The vehicle enters the drop coil finally. There is a railing on it. If there is another vehicle behind the vehicle, the railing continues to be raised when the vehicle has passed the railing and the railing has not completely landed. When there is no vehicle passing through the lane, the railing is dropped.

The business processes of an ETC control system in the entrance and exit of a highway are roughly the same. At the entrance, it writes the entrance information of the vehicle into the IC card. While at the exit, it calculates and collects vehicle tolls. The transaction between it and the vehicle is completed at the exit. Hence, we focus on its business process at the exit, as shown in Fig. 1.

This control logic process seems perfect. However, when a vehicle follows the previous vehicle into an ETC lane, the distance between the two vehicles is too small. For RSU, sometimes its communication area may be large and its ability to accurately recognize a vehicle may be poor. These cause it to only communicate with the previous vehicle to complete the transaction, but let the vehicle following the vehicle pass the railing without paying fees. *Vehicle Identification* can not guarantee that the OBU of every vehicle inside the communication area of RSU can be communicated. *Control Rail Up* fails to verify which vehicle passed. When the railing has not completely landed, if the railing senses the vehicle following the previous vehicle, the railing remains raised. It is difficult to design a completely flawless control mechanism in the business process of an ETC control system. Therefore, we suppose that the business process in Fig. 1 is a control mechanism specification, and use it as a case study to illustrate our method in this paper.

3 MODELING METHOD

3.1 Colored Petri net

CPN is a graphical language for constructing concurrent system models and analyzing their properties. CPN ML language is based on the functional programming language Standard ML (SML). CPN ML embeds the Standard ML language and extends it with constructs for defining color sets and declaring variables [36]. Compared with an original Petri net, CPN reduces the complexity of modeling. A CPN model is usually more graphically concise. It is suitable for modeling the business

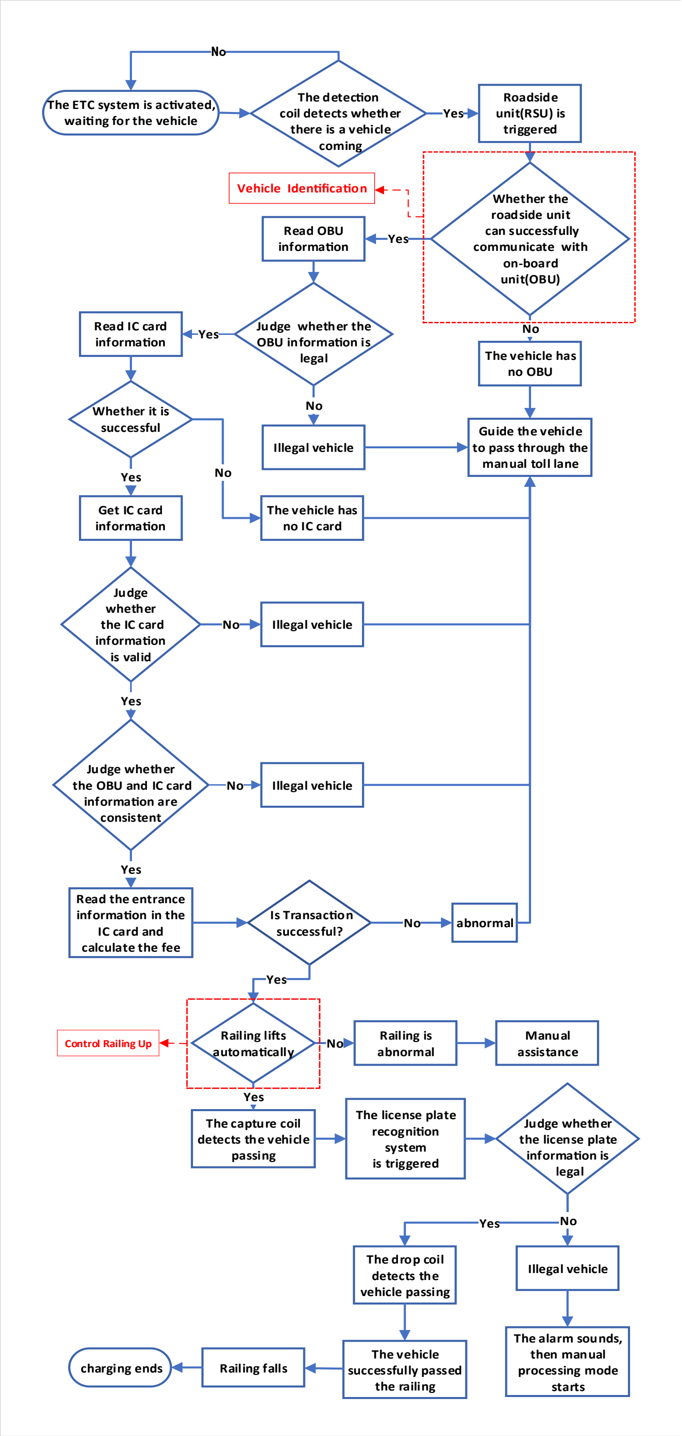


Fig. 1. Business flow chart of an ETC control system at the exit of a highway

interaction process of an ETC control system. Moreover, it has some unique advantages. It can not only dynamically simulate its business process, but also record the real-time information generated by the system. Compared with other modeling tools, it is easier for us to observe its control process, and use the formal methods to analyze whether the system model is correct.

Definition 3.1. [36]. A CPN is a nine-tuple $CPN = (P, T, A, \Sigma, V, C, G, E, I)$, where:

1) P is a finite set of places, which can not only portray the state of ETC control system, but also store data.

2) T is a finite set of transitions such that $P \cap T = \emptyset$, which portrays the behaviors of ETC control system, and events that occur during the system operation.

3) $A \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs from place to transition and from transition to place, which indicates where the tokens flow.

4) Σ is a finite set of non-empty color sets.

5) V is a finite set of typed variables such that $Type[v] \in \Sigma$ for all variables $v \in V$.

6) C: $P \longrightarrow \Sigma$ is a color set function that assigns a color set to each place.

7) G: $T \longrightarrow \Lambda_V$ is a guard function that assigns a guard to each transition t such that $Type[G(t)] = Bool$.

8) E: $A \longrightarrow \Theta_V$ is an arc expression function that assigns an arc expression to each arc a such that $Type[E(a)] = C(p)_{MS}$, where p is the place connected to arc a.

9) I: $P \longrightarrow EXPR_{MS}$ is an initialization function that assigns an initialization expression to each place p such that $Type[I(p)] = C(p)_{MS}$.

Definition 3.2. [36]. For a Colored Petri net $CPN{=}(P, T, A, \Sigma, V, C, G, E, I)$, we define the following concepts:

1) A marking is a function M that maps each place $p \in P$ into a multiset of token $M(p) \in C(p)_{MS}$.

2) The initial marking M_0 is defined by $M_0(p) = I(p)$ for all $p \in P$, and indicates the input of the model.

3) The variables of a transition t are denoted $Var(t) \subseteq V$ and consist of the free variables appearing in the guard of t and in the arc expressions of arcs connected to t.

4) A binding of a transition t is a function b that maps each variable $v \in Var(t)$ into a value $b(v) \in Type[v]$. The set of all bindings for a transition t is denoted $B(t)$.

5) A binding element is a pair $\langle t, b\rangle$ such that $t \in T$ and $b \in B(t)$. The set of all binding elements $BE(t)$ for a transition t is defined by $BE(t) = \{\langle t, b\rangle | b \in B(t)\}$. The set of all binding elements in a CPN model is denoted BE.

Definition 3.3. [36]. For a CPN model, the enabling and occurrence of a binding element is defined as follows:

A binding element $\langle t, b\rangle \in BE$ is enabled in a marking M if and only if the following two properties are satisfied:

1) $G(t)\langle b\rangle$.

2) $\forall p \in P : E(p, t)\langle b\rangle <<= M(p)$.

When $\langle t, b\rangle$ is enabled in M, it may occur, leading to the marking M' defined by:

3) $\forall p \in P : M'(p) = (M(p) - -E(p, t)\langle b\rangle) + +E(t, p)\langle b\rangle$.

3.2 Process modeling

This section presents the modeling scheme of the business process of an ETC control system at the exit of a highway. Our modeling purpose is to analyze whether there are faulty states in the system model. If so, the model needs to be modified.

While modeling an ETC control system, we adopt the method of modular modeling. If a complex system can be easily divided into different modules or subsystems with different functions, we can first establish a CPN model for each separated module or subsystem, and then obtain the CPN model of the entire system through CPN composition operations. Based on the business process of the ETC control system introduced in Section II, we need the following three modeling steps:

3.2.1 *Modeling possible states and events in the process of vehicle driving.*

Before entering an ETC lane, a vehicle must be installed with a legal OBU device and a legal IC card. OBU stores vehicle information. The IC card not only stores the vehicle information, but also records the entrance address information. The above relevant information is all stored in the CPN places. The relevant color set is defined as follows:

colset CarinfxEn = list carinformationxEntrance;

*colset carinformationxEntrance = Product carinformation * Entrance;*

*colset carinformation = record car: Car * obusignal: OBUsignal * obu: OBU * icsignal: ICsignal * ic: IC;*

*colset Entrance = record road: Road * tollname: Tollname * owner: ownername * ct: cartype * num: carnumber;*

var ci: carinformation;

var entrance: Entrance;

var carinfxen, carinfxen1, carinfxen2, carinfxen3 : CarinfxEn;

List, *Product*, and *record* are the data types in CPN. In CPN, *list* can be used to describe the order and repetition of elements in a color set. *Product* represents the Cartesian product of sets of colors. In CPN, the *Product* type can be used to describe the combination of multiple color sets. *Record* represents a structured collection of colors where each element has a unique label. In CPN, the record type can be used to describe the properties and structure of different elements in a color set. We use the color set constructor *record* to record vehicle information, including license plate information, OBU signals, information stored in OBU, IC card signals, and vehicle information stored in IC cards. The information of the entrance address is also recorded by the color set constructor *record*, including the entrance address, name of the road, the vehicle owner and license plate number.

The initial marking $M_0(P_1)$ = 1' [({car = {owner = " Jack ", ct = 1, num = " 123456 "}, obusignal = 1, obu = {owner = " Jack ", ct = 1, num = " 123456 "}, icsignal = 1, ic = {owner = " Jack", ct = 1, num = " 123456 "}}, {road = " 123 ", tollname = " Tongguan toll station" , owner = " Jack ", ct = 1, num = " 123456 "}), ({car = {owner = " Ana", ct = 2, num = " 234567 "}, obusignal = 1, obu = {owner = " Ana", ct = 2, num = " 234567 "}, icsignal = 1, ic = {owner = " Ana ", ct = 2, num = " 234567 "}}, {road = " 123 ", tollname = " Tongguan toll station" , owner = " Ana ", ct = 2, num = " 234567 "})], means that two vehicles are queuing to enter the detection coil.

About the first vehicle, its license plate information includes: the owner is Jack, the type is 1, the license plate number is 123456; the signal of the OBU is normal, and its storage information is consistent with the license plate information; the IC card signal is also normal, and the stored vehicle information is consistent with the license plate information; the name of the road where the entrance is located is 123, the name of the entrance toll station is Tongguan toll station , and the other entrance records are consistent with the license plate information. As for the second vehicle, its license plate information includes: the owner is Ana, the type is 2, the license plate number is 234567; the signal of OBU is normal, its storage information is consistent with the license plate information; the signal of IC card is also normal, and the stored vehicle information is consistent with the license plate information; the name of the road where the entrance is located is 123, the name of the entrance toll station is 2, the license plate number is 234567, and the remaining entrance records are consistent with the license plate information.

After the vehicle enters the ETC lane, it passes through the detection coil and the capture coil in turn, while the ETC control system identifies the vehicle and complete the online transaction. In the drop coil, the vehicle passes through the automatic railing, and after the vehicle leaves the drop coil, the rail lands. See Table 4 in Appendix for the meaning of the places and transitions involved in this module model.

3.2.2 *Modeling the states and events that occur inside an ETC control system.*

While a vehicle is driving on the detection coil, the system checks the legitimacy of the information of OBU and IC card, calculates fee and deducts the fee from IC card, and then raises the rail. After the vehicle enters the capture coil, the license plate recognition system is triggered, and the system checks the legality of the license plate information in order to prohibit driving without a license plate.

In the process of judging the legality of a vehicle, a total of four information checks are involved, i.e., checking the legality of the information stored in the OBU, checking the legality of the information stored in the IC card, checking whether the information stored in the OBU and the IC card are consistent, and checking whether the license plate information is legal. In the process of checking the information, once a vehicle is found to be illegal, this case is immediately transferred to a manual processing mode. The names and color sets of some places used to store this information are shown in Table 1.

Table 1. Color sets of some places.

Places	Color sets	Definitions
OBU information(P_{12})	OBU	= record owner: ownername * ct: cartype * num: carnumber;
database 1(P_{13})	Database	= list database; (colset database = record owner: ownername * ct: cartype * num: carnumber)
legal IC information(P_{18})	IC	= record owner: ownername * ct: cartype * num: carnumber;
carnumber (P_{43})	Car	= record owner: ownername * ct: cartype * num: carnumber;

After the vehicle enters the drop coil, it passes through the railing. When it leaves the drop coil, the railing automatically drops. See Table 5 in Appendix for the meaning of the places and transitions involved in this module model.

3.2.3 *Modeling the control system of the railing and the state of the railing.*

Before an ETC control system is activated, the railing is in a landing state. After a vehicle pays toll successfully, the computer system sends a upward command to the railing controller; when the vehicle passes the railing and leaves the drop coil, the railing drops. It is worth noting that if there is another vehicle behind the vehicle at this time, the railing continues to rise. See Table 6 in Appendix for the meaning of the places and transitions involved in this module model.

Finally, we combine the three modules to establish an ETC control system in highways model. To make it clear, we have circled the range of the three modules in Fig. 2. The three module models are circled with red, green, and blue dashed lines, respectively.

In the process of synthesizing the three modules, we utilize the idea of composition method from original Petri net, including synchronous and shared compositions. There are no clear definitions of shared and synchronous compositions of CPN, so we define them as follows.

Definition 3.4. Assume $CPN_i = (P_i, T_i, A_i, \Sigma_i, V_i, C_i, G_i, E_i, I_i)(i = 1, 2)$,

1) if $P_1 \cap P_2 \neq \emptyset, T_1 \cap T_2 = \emptyset$, then $CPN = (P_1 \cup P_2, T_1 \cup T_2, A_1 \cup A_2, \Sigma_1 \cup \Sigma_2, V_1 \cup V_2, C_1 \cup C_2, G_1 \cup G_2, E_1 \cup E_2, I_1 \cup I_2)$, which is the sharing composition net of the CPN_1 and CPN_2.

2) if $P_1 \cap P_2 = \emptyset, T_1 \cap T_2 \neq \emptyset$, then $CPN = (P_1 \cup P_2, T_1 \cup T_2, A_1 \cup A_2, \Sigma_1 \cup \Sigma_2, V_1 \cup V_2, C_1 \cup C_2, G_1 \cup G_2, E_1 \cup E_2, I_1 \cup I_2)$, which is the synchronous composition net of the CPN_1 and CPN_2.

According to the proposed modular models characteristics, we find that there are public places among modules, such as P_3, P_9 and P_{10} between the first module and the second module. According to the above definitions, we get an ETC control system model diagram based on CPN, as shown in Fig. 2. There are 47 places, 44 transitions, 140 directed arcs, 29 color sets and 25 variables defined in this model. The size of the model is not small, and the number of state spaces is large.

4 ANALYSIS METHOD

It can be known through social investigation and consulting authoritative literature that a vehicle can explore the logical flaws of an ETC control system via some special behaviors, and thus avoid paying fees. We focus on whether there are faulty states in an ETC control system modeled by CPN, i.e., vehicle fee evasion. The so-called faulty states are that there are incorrect tokens in some places. The slicing method is suitable for solving this problem. Therefore, we propose a formal analysis method to analyze the system model based on CPN, which is a dynamic slicing method. This section introduces in detail how to successfully obtain a dynamic slicing for CPN.

CPN gives a data value for each token and defines a color set for each place. Compared to an original Petri net, both input arcs and output arcs have extended arc expression functions to describe data operations, and transitions have also extended guard functions and priority functions to further constrain the occurrence conditions of transitions [48]. Therefore, theoretically speaking, the dynamic slicing idea is also applicable to CPN.

To extract a dynamic slicing of CPN, the dynamic slicing criterion must be defined first. We consider how to define the dynamic slicing criterion from the following three perspectives:

1) In a system modeled by CPN, usually some places represent specific conditions. If there is an error in the design of a system, it will be reflected in these places, that is to say, a incorrect token value in a place is the error we want to find. In this paper, we want to check whether there are errors in the design of an ETC control system, so these places with key properties are what we need to pay attention to.

2) In a system modeled by CPN, each place is defined with a color set. In some cases, multiple places may be defined with a same color set. To facilitate the described model, we define multiple variables to bind the tokens stored in different places, therefore maybe only some variables are of interest to us.

3) The running result of a model changes with an initial marking. The initial marking will affect the places we are interested in. Therefore, we also use it as a slicing criterion.

The specific definition of the dynamic slicing criterion is as follows.

Definition 4.1. For a $CPN=(P, T, A, \Sigma, V, C, G, E, I)$, the dynamic slicing criterion for CPN is a three-tuple $\langle M_0, Q, W\rangle$, where

1) M_0 is an initial marking of CPN;

2) Q is a set of places that we are interested in, and $Q \subseteq P$;

3) W is a finite set of typed variables such that $Type[w] \in C(p_i)_{MS}$ for all variables $w \in W$ and all places $p_i \in Q$.

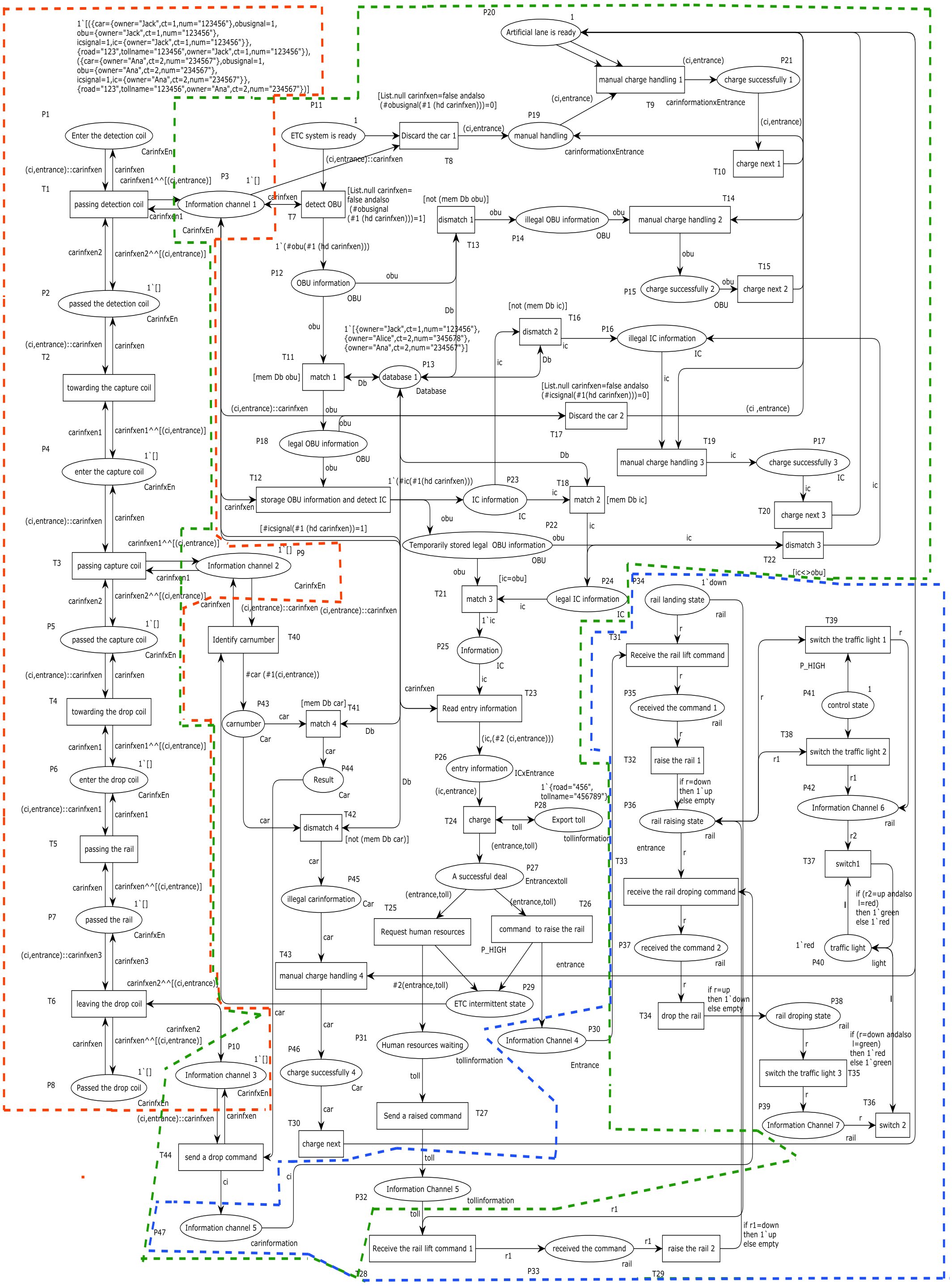


Fig. 2. An ETC control system in highways model based on CPN

Generally speaking, elements which can pass tokens to the interesting places include not only the transitions and places related to them, but also the variables in the arc expressions that related to them. In fact, the dynamic slicing criterion is the formal definition of faulty states. In different application scenarios, the dynamic slicing criterion can depict various faulty states. In summary, compared with the dynamic slicing criterion of a Petri net, we further extend the dynamic slicing criterion to the color set variables in the arc expression functions.

Next, we start to extract the dynamic slicing of a model. The core idea is to capture all token flow paths related to place set Q. We start our work from the following two aspects:

1) Calculate the most possible path along which the tokens flow to the place set Q, and capture the most possible token flow path related to Q and variable set W;

2) Calculate the most possible flow path of the token under the initial marking.

The dynamic slicing of CPN is defined as follows.

Definition 4.2. For $CPN{=}(P, T, A, \Sigma, V, C, G, E, I)$, let $\langle M_0,\ Q,\ W\rangle$ be the dynamic slicing criterion. Given $CPN'{=}(P', T', A', \Sigma', V', C', G', E', I')$, we say that CPN' is a dynamic slice of CPN for $\langle M_0, Q, W\rangle$, if the following conditions hold:

1) The CPN' is a subnet of CPN;

2) For the initial marking M_0 of CPN, if there is a firing sequence of binding elements making that $M_0[(t_1, b_1) > M_1[(t_2, b_2) > \ldots M_{n-1}[(t_n, b_n) > M_n$, and $\exists\ p_i \in Q,\ t_n \in \cdot\ p_i$; then in CPN', $M'_0 = M_0|_{P'}$, there is also a firing sequence of binding elements $\sigma'{=}(t_1, b_1) \ldots (t_m, b_m)(m \leq n)$ making that $M'_0[(t_1, b_1) > M'_1[(t_2, b_2) > \ldots M'_{m-1}[(t_m, b_m) > M'_m$, and $\exists p_i \in Q,\ t_m \in \cdot p_i$; the transition set $(t_1 t_2 \cdots t_m)$ of σ' is a subset of $(t_1 t_2 \cdots t_n)$ in σ, and the binding set $(b_1 b_2 \cdots b_m)$ of σ' is a subset of $(b_1 b_2 \cdots b_n)$ in σ;

3) The sets of places and transitions that cannot contribute tokens to Q do not exist in CPN';

4) For the initial marking M_0 of CPN, if there does not exist any firing sequence $\sigma{=}(t_1, b_1)(t_2, b_2) \ldots (t_n, b_n)$ making that $M_0[(t_1, b_1) > M_1[(t_2, b_2) > \ldots M_{n-1}[(t_n, b_n) > M_n$, and for $\forall\ p_i \in Q,\ t_n \notin \cdot\ p_i$, then $CPN' = \emptyset$.

In Definition 4.2, the firing sequence is the transition queue in which a transition can fire one after another, which is a widely used term in Petri nets. The above condition means that if there is a firing sequence $\sigma = (t_1, b_1)(t_2, b_2) \ldots (t_n, b_n)$ which can contribute tokens to the dynamic criterion, then t_n must belong to the pre-set of the place set Q. To successfully extract the dynamic slice of the CPN, there are three steps:

Firstly, the initial marking is not considered, only the places set Q and variables W are specified. We call this step static slicing. The static slicing criterion for static slicing is defined as follows.

Definition 4.3. For a $CPN = (P, T, A, \Sigma, V, C, G, E, I)$, the static slicing criterion for CPN is a two-tuple $\langle Q, W\rangle$, where

1) Q is a set of places that we are interested in, and $Q \subseteq P$;

2) W is a finite set of typed variables that we are interested in, such that $Type[w] \in C(p_i)_{MS}$ for all variables $w \in W$ and all places $p_i \in Q$.

The Q and W in $\langle Q, W\rangle$ are the same as Q and W in the dynamic slicing criterion $\langle M, Q, W\rangle$. The static slicing of CPN is defined as follows.

Definition 4.4. For $CPN = (P, T, A, \Sigma, V, C, G, E, I)$, let $\langle Q, W\rangle$ be the static slicing criterion. Given $CPN_1{=}(P_1, T_1, A_1, \Sigma_1, V_1, C_1, G_1, E_1, I_1)$, we say that CPN_1 is a static slice of CPN with respect to $\langle Q, W\rangle$, if the following conditions hold:

1) For an arbitrary element e $\in P_1 \cup T_1$ satisfying e $\notin Q$, there is at least one directed arc from e to p for $p \in Q$.

2) For an element e $\in P \cup T$ in CPN, if there is not directed arc from e to p, then e and the arcs

associated with it do not exist in CPN_1.

3) For every $p \in Q$ and $\forall\ t \in T$ in CPN, if there is one directed arc from t to p and isn't any variable $v_i \in W$ in $E(t, p)$, then t and the arcs associated with it do not exist in CPN_1.

The static slicing of CPN captures the token flow path related to place set Q and variable W. The specific steps are shown in Algorithm 1.

Algorithm 1: Static slicing algorithm

```
Input: $CPN = (P, T, A, \Sigma, V, C, G, E, I), \langle Q, W \rangle$
Output: $CPN_1 = (P_1, T_1, A_1, \Sigma_1, V_1, C_1, G_1, E_1, I_1)$
Let a place set $S = \emptyset$, for each $p \in S$, $C(p) \notin \{C(p_i)|p_i \in Q\}, S \subseteq P_1, P_1 = Q, T_1 = \emptyset, A_1 = \emptyset, \Sigma_1 = \{C(p_i)|p_i \in Q\}, V_1 = \emptyset,$
    $C_1 = C(p_i), G_1 = \emptyset, E_1 = \emptyset, I_1 = \emptyset$.
while $(Q \neq \emptyset) or (S \neq \emptyset)$ do
    for each $p_i \in Q$ do
        for each $t_i \in \cdot p_i$ do
            if there is at least one variable $v_i \in W$ in $E(t_i, p_i)$ then
                if $t_i$ doesn't exist in $CPN_1$ then
                    $T_1 = T_1 \cup \{t_i\}, A_1 = A_1 \cup \{(t_i, p_i)\}, E_1 = E_1 \cup \{E(t_i, p_i)\}, G_1 = G_1 \cup \{G(t_i)\}$
                else
                    $A_1 = A_1 \cup \{(t_i, p_i)\}, E_1 = E_1 \cup \{E(t_i, p_i)\}$
                for each $v_i \in E(t_i, p_i)$ do
                    $V_1 = V_1 \cup \{v_i\}$
            for each $p_i \in \cdot t_i$ do
                if $p_i$ exists in the $CPN_1$ then
                    $A_1 = A_1 \cup \{(p_i, t_i)\}, E_1 = E_1 \cup \{E(p_i, t_i)\}, V_1 = V_1 \cup \{v_i|v_i \in ||E(p_i, t_i)||\}$
                else if $p_i$ doesn't exist in the $CPN_1$ then
                    $P_1 = P_1 \cup \{p_i\}, A_1 = A_1 \cup \{(p_i, t_i)\}, E_1 = E_1 \cup \{E(p_i, t_i)\}, V_1 = V_1 \cup \{v_i|v_i \in ||E(p_i, t_i)||\},$
                      $\Sigma_1 = \Sigma_1 \cup \{C(p_i)\}, C_1 = C_1 \cup \{C(p_i)\}, I_1 = I_1 \cup \{I(p_i)\}$
                if $C(p_i) \in \{C(p_i)|p_i \in Q\}$ then
                    $Q = Q \cup \{p_i\}$
                else
                    $S = S \cup \{p_i\}$
        $Q = Q - \{p_i\}$;
    for each $p_i \in S$ do
        for each $t_i \in \cdot p_i$ do
            if $t_i$ doesn't exist in $CPN_1$ then
                $T_1 = T_1 \cup \{t_i\}, A_1 = A_1 \cup \{(t_i, p_i)\}, V_1 = V_1 \cup \{v_i|v_i \in ||E(t_i, p_i)||\}, E_1 = E_1 \cup \{E(t_i, p_i)\}, G_1 = G_1 \cup \{G(t_i)\}$
            else
                $A_1 = A_1 \cup \{(t_i, p_i)\}, V_1 = V_1 \cup \{v_i|v_i \in ||E(t_i, p_i)||\}, E_1 = E_1 \cup \{E(t_i, p_i)\}$
            for each $p_i \in \cdot t_i$ do
                if $p_i$ exists in the $CPN_1$ then
                    $A_1 = A_1 \cup \{(p_i, t_i)\}, E_1 = E_1 \cup \{E(p_i, t_i)\}, V_1 = V_1 \cup \{v_i|v_i \in ||E(p_i, t_i)||\}$
                else if $p_i$ doesn't exist in the $CPN_1$ then
                    $P_1 = P_1 \cup \{p_i\}, A_1 = A_1 \cup \{(p_i, t_i)\}, E_1 = E_1 \cup \{E(p_i, t_i)\}, V_1 = V_1 \cup \{v_i|v_i \in ||E(p_i, t_i)||\},$
                      $\Sigma_1 = \Sigma_1 \cup \{C(p_i)\}, C_1 = C_1 \cup \{C(p_i)\}, I_1 = I_1 \cup \{I(p_i)\}$
                if $C(p_i) \in \{C(p_i)|p_i \in Q\}$ then
                    $Q = Q \cup \{p_i\}$
                else
                    $S = S \cup \{p_i\}$
            $S = S - \{p_i\}$
```

Algorithm 1 describes how to extract a static slice of CPN. We consider it from two perspectives:

1) The token value of a place p depends on its input transitions and the variables in its input arc expression;

2) The tokens passed by the transitions come from its input place.

The purpose of Algorithm 1 is to obtain a static slice CPN_1=$(P_1, T_1, A_1, \Sigma_1, V_1, C_1, G_1, E_1, I_1)$, initially P_1= Q, Σ_1=$\{C(p_i)|p_i \in Q\}$. We set a place set S such that the color set data type of each place in this set S cannot be found in the color set data type of all places in the set Q. While extracting $p \in CPN_1$ from CPN, Algorithm 1 automatically divides the places into 2 categories, if $C(p) \in C(p_i)(p_i \in Q)$, put the place p into the place set Q, otherwise, put it into a place set S, which correspond to lines 17 to 20 and lines 33 to 36 of the algorithm. Initialize S to be an empty set.

The execution of Algorithm 1 can be summarized in two big steps:

1) Lines 3 to 21 in Algorithm 1: for the variables defined by the color sets of the places in the set Q, not all variables are of interest to us, therefore, while looking for the transitions that the tokens in the places rely on, we need to consider whether the variables on the input arc expressions are of interest to us. If not, the transition is considered irrelevant. Otherwise, lines 5 to 11 of the algorithm are executed to collect the transitions and related elements into the CPN_1.

Lines 12 to 20 of Algorithm 1: by judging whether the variables in the input arc expressions of the places are of interest to us, we find all the places that can pass the token to p_i through $t_i \in \cdot p_i$. If these places have already exited in CPN_1, Algorithm 1 executes line 14; if these places do not exist in CPN_1, Algorithm 1 executes line 16.

When Q is empty, this step ends and the execution proceeds to line 22.

2) Lines 22-37 of Algorithm 1: for every place in the place set S, there is no need to consider whether the variables in the input arc expression are of interest to us while looking for relevant transitions. If these transitions have already exited in CPN_1, Algorithm 1 executes line 25; if these transitions do not exist in CPN_1, Algorithm 1 executes line 27.

Going back to the places where these transitions are connected, if these places have already exited in CPN_1, Algorithm 1 executes line 30; if these places do not exist in CPN_1, Algorithm 1 executes line 32.

In Algorithm 1, backtrack the places in the sets Q and S, and remove the places that have been backtracked from the set Q or S, until the set Q and the set S are empty. In Algorithm 1, line 2 refers to the termination condition of Algorithm 1 is that both Q and S are empty.

Secondly, consider the initial marking M_0, and calculate the most possible flow path of the tokens in the initial marking M_0 without running CPN. We call this process forward slicing, and the steps are described in Algorithm 2.

Algorithm 2 describes how to obtain a forward slice according to the dynamic slicing criterion. We consider it from three aspects:

1) Under a marking M, the enablement and occurrence of a binding element $\langle t, b\rangle$ need to satisfy $G(t)\langle b\rangle$ and $\forall p \in P$: $E(p,t)\langle b\rangle <<= M(p)$;

2) Under a marking M, a transition with higher priority occurs firstly;

3) For $\forall p \in P$: $C(p) \in C(p_i)(p_i \in Q)$, while capturing the flow path of tokens in these places, we only capture the tokens bound to the variables we are interested in on the output arc expression.

The purpose of Algorithm 2 is to obtain $CPN_2 = (P_2, T_2, A_2, \Sigma_2, V_2, C_2, G_2, E_2, I_2)$ according to the dynamic slicing criterion $\langle M_0, Q, W\rangle$. Considering all transitions that can be enabled and occurred under the initial marking M_0, only these transitions are bound to tokens, so that these tokens have opportunities to be passed to the place net Q.

For every transition, we consider two cases:

1) If it only has an input place, and the color set of the input place is inconsistent with the color sets of places in the place set Q, its input place, input arc and other related elements are of interest to us, the specific steps are described in lines 7-16 of Algorithm 2;

2) If the number of input places is more than one, then only if the color sets of all input places are consistent with the color sets of the place set Q, and we are not interested in the variables on

Algorithm 2: Forward slicing algorithm

```
   Input: $CPN = (P, T, A, \Sigma, V, C, G, E, I), \langle M_0, Q, W \rangle$
   Output: $CPN_2 = (P_2, T_2, A_2, \Sigma_2, V_2, C_2, G_2, E_2, I_2)$
1  $P_2 = \{p_i | M_0(p_i) \neq \emptyset\}, T_2 = \emptyset, A_2 = \emptyset, \Sigma_2 = \{C(p_i) | p_i \in P_2\}, V_2 = \emptyset, C_2 = C(p_i), G_2 = \emptyset, E_2 = \emptyset, I_2 = I(p_i)(p_i \in P_2)$.
2  Compute all enabled binding elements $\langle t_i, b_j \rangle$ in initial marking $M_0$, then get all enabled transitions $t_i$ in initial marking $M_0$;
3  while $\{t_i | i = 1, 2, ..., n\} \neq \emptyset$ do
4      if $t_i$ doesn't exist in $CPN_2$ then
5          for each $t_i$ do
6              $p_i = \cdot t_i$
7              if $|p_i| = 1$ then
8                  if $C(p_i) \in \{C(q_i) | q_i \in Q\}$ then
9                      if there is at least one variable $v_i \in W$ in $E(p_i, t_i)$ then
10                         $T_2 = T_2 \cup \{t_i\}, E_2 = E_2 \cup \{E(p_i, t_i)\}, A_2 = A_2 \cup \{(p_i, t_i)\}, G_2 = G_2 \cup \{G(t_i)\}$
11                         for every $v_i \in E(p_i, t_i)$ do
12                             $V_1 = V_1 \cup \{v_i\}$
13                 else
14                     $T_2 = T_2 \cup \{t_i\}, A_2 = A_2 \cup \{(p_i, t_i)\}, E_2 = E_2 \cup \{E(p_i, t_i)\}, V_1 = V_1 \cup \{v_i | v_i \in ||E(p_i, t_i)||\}$,
                       $G_2 = G_2 \cup \{G(t_i)\}$
15                 for each $p_i = t_i \cdot$ do
16                     $P_2 = P_2 \cup \{p_i\}, A_2 = A_2 \cup \{(t_i, p_i)\}, E_2 = E_2 \cup \{E(t_i, p_i)\}, C_2 = C_2 \cup \{C(p_i)\}, \Sigma_2 = \Sigma_2 \cup \{C(p_i)\}$
17             else if $|p_i| \geq 2$ then
18                 if it is not all $C(p_i) \in \{C(q_i) | q_i \in Q\}$ then
19                     for each $p_i$ do
20                         $P_2 = P_2 \cup \{p_i\}, A_2 = A_2 \cup \{(p_i, t_i)\}, E_2 = E_2 \cup \{E(p_i, t_i)\}, C_2 = C_2 \cup \{C(p_i)\}, V_2 = V_2 \cup \{v_i\}$,
                           $\Sigma_2 = \Sigma_2 \cup \{C(p_i)\}$
21                     $T_2 = T_2 \cup \{t_i\}, G_2 = G_2 \cup \{G(t_i)\}$
22                 else if there is at least one variable $v_i \in W$ in $E(p_i, t_i)$ then
23                     for each $p_i$ do
24                         $P_2 = P_2 \cup \{p_i\}, V_2 = V_2 \cup \{v_i\}, A_2 = A_2 \cup \{(p_i, t_i)\}, E_2 = E_2 \cup \{E(p_i, t_i)\}, \Sigma_2 = \Sigma_2 \cup \{C(p_i)\}$,
                           $C_2 = C_2 \cup \{C(p_i)\}$
25                     $T_2 = T_2 \cup \{t_i\}, G_2 = G_2 \cup \{G(t_i)\}$
26                 for each $p_i = t_i \cdot$ do
27                     $P_2 = P_2 \cup \{p_i\}, A_2 = A_2 \cup \{(t_i, p_i)\}, E_2 = E_2 \cup \{E(t_i, p_i)\}, C_2 = C_2 \cup \{C(p_i)\}, \Sigma_2 = \Sigma_2 \cup \{C(p_i)\}$
28     Get the new marking $M$, recalculate enabled binding elements $\langle t_i, b_j \rangle$ under the marking
```

the input arc expressions, the transition, and relevant elements such as input places and input arcs will be not the parts we are interested in, otherwise, these elements will be also included in the forward slice. Details are seen in lines 21-34 of Algorithm 2. For the transitions we are interested in, related elements such as the output places and output arcs are also included in the forward slice.

Then calculate the reachable state M when the transition occurs, for the transitions that can be enabled and occurred in the reachable state M, we continue the above operations, until for any reachable marking M of the initial state M_0, the transitions that can be enabled and occurred and their related elements are all included in the forward dynamic slice.

Finally, we obtain a dynamic slicing by taking the common part of the forward slice and the static slice. This step is described in Algorithm 3.

Algorithm 3: Dynamic slicing algorithm

```
   Input: $CPN_1 = (P_1, T_1, A_1, \Sigma_1, V_1, C_1, G_1, E_1, I_1), CPN_2 = (P_2, T_2, A_2, \Sigma_2, V_2, C_2, G_2, E_2, I_2)$
   Output: $CPN' = (P', T', A', \Sigma', V', C', G', E', I')$
1  $P' = P_1 \cap P_2, T' = T_1 \cap T_2, A' = A_1 \cap A_2, \Sigma' = \Sigma_1 \cap \Sigma_2, V' = V_1 \cap V_2, C' = C(p'), G' = G(t'), E' = E_1 \cap E_2, I' = I(p')$.
```

The dynamic slicing of CPN obtained according to Algorithm 3 has a much smaller scale than the original CPN, and the number of reachable states of the corresponding model is also much less, so state space explosion can be avoided.

We can show that Algorithms 1, 2 and 3 are all terminable and sound. And the proof process is given by taking Algorithm 1 as an example. The other two are similar.

Theorem 4.5. *Algorithm 1 can be terminated.*

Proof. As the number of places and transitions of a CPN is limited, so the number of $p_i \in Q$ and $t_i \in \cdot p_i$ in lines 3 and 4 is limited. In the same way, the number of $p_i \in S$ and $t_i \in \cdot p_i$ in lines 22 and 23 is also limited. The number of elements in a CPN is finite, including not only the number of places and transitions, but also the number of arc expressions, color sets, variables, types of color sets, and the length of guard functions and initial marking. Therefore, the execution of lines 5 to 11 and 13 to 20 must take a finite amount of time. The number of $v_i \in ||E(t_i, p_i)||(||E(t_i, p_i)||$ denotes the set of variables in this arc expression $||E(t_i, p_i)||$) and $p_i \in \cdot t_i$ in lines 10 and 12 is limited. In the same way, the execution of lines 24 to 27 and 29 to 36 must take a finite amount of time. The number of $p_i \in \cdot t_i$ in lines 28 is limited. □

Theorem 4.6. *For CPN $= (P, T, A, \Sigma, V, C, G, E, I)$, let $\langle M_0, Q, W\rangle$ be the dynamic slicing criterion. CPN_1 is the static slicing of CPN acquired through Algorithm 1, and CPN_2 is the forward slicing of CPN acquired through Algorithm 2. According to Algorithm 3, we get $CPN' = (P', T', A', \Sigma', V', C', G', E', I')$. Therefore, CPN' must be the dynamic slicing of CPN.*

Proof. Firstly, in Algorithm 1, for each $p_i \in Q$, the target token value is passed through the transitions adjacent to the places in Q. However, in fact, there may be more than one transition adjacent to each $p_i \in Q$. As for each $t_i \in \cdot p_i$, if there is at least one variable $v_i \in W$ in $E(t_i, p_i)$, collect the transitions and related elements into the CPN_1. This means that, in CPN_1, the transitions adjacent to each $p_i \in Q$ has one directed arc to p_i for $p_i \in Q$. Then find all places $p_i \in \cdot t_i$ that can transmit the token to p_i ($p_i \in Q$) through $t_i \in \cdot p_i$ ($p_i \in Q$). This means that each $p_i \in \cdot t_i$ has directed arcs from p_i to p for $p_i \in Q$. For each place p_i in the place set S, collect each transition $t_i \in \cdot p_i$ and related elements into the CPN_1. This means that, in CPN_1, the transitions adjacent to each $p_i \in S$ has one directed arc to p_i for $p_i \in S$. For each $p_i \in \cdot t_i$, collect the places and related elements into the CPN_1. This means that each $p_i \in \cdot t_i$ has directed arcs from p_i to p for $p \in S$. According to lines 17-20 of Algorithm 1, there is directed arcs from $p \in S$ to $p_i \in Q$. For every $p \in Q$ and $\forall t \in T$ in CPN, if there is one directed arc from t to p, there is at least one variable $v_i \in W$ in $E(t_i, p_i)$. In general, Algorithm 1 can satisfy *Definition 4.4*, and it is correct.

Algorithm 2 gets all enabled transitions t_i in initial marking M_0. For the initial marking M_0 of CPN, if there is a firing sequence of binding elements making that $M_0[(t_1, b_1) > M_1[(t_2, b_2) > \dots M_{n-1}[(t_n, b_n) > M_n$, and $\exists\, p_i \in Q$, $t_n \in\ p_i$. According to the traveling process of Algorithm 2, only the binding elements $< t_i, b_j >$ can be enabled in CPN would be included in CPN_2. There is a firing sequence of binding elements $\sigma' = (t_1, b_1)(t_2, b_2)\dots(t_m, b_m)$ (m ≤ n) making that $M_0[(t_1, b_1) > M_1[(t_2, b_2) > \dots M_{m-1}[(t_m, b_m) > M_m$, and $\exists\, p_i \in Q$, $t_m \in\ p_i$. The transition set $(t_1, t_2, \dots, t_m)$ of σ' is a subset of $(t_1, t_2, \dots, t_n)$ in σ. It calculates the most possible flow path of the tokens in the initial marking M_0 without running CPN. Algorithm 2 is correct. The input to Algorithm 3 is the intersection of two slices of the CPN obtained by Algorithms 1 and 2. This indicates that CPN' must be a dynamic slice of the CPN. □

Regarding the time complexity of the proposed dynamic slicing method, Algorithm 1 is a backward tracking algorithm, and Algorithm 2 is a forward traversal algorithm, their cost are both bounded by the number of elements in CPN, thus, the cost of our algorithms is $O(2(|P| + |T| + |A| + |\Sigma| + |V|))$, $|P|$ is the number of places, $|T|$ is the number of transitions, $|A|$ is the number of arcs, $|\Sigma|$ the number of color sets and $|V|$ is the number of variables in CPN. It can be seen that a dynamic slicing method can be concerned lightweight. In the worst case, a dynamic slice is as large as the original model.

5 A CASE STUDY AND DISCUSSION

To detect whether an ETC control system design has flaws, the modeling method and formal analysis method have been given in Sections III and IV. There are three stages for detecting whether an ETC control system design has a certain flaw:

1) Specification of an ETC control system with CPN.
2) A dynamic slice of the CPN model.
3) Full state spaces of a dynamic slice.

We illustrate these with a case.

5.1 A case study

We use the dynamic slicing method to analyze an ETC control system model established in Section III to detect whether there is a flaw mentioned in Section II.

Firstly, before extracting a dynamic slicing of the model, we should determine a dynamic slicing criterion. In this model, we detect whether such a state f exists: when the first vehicle has passed the railing (P_7) and the state of the railing is up ($M(P_{36})$ =1'up), the following vehicle also passes the railing (P_7). The existence of this state f means that there is a flaw in the system that vehicles can pass through the railing without paying fees. Therefore, the places we are interested in are P_7 and P_{36}, initially $Q = \{P_7, P_{36}\}$. The color sets and corresponding variables of these two places are shown in Table 2.

Table 2. Color sets and variables of the places of interest.

Places	Color sets	Variables
P_7	CarinfxEn	carinfxen, carinfxen1, carinfxen2, carinfxen3
P_{36}	rail	r, r1, r2

Each element of the system model only interacts with its immediate neighbors, so it can be known that among the input transitions of the place P_7, it is impossible for transition T_6 to pass the incorrect tokens to the place P_7. So it is also impossible for the variables on the input arc expression from transition T_6 to place P_7 to bind the incorrenct token. Thus the variable "carinfxen3" on the input arc expression from transition T_6 to place P_7 is not interesting to us. By parity of reasoning, the variables "r1" and "r2" are also not interesting to us.

For P_7, we determine carinfxen, carinfxen1, carinfxen2 as the variables we are interested in, and for P_{36}, we determine r as the variable of interest. Therefore, W = {carinfxen, carinfxen1, carinfxen2, r}.

The initial marking of the model is $M_0(P_1)$ = 1' [({car = {owner = " Jack ", ct = 1, num = " 123456 "}, obusignal = 1, obu = {owner = " Jack ", ct = 1, num = " 123456 "}, icsignal = 1, ic = {owner = " Jack", ct = 1, num = " 123456 "}}, {road = " 123 ", tollname = " Tongguan toll station", owner = " Jack ", ct = 1, num = " 123456 "}), ({car = {owner =" Ana ", ct = 2, num = " 234567 "}, obusignal = 1, obu = {owner = " Ana", ct = 2, num = " 234567 "}, icsignal = 1, ic = {owner = " Ana", ct = 2, num = " 234567 "}}, {road = " 123 ", tollname = " Tongguan toll station ", owner = " Ana ", ct = 2, num = " 234567 "})], $M_0(P_{11})$ = 1' ( ), $M_0(P_{13})$ = 1' [{owner = " Jack ", ct = 1, num = " 123456 "}, {owner = " Alice ", ct = 2, num = " 345678 "}, {owner = " Ana ", ct = 2, num = " 234567 "}], $M_0(P_{20})$ = 1' ( ), $M_0(P_{28})$ = 1' {road = " 456 ", tollname = " 456789 "}, $M_0(P_{34})$ = 1' down, $M_0(P_{40})$ = 1' red, $M_0(P_{41})$ = 1' ( ).

Then according to Definition 4.3 and Algorithm 1, we extract the static slicing of the model, as shown in Fig. 3; according to Definition 4.1 and Algorithm 2, we extract the forward slicing of the model, as shown in Fig. 4; according to Algorithm 3, we get the dynamic slicing of the model, as shown in Fig. 5. It can be known from Fig. 5 that the dynamic slice is smaller than the model in Fig.2. Finally, we analyze the state space of the dynamic slice in Fig. 5 to detect whether there is a faulty state f. We find that the model has such a fault state f,

$M(P_{36})$ =1' up

$M(P_7)$= 1' [({car = {owner = " Jack ", ct = 1, num = " 123456 "}, obusignal = 1, obu = {owner = " Jack ", ct = 1, num = " 123456 "}, icsignal = 1, ic = {owner = " Jack", ct = 1, num = " 123456 "}}, {road = " 123 ", tollname = " Tongguan toll station " , owner = " Jack ", ct = 1, num = " 123456 "}), ({car = {owner = " Ana ", ct = 2, num = " 234567 "}, obusignal = 1, obu = {owner = " Ana", ct = 2, num = " 234567 "}, icsignal = 1, ic = {owner = " Ana ", ct = 2, num = " 234567 "}}, {road = " 123 ", tollname = " Tongguan toll station " , owner = " Ana ", ct = 2, num = " 234567 "})]

indicating that the control process of the ETC control system has a flaw that a vehicle that has not completed the payment passing the railing by following the previous vehicle, and needs to be further modified.

Figs. 6 and 7 are the partial state space reports of the original model and the dynamic slicing respectively. Fig. 6 shows that the size of full state spaces of the original model is 663. Fig. 7 shows that the size of full state spaces of the dynamic slice of the model is 501. Obviously, for a dynamic slicing, the size of the state space is much less than the original model, so the efficiency to analyze dynamic slicing is much higher than analyzing the original model.

5.2 Result from the case study

For the control logic process of an ETC control system, we propose a method to eliminate the state that a vehicle that has not completed the payment passing the railing by following the previous vehicle, so that each vehicle only passes the railing when the payment is completed.

The existence of the faulty state means that an ETC control system cannot guarantee that every vehicle passing the railing has a corresponding transaction record. In other words, if an ETC control system can check whether each vehicle that will pass the railing has a corresponding transaction record, then the above phenomenon may be eliminated. It can be known from the control process of an ETC control system in Fig. 5 that after a vehicle completes the transaction, the computer system temporarily stores the transaction record of the vehicle, immediately after the vehicle enters the capture coil, a license plate recognition system is triggered, and the computer temporarily stores the license plate information of the vehicle. Therefore, we try to add a business to the system, that is, the computer system checks whether the transaction record and the license plate information are consistent, if they are consistent, the vehicle is allowed to pass through the railing, otherwise, an alarm sounds. A modified ETC control system model is shown in Fig. 8.

Analyze the modified system model and find that there is such a state in the model: when the first vehicle has passed the railing (P_7) and the state of railing is up($M(P_{36})$=1' up), the following vehicle that doesn't complete the payment cannot pass the railing(P_7). The existence of this state means that there isn't a flaw in the system that vehicles can pass the railing without paying fees by following the previous vehicle.

For a modified ETC control system, a vehicle requires 5 information checks, including check the legality of the information stored in OBU, check the legality of the information stored in the IC card, check whether the information stored in OBU and the IC card are consistent, check whether the license plate information is legal, and check whether the transaction records and license plate information are consistent. Only these 5 information checks are correct, can the vehicle successfully

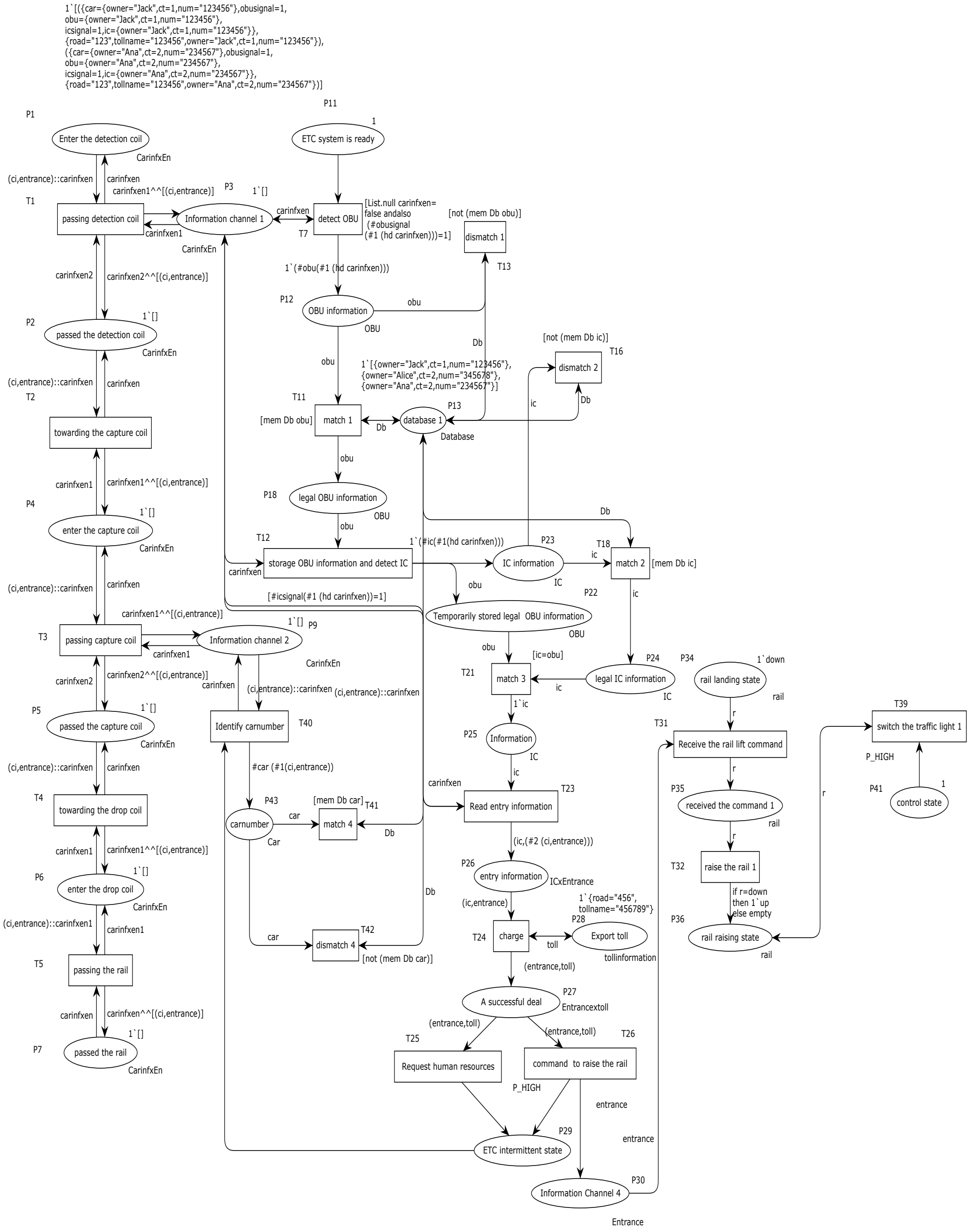


Fig. 3. The static slicing of the model

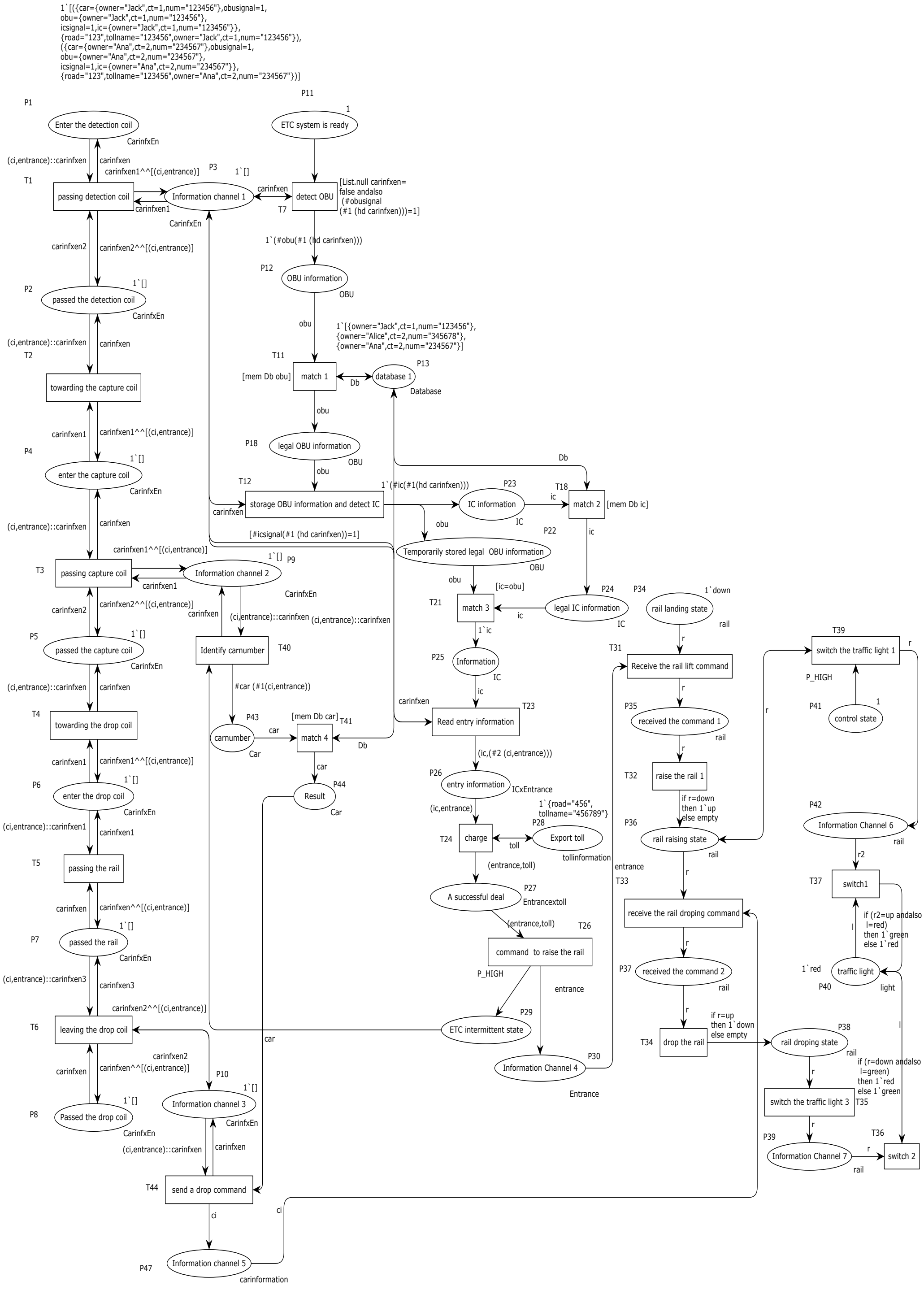


Fig. 4. The forward slicing of the model

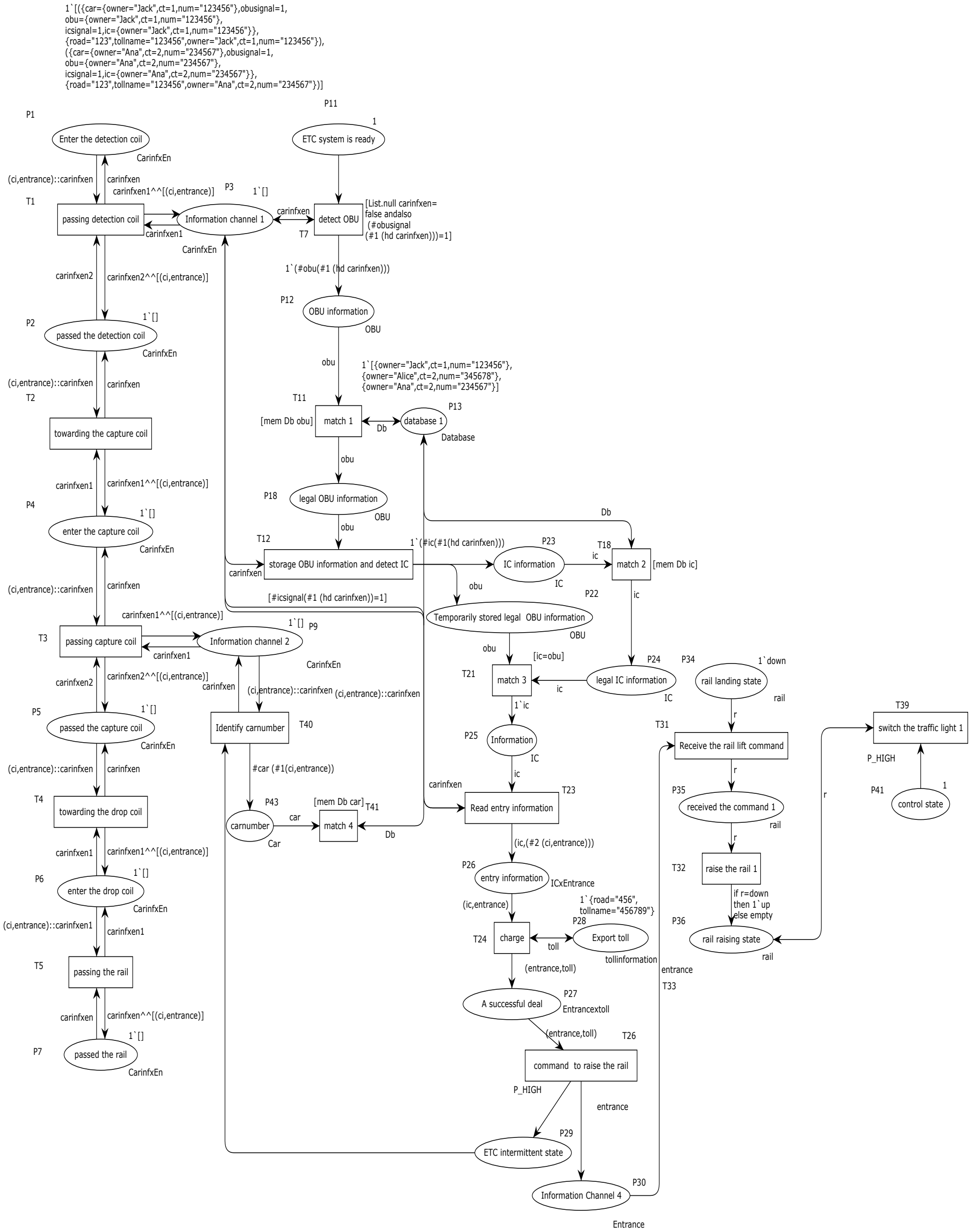


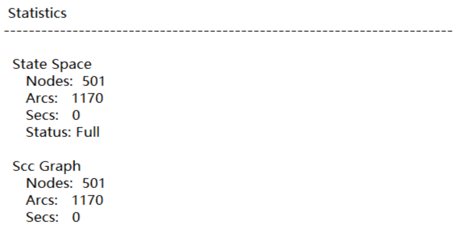
Fig. 5. The dynamic slicing of the model

```
Statistics
------------------------------------------------------------------------

 State Space
    Nodes:  663
    Arcs:   1604
    Secs:   0
    Status: Full

 Scc Graph
    Nodes:  663
    Arcs:   1604
    Secs:   0
```

Fig. 6. The amount of full state space in the original model



```
Statistics
------------------------------------------------------------------------

 State Space
    Nodes:  501
    Arcs:   1170
    Secs:   0
    Status: Full

 Scc Graph
    Nodes:  501
    Arcs:   1170
    Secs:   0
```

Fig. 7. The amount of full state space in the dynamic slicing

pass the railing. In the modified model, the names of some places that are used to store the above information and their color set definitions are shown in Table 3.

Table 3. Color sets of some places.

Places	Color sets	Definition
OBU information(P_{12})	OBU	= record owner: ownername * ct: cartype * num: carnumber;
database 1(P_{13})	Database	= list database; (colset database = record owner: ownername * ct: cartype * num: carnumber)
legal IC information(P_{18})	IC	= record owner: ownername * ct: cartype * num: carnumber;
carnumber (P_{43})	Car	= record owner: ownername * ct: cartype * num: carnumber;
A successful deal(P_{27})		
transaction record(P_{48})	Entrancextoll	= product Entrance * tollinformation;

5.3 Discussion

This paper provides a formal method to model and analyze an ETC control system in highways. The case study shows that CPN and dynamic slicing method can detect whether there is a flaw, i.e., a vehicle passes the railing by following the previous vehicle in an ETC control system. This formal method can be considered as a reliable and universal method. For other logic issues, it can also detect whether there is a flaw in the control logic process of an ETC control system. Moreover, it has unique advantages compared with other methods.

Previous studies [10–15] propose some technologies to improve one section of an ETC control system. By data mining and big data analysis on the regular patterns of data information of prior vehicles, Refs. [16–24] establish recognition models for vehicle toll evasion behaviors, and then predict what happens to the target vehicles based on the proposed models. It is also useful to predict vehicles with fee evasion. However, its application must be based on the existing prior data. It is

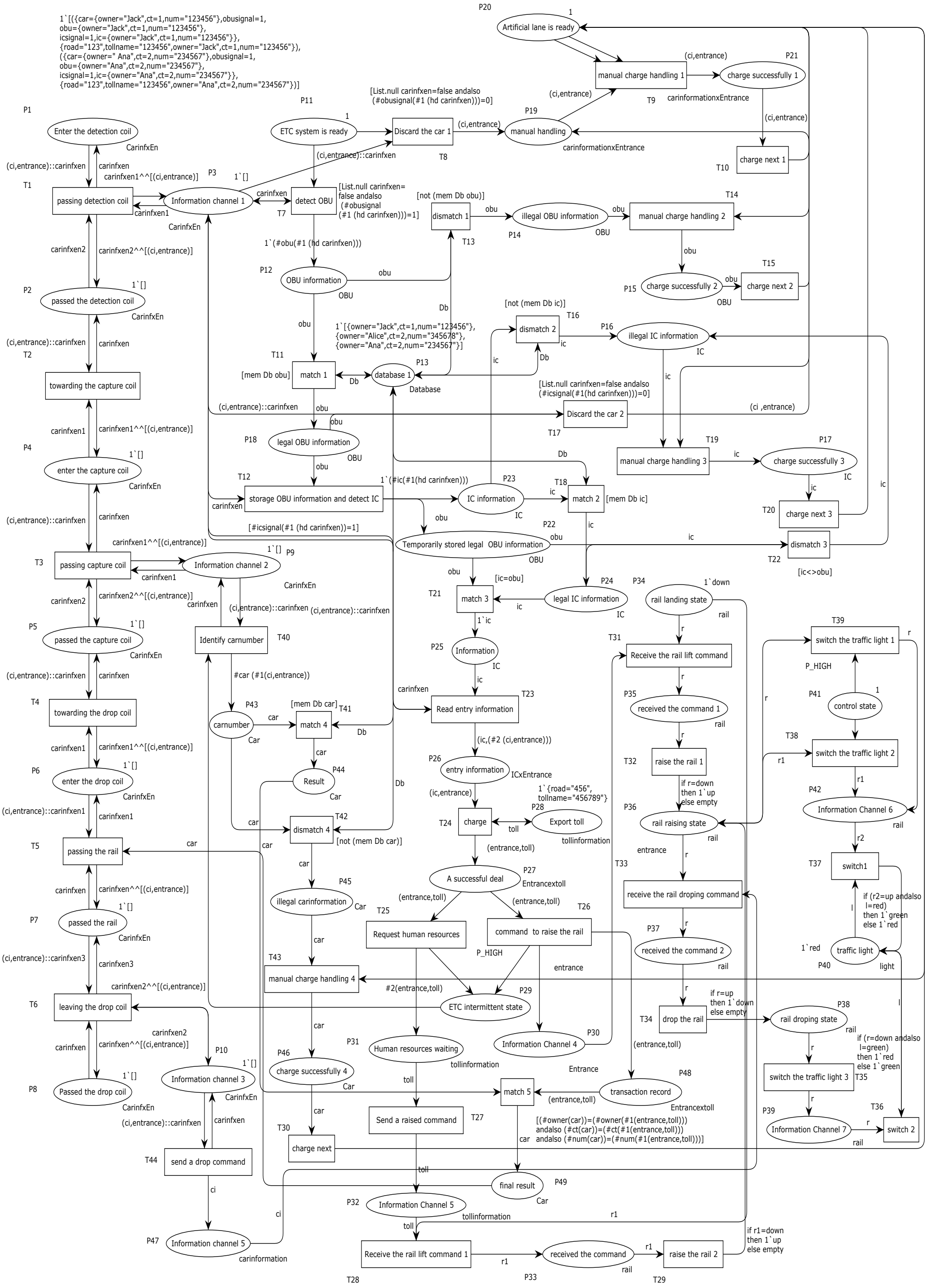


Fig. 8. A modified ETC control system model

helpless to detect the logical flaws and repair the business system at the design phase from the business process perspectively.

An ETC control system is a large-scale complex business interaction system. Both improving partial technologies and big data analysis do not concern its whole control process and cannot get rid of its control logic flaws.

The proposed formal modeling and analysis method compensates for the shortcomings of the previous methods. As we have analyzed before, there are flaws in the control logic process of an ETC control system. For the formal method, it can detect flaws in its control logic process and solve them by modifying its business process. The advantage of the formal method is self-evident.

In summary, both previous and our proposed method have their own advantages and disadvantages respectively. They can complement each other.

6 CONCLUSION

This paper focuses on modeling and analysis methods for an ETC control system at the design phase. In order to discover the potential logical flaws of the system at the design phase, this paper adopt CPN to formally model the control logic process of an ETC system, and propose a formal analysis method named dynamic slicing to analyze whether there is a flaw in the process. As a case study, we solve the problem that a vehicle passes the rail by following the previous vehicle. It shows the effectiveness of our proposed methodology. For the dynamic slicing method of CPN, we have proved that it can automatically identify the part we are interested in and simplify the analyzing of a model. It is helpful for us to analyze the CPN model. However, there are many issues in ETC control systems, and in future, we intend to apply the proposed methodology to other control issues and even other fields [49–53]. Besides, more effective formal methods should be extended and explored. In addition, despite the effectiveness of our proposed approach, CPN tools have limited use for our models, which ensures that there is scope for further developing the related tools or functions for supporting the full automation of the proposed methodology, or combining third party components with CPN tools.

7 ACKNOWLEDGMENTS

This work was supported in part by the Open Research Fund of Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety, China, under Grant CSBD2022-ZD05; in part by the Fundamental Research Funds for the Central Universities under Grant GK202205039; and in part by the Natural Science Foundation of Shaanxi Province under Grant 2021JM-205.

REFERENCES

[1] E. H. Charan, I. Pal, A. Sinha, R. K. R. Baro, and V. Nath. Electronic toll collection system using barcode technology. In *Nanoelectronics, Circuits and Communication Systems*, pages 549–556. Springer, 2019.

[2] M. M. Desai and J. J. Patoliya. Smart toll collection system using embedded linux environment. In *2017 2nd International Conference for Convergence in Technology (I2CT)*, pages 79–83. IEEE, 2017.

[3] S. Ahmed, T. M. Tan, A. M. Mondol, Z. Alam, N. Nawal, and J. Uddin. Automated toll collection system based on rfid sensor. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–3. IEEE, 2019.

[4] Y. Tan and X. Li. Behavior Analysis and Measure Research on Highway ETC Toll Evasion. *Technology & Economy in Areas of Communications*, 16(3):30–33, 2014.

[5] C. Su. The constitution and advantages and disadvantages of ETC electronic toll collection system on expressway. *The World of Transportation*, (18):150–151, 2020.

[6] Q. Dong. Ways and Countermeasures of Vehicle Escaping Fee in Expressway Toll Station. *Modern Economic Information*, (19):350, 2019.

[7] S. Zhao. A brief talk on the behavior of ETC fee evasion on expressway and its countermeasures. *Science and Technology Innovation and Application*, (35):279, 2014.

[8] M. Qiu. The main forms and prevention measures of highway vehicle escaping toll. *Western China Communications Science & Technology*, 3:38–42, 2011.

[9] H. Shao. Analysis of expressway vehicle fee evasion types and countermeasures. *New Economy*, (14):128, 2015.

[10] L. Zhu and L. He. Patent Survey on Electronic Toll Collection (ETC) Technologies. *China Invention & Patent*, 16(S2):95–99, 2019.

[11] H. Huang, S. Xiao, and X. Meng. Application of RFID and SNMP technology in highway electronic toll collection system. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 3, pages 383–385, 2010.

[12] S. Lu, T. He, and Z. Gao. Design of electronic toll collection system based on global positioning system technique. In *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, volume 1, pages 350–353, 2009.

[13] M. Randriamasy, A. Cabani, H. Chafouk, and G. Fremont. Geolocation process to perform the electronic toll collection using the ITS-G5 technology. *IEEE Transactions on Vehicular Technology*, 68(9):8570–8582, 2019.

[14] R. Abdulla, A. Abdillahi, and M. K. Abbas. Electronic toll collection system based on radio frequency identification system. *International Journal of Electrical and Computer Engineering*, 8(3):1602–1610, 2018.

[15] P. Jain, P. Dhillon, A. V. Singh, K. Vats, and S. Tripathi. A unique identity based automated toll collection system using RFID and image processing. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 988–991, 2018.

[16] H. Diao. Escaped toll analysis of ETC system customer data based on NN model of stable GA. *Computer and information technology*, 14(4):16–19, 2006.

[17] S. Xia, H. Qu, and Y. Tang. WEKA in Studying Expressway Toll Evasion by Changing Cards. *Journal of Shanghai Ship and Shipiing Research Institute*, 36(4):71–76, 2013.

[18] L. Chu, X. Guo, G. Song, J. Wu, and K. Xie. Clustering Based Identification and Analysis of Defrauding of Highway Toll. *The 8th China Intelligent Transportation Annual Conference*, pages 59–66, 2013.

[19] Y. Zhao, S. Wu, Z. Lin, and T. Chang. Study on the prediction model of toll fraud behavior for highway pass card. *China Sciencepaper*, 10(19):2245–2251, 2015.

[20] Y. Wang, F. Li, X. Cui, and L. Zhang. Research on Types and Data Characteristics of Freeway Escape in Henan Province. In *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pages 621–625. IEEE, 2020.

[21] S. Wu. *The Study of Anti-defrauding in Network Expressway based on Data Mining.* PhD thesis, South China University of Technology, 2016.

[22] B. Chen. *Research and Application of Data Mining in Highway Prevent evasion.* PhD thesis, South China University of Technology, 2014.

[23] Q. Ma. *Research and Application of Mining Vehicle Abnormal Behabior on Expressway Based on Traffic Big Data.* PhD thesis, North China University of Technology, 2019.

[24] X. Li. *The research and application of Toll Audit in Networked Expressway based on Data Mining.* PhD thesis, South China University of Technology, 2014.

[25] G. Zhang. *Introduction to formal methods.* Tsinghua University Press, 2015.

[26] R. Wisniewski, M. Zhou, L. Gomes, M. P. Fanti, and R. Kumar. Special issue on recent advances in Petri nets, automata, and discrete-event hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(3):3484–3487, 2020.

[27] B. Huang, M. Zhou, C. Wang, A. Abusorrah, and Y. Al-Turki. Deadlock-free supervisor design for robotic manufacturing cells with uncontrollable and unobservable events. *IEEE/CAA Journal of Automatica Sinica*, 8(3):597–605, 2021.

[28] S. Basu and R. Kumar. Control of non-deterministic systems with μ-calculus specifications using quotienting. *IEEE/CAA Journal of Automatica Sinica*, 8(5):953–970, 2021.

[29] W. Yu, C. Yan, Z. Ding, C. Jiang, and M. Zhou. Analyzing E-Commerce business process nets via incidence matrix and reduction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(1):130–141, 2018.

[30] N. Wu and M. Zhou. Colored timed Petri nets for modeling and analysis of cluster tools. *Asian Journal of Control*, 12(3):253–266, 2010.

[31] R. Robidoux, H. Xu, L. Xing, and M. Zhou. Automated modeling of dynamic reliability block diagrams using Colored Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(2):337–351, 2010.

[32] W. Yu, C. Yan, Z. Ding, C. Jiang, and M. Zhou. Modeling and validating e-commerce business process based on petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(3):327–341, 2013.

[33] W. Yu, Z. Ding, L. Liu, X. Wang, and R. Crossley. Petri net-based methods for analyzing structural security in e-commerce business processes. *Future Generation Computer Systems*, 109:611–620, 2020.

[34] L. Chen, T. Tang, X. Zhao, and E. Schnieder. Verification of the safety communication protocol in train control system using colored Petri net. *Reliability Engineering & System Safety*, 100:8–18, 2012.

[35] K. Farah, K. Chabir, and M. N. Abdelkrim. Colored Petri nets for modeling of networked control systems. In *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pages 226–230. IEEE, 2019.

[36] K. Jensen and L. M. Kristensen. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer, 2009.

[37] Y. I. Khan, A. Konios, and N. Guelfi. A Survey of Petri Nets Slicing. *ACM Computing Surveys*, 51(5):109.1–109.32, 2019.

[38] C. Chang and H. Wang. A slicing algorithm of concurrency modeling based on Petri nets. In *1986 International Conference on Parallel Processing, University Park, PA*, pages 789–792, 1986.

[39] W. J. Lee, S. D. Cha, Y. R. Kwon, and H. Kim. A slicing-based approach to enhance Petri net reachability analysis. *Journal of Research and Practice in Information Technology*, 32(2):131–143, 2000.

[40] A. Rakow. Slicing Petri nets. In *Proceedings of the Workshop on FABPWS*, pages 56–70. Citeseer, 2007.

[41] A. Rakow. Slicing Petri nets with an application to workflow verification. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 436–447. Springer, 2008.

[42] M. Llorens, J. Oliver, J. Silva, S. Tamarit, and G. Vidal. Dynamic slicing techniques for Petri nets. *Electronic Notes in Theoretical Computer Science*, 223:153–165, 2008.

[43] W. Yu, Z. Ding, and X. Fang. Dynamic slicing of Petri nets based on structural dependency graph and its application in system analysis. *Asian Journal of Control*, 17(4):1403–1414, 2015.

[44] M. Wang, Z. Ding, P. Zhao, W. Yu, and C. Jiang. A dynamic data slice approach to the vulnerability analysis of e-commerce systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(10):3598–3612, 2020.

[45] Y. Khan. Slicing high-level petri nets. 2014.

[46] P Chariyathitipong and W. Vatanawood. Dynamic slicing of time petri net based on mtl property. *IEEE Access*, 10:45207–45218, 2022.

[47] W. Yu, C. Yan, Z. Ding, and X. Fang. Extended and improved slicing technologies for Petri nets. *High Technology Letters*, 19(01):98–104, 2013.

[48] CPN Tools homepage. http://cpntools.org/.

[49] C. Xia and C. Li. Property preservation of Petri synthesis net based representation for embedded systems. *IEEE/CAA Journal of Automatica Sinica*, 8(4):905–915, 2021.

[50] Z. Liu, R. Zheng, W. Lu, and S. Xu. Using event-based method to estimate cybersecurity equilibrium. *IEEE/CAA Journal of Automatica Sinica*, 8(2):455–467, 2021.

[51] C. Liu, Q. Zeng, L. Cheng, H. Duan, M. Zhou, and J. Cheng. Privacy-preserving behavioral correctness verification of cross-organizational workflow with task synchronization patterns. *IEEE Transactions on Automation Science and Engineering*, 18(3):1037–1048, 2021.

[52] Q. Zeng, C. Liu, H. Duan, and M. Zhou. Resource conflict checking and resolution controller design for cross-organization emergency response processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(10):3685–3700, 2020.

[53] J. Luo, M. Zhou, and J. Q. Wang. A place-timed Petri net-based method to avoid deadlock and conflict in railway networks. *IEEE Transactions on Intelligent Transportation Systems*, doi: 10.1109/TITS.2021.3095661, 2021.

A APPENDIX

Table 4. The name and meaning of the places and transitions of module 1.

Places	Meaning	Transitions	Meaning
P_1	The vehicle enters the detection coil	T_1	The vehicle is passing through the detection coil
P_2	The vehicle has passed through the detection coil	T_2	The vehicle drives towards the capture coil
P_3	The message channel used to notify the ETC system that the vehicle has entered the detection coil	T_3	The vehicle is passing through the capture coil
P_4	The vehicle enters the capture coil	T_4	The vehicle drives towards the drop coil
P_5	The vehicle has passed through the capture coil	T_5	The vehicle is passing the railing
P_6	The vehicle enters the drop coil	T_6	The vehicle is leaving the drop coil
P_7	The vehicle has passed the railing	P_8	The vehicle has passed the drop coil

P_9	The message channel used to notify the ETC system that the vehicle is entering the capture coil	P_{10}	The message channel used to notify the ETC system that the vehicle has entered the drop coil

Table 5. The name and meaning of the places and transitions of module 2.

Places	Meaning	Transitions	Meaning
P_{11}	The ETC system is in a waiting state	T_7	The ETC system detects whether the vehicle has an OBU
P_{12}	OBU Information	T_8	This vehicle has no OBU and cannot continue in the ETC lane
P_{13}	Online database	T_9	Manual charge processing
P_{14}	Illegal OBU information	T_{10}	Move on to the next vehicle
P_{15}	Charge successfully	T_{11}	The information stored in the OBU is judged to be valid and legitimate according to the online database
P_{16}	Illegal IC information	T_{12}	The legitimate OBU information is temporarily stored in the system and whether the vehicle has an IC card is detected
P_{17}	Charge successfully	T_{13}	According to the online database, it is judged that the information stored by the OBU is illegal
P_{18}	Legal OBU information	T_{14}	Manual charge
P_{19}	The vehicle is transferred to the manual handling state	T_{15}	Move on to the next vehicle
P_{20}	The artificial channel is in a waiting state	T_{16}	According to the online database, the information stored in the IC card is judged to be illegal
P_{21}	charge successfully	T_{17}	This vehicle has no IC card and cannot continue in the ETC lane
P_{22}	Temporarily stored legal OBU information	T_{18}	According to the online database, the information stored in the IC card is judged to be valid and legal
P_{23}	The information stored in an IC card	T_{19}	The vehicle is transferred to the manual handling state
P_{24}	Legitimate IC information	T_{20}	Move on to the next vehicle
P_{25}	The IC information of the staging system	T_{21}	Check the OBU information and IC card information of the vehicle, and judge that the information is consistent
P_{26}	Entry information	T_{22}	The OBU and IC card information of the vehicle are inconsistent
P_{27}	deals are done	T_{23}	Reading entry information
P_{28}	Name of exit toll station	T_{24}	Charge successfully
P_{29}	The ETC system is in an intermittent state	T_{25}	Request to lift the rail in human resources
P_{30}	The message channel used to notify the railing controller to raise the railing	T_{26}	Send the command to lift the rail
P_{31}	The artificial lift rail is in a waiting state	T_{27}	Send a raised command
P_{32}	The message channel used to notify the railing controller to raise the railing	T_{28}	The manual receives the command for lifting the rail
P_{33}	Human taking orders	T_{29}	Raise the rail

P_{43}	license plate number	T_{30}	Proceed to charge the next vehicle
P_{44}	Legal license plate number	T_{31}	Receiving the command to raise the rail
P_{45}	Illegal license plate information	T_{40}	Recognize the license plate number
P_{46}	Charge successfully	T_{41}	Determine whether the license plate number is legitimate according to the online database
P_{47}	The information channel used to notify the railing controller to drop the railing	T_{42}	The license plate number is illegal
T_{43}	Manual charge	T_{44}	Send the command to lift the rail

Table 6. The name and meaning of the places and transitions of module 3.

Places	Meaning	Transitions	Meaning
P_{34}	The rail is in a landing state	T_{32}	Raise the rail
P_{35}	The railing controller accepts the command	T_{33}	Receive the command to land the rail
P_{36}	The railing is in a rising state	T_{34}	Receive the command to drop the rail
P_{37}	The rail controller has received a new command	T_{35}	Prepare to switch the traffic light
P_{38}	The rail is in a landing state	T_{36}	Switch the traffic light
P_{39}	The information channel used to inform the traffic light control system to switch the light	T_{37}	Switch the traffic light
P_{40}	Traffic light state	T_{38}	Prepare to switch the traffic light
P_{41}	The traffic light control system is in a ready state	T_{39}	Prepare to switch the traffic light
P_{42}	The information channel used to inform the traffic light control system to switch the light		

Wangyang Yu received the M.S. degree in computer software and theory from Shandong University of Science and Technology, Qingdao, China, in 2009, and the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2014. He is currently an Associate Professor with the College of Computer Science, Shaanxi Normal University, Xi'an, China. He was also a Visiting Scholar with the University of Derby, Derby, U.K., from 2016 to 2017. His research interests include the theory of Petri nets, formal methods in software engineering and trustworthy software.

Jinming Kong received her M.S. degree in computer software and theory from the School of Computer Science, Shaanxi Normal University, Xi'an, China, in 2022. She is currently an assistant teacher with Puyang Institute of Technology, Henan University, Puyang, China. Her research interests include the theory of Petri nets, formal methods in software engineering, and formal modeling of online social networks.

Zhijun Ding received the M.S. degree from Shandong University of Science and Technology, Taian, China, in 2001, and Ph.D. degree from Tongji University, Shanghai, China, in 2007. Now he is a Professor of the Department of Computer Science and Technology, Tongji University. His research interests are in formal engineering, Petri nets, services computing, and workflows. He has published more than 100 papers in domestic and international academic journals and conference proceedings.

Xiaojun Zhai received his Ph.D. in the area of Electrical and Electronic Engineering from University of Hertfordshire in 2013, United Kingdom. Before join in the University of Essex, Dr Zhai held a Senior Lecturer position in real-time software engineering at University of Derby. Prior to his appointment at UoD, Dr Zhai held a number of full-time research positions in the UK. Previously, he was research associate and senior research officer at University of Leicester and University of Essex, UK, respectively. He has published more than 140 publications in the area of reconfigurable computing, connected health application, Internet of Things, deep learning, image and signal processing and their accelerations using multi-core and FPGA systems during his career to date.

Zhiqiang Li received the PhD degree from Wuhan University, Wuhan, China, in 2018. He is currently an Associate Professor with the School of Computer Science, Shaanxi Normal University, Xi’an, China. He has published more than 20 research papers in the international journals and conferences, including TSE, TR, TIP, AUSE, STVR, and ICSME. His main research interests include intelligence software engineering and machine learning.

Qi Guo received the master’ s degree from the School of Computer Science, Shaanxi Normal University, in 2023. Her research interests include Petri net theory and Formal method.