

A refinement and abstraction method of the SPZN formal model for intelligent networked vehicles systems

Yang Liu¹, Yingqi Fan^{1*}, Ling Zhao¹ and Bo Mi¹

¹Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China
[e-mail: liuyang13@cqjtu.edu.cn, fanyingqi@mails.cqjtu.edu.cn, zhao.ling@163.com, mi_bo@163.com]

*Corresponding author: Yingqi Fan

*Received August 20, 2023; revised November 15, 2023; accepted January 11, 2024;
published January 31, 2024*

Abstract

Security and reliability are the utmost importance facts in intelligent networked vehicles. Stochastic Petri Net and Z (SPZN) as an excellent formal verification tool for modeling concurrent systems, can effectively handles concurrent operations within a system, establishes relationships among components, and conducts verification and reasoning to ensure the system's safety and reliability in practical applications. However, the application of a system with numerous nodes to Petri Net often leads to the issue of state explosion. To tackle these challenges, a refinement and abstraction method based on SPZN is proposed in this paper. This approach can not only refine and abstract the Stochastic Petri Net but also establish a corresponding relationship with the Z language. In determining the implementation rate of transitions in Stochastic Petri Net, we employ the interval average and weighted average method, which significantly reduces the time and space complexity compared to alternative techniques and is suitable for expert systems at various levels. This reduction facilitates subsequent comprehensive system analysis and module analysis. Furthermore, by analyzing the properties of Markov Chain isomorphism in the case study, recommendations for minimizing system risks in the application of intelligent parking within the intelligent networked vehicle system can be put forward.

Keywords: refinement, abstraction, SPZN, intelligent networked vehicle.

1. Introduction

The intelligent networked vehicle system is an innovative vehicle system that combines advanced computer technology and vehicle communication technology [1-5]. Its design goal is to provide everyone with a safer, more efficient and intelligent travel experience through real-time communication with surrounding vehicles, infrastructure and cloud servers. However, promising the correctness and reliability of the intelligent networked vehicle systems has been a challenging task due to their complexity and criticality [6-9].

To address these challenges, researchers often verify system functionality and performance by simulating real-world scenarios and performing virtual simulations using sensor data. Among many methods, formal methods are gradually emerging in the development and verification of intelligent networked vehicle systems. Formal method is a verification method based on mathematics and logic, which is used to formally describe the behavior and properties of the system, and reason and analyze it [10-12]. Formal methods are often widely used in biology [13,14], hardware [15], software [16], industry [17-19] and other fields. In intelligent networked vehicle systems, formal methods can be used to establish precise mathematical models to describe the states, events, and transition rules of the system. These models can be verified with system-level properties through formal verification techniques such as model checking, theorem proving, and symbolic execution [20-22].

Among various formal methods, Petri Net, as a powerful formal tool, can be used to describe the behavior of concurrent systems [23]. It has a clear semantic specification and graphical representation, can accurately describe the concurrent behavior and state transition of the system, and helps us to understand the overall working principle and interaction process of the system. Through the application of Petri Net, researchers and engineers can establish system models, and apply techniques such as model checking, state space analysis, and performance optimization for verification and analysis. In the intelligent networked vehicle system, Petri Net can be used to establish the behavior model of the system, describing the interaction and concurrent operation among vehicles, sensors, communication modules and control algorithms [24-26]. The performance and correctness of the system can be deduced by analyzing the state transition and transition trigger conditions of Petri Net. Compared with machine learning, this method does not require a large amount of data for verification in intelligent networked vehicle systems, and Petri Net are very suitable for concurrent structures in intelligent networked vehicle systems.

However, intelligent networked vehicle system is different from other systems. It has a large number of nodes, and using Petri Net for modeling will cause the problems of excessive model size and state space explosion. Based on this background, this paper proposes a refinement and abstraction method based on Stochastic Petri Net and Z (SPZN), 1) According to the different requirements required at different stages in formal verification, refined or abstract methods for analysis and verification respectively can be chosen. The refinement method can be used to conduct in-depth research on the underlying structure and processes of the system from a microscopic perspective. The use of abstract methods can study the entire system from a macro perspective and effectively avoid the state explosion problem by reducing the complexity of the model. 2) A method to determine the transition implementation rate is provided. Through the method of combining expert system and weighted average, the implementation rate of transition can be effectively determined. This method can be applied to situations where the professional level gap between experts is large and the expert opinions do not overlap. 3) The refining and abstraction methods proposed in this paper are practically applied effectively to reduce model complexity and avoid state explosion. And the transition

implementation rate determination method was used to effectively determine the transition implementation rate. Based on the steady-state probability transition trend, measures to improve safety and reliability were proposed.

This paper is structured as follows: Section 2 provides an overview of the relevant foundational knowledge. In Section 3, the SPZN-based refinement and abstraction techniques are comprehensively elaborated. Section 4 describes the nature of the model. Section 5 presents a case study that demonstrates the application of the proposed method. Section 6 offers a discussion and qualitative analysis of the case study results. Finally, Section 7 concludes the paper by giving final remarks, as well as suggestions for future works.

2. Fundamentals

2.1 Intelligent networked vehicles

The intelligent networked vehicle system consists of several key components and subsystems, which are used to realize various functions of the intelligent networked vehicle system such as automatic driving, navigation, and route optimization. At present, the most widely used and most recognized intelligent networked vehicle system structure is the integrated structure of vehicles, roads, people, and clouds. In this structure, connections can be established between vehicles, roads, people, and clouds to realize information exchange, as shown in **Fig. 1**. This figure clearly describes the information transmission among vehicles, roads, people, and clouds. The integrated structure of vehicles, roads, people and clouds provides people with a safer, more efficient and intelligent travel experience, and also provides more data for the traffic management department, and promotes the development of intelligent transportation.

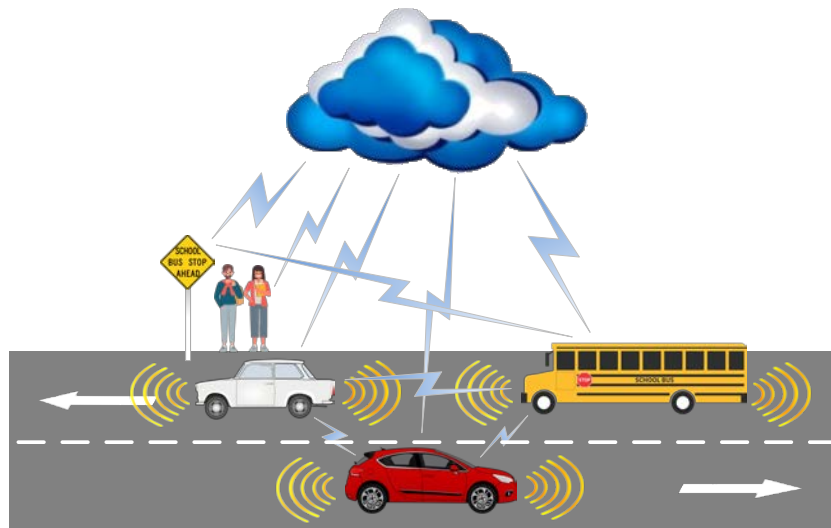


Fig. 1. Intelligent networked vehicle system architecture.

Fig. 1 simplifies the transmission of some information. In real life, the information flow of the intelligent networked vehicle system is far more complex than the above figure. The information flow between vehicles, roads, people, and clouds in the intelligent networked vehicle system mainly includes the following [27-29]:

(1) Vehicle perception and data collection: Intelligent networked vehicles sense the surrounding environment in real time through on-board sensors and collect vehicle data such as the current status, location, speed, tire pressure, etc.;

(2) Interaction between vehicles (V2V): Intelligent networked vehicles interact through the vehicle-to-vehicle communication network, such as: traffic information sharing, vehicle group coordinated driving, etc.;

(3) Interaction between vehicles and infrastructure (V2I): Intelligent networked vehicles interact with infrastructure to obtain information such as speed limit information, traffic signal status, and road congestion ahead, and transmit the current status and location of vehicles to the infrastructure;

(4) Interaction between vehicles and pedestrians (V2P): Intelligent networked vehicles sense the presence and behavior of pedestrians around the body through on-board sensors, and take corresponding measures according to different situations, such as: safety warning, braking, emergency avoidance, etc.;

(5) Interaction between the vehicle and the cloud (V2C): Intelligent networked vehicle uploads the collected data to the cloud for storage and processing, and the cloud uses technologies such as big data analysis and machine learning to analyze the vehicle data to realize real-time traffic status monitoring, road condition prediction, traffic optimization and other functions, and send decisions to intelligent networked vehicles according to the prediction results, such as route optimization, traffic accident warning, etc., and the vehicle will perform corresponding operations according to the cloud decision;

(6) Interaction between pedestrians and the cloud: pedestrians can obtain relevant personalized information from the cloud, such as: real-time bus information, weather forecast, walking navigation, etc.;

(7) Interaction between the infrastructure and the cloud: the infrastructure can upload the collected real-time road condition monitoring data and road condition monitoring data for a certain period of time to the cloud for storage and processing, and the cloud can assist in predicting and guiding intelligent networked vehicles based on the data before making relevant decisions, the cloud can also update the infrastructure information according to the current relevant policies, laws and regulations, such as changing the speed limit and traffic limit of the current road section.

The intelligent networked vehicle system is an intricate network composed of vehicles, roads, people and clouds. Among all elements in the system, the intelligent networked vehicles constitute a fundamental component. These vehicles, acting as mobile nodes, can sense real-time traffic and environmental conditions and transmit this information to roadside units and cloud servers. The significance of vehicle sensors in perceiving vehicle surroundings cannot be underestimated. Studies by Al-Turjman[30], Fayyad[31], Wang[32] and other mainstream intelligent networked vehicle companies have shown that most intelligent networked vehicles are equipped with at least ten sensors, such as radar sensors, laser sensors, and video monitors. Fig. 2 showcases the distribution map of the onboard sensors of intelligent networked vehicles. Generally, radar sensors are situated at the vehicle's four corners, while laser sensors are placed at the vehicle's front and back. Moreover, video sensors and GPS (Global Positioning System) receivers are installed on the roof. It is worth mentioning that the allocation of vehicle sensors varies among businesses, scientific researches, and practical applications.

The description above demonstrates that the intelligent networked vehicle system is a complex network with numerous concurrent structures and a high degree of randomness. Consequently, the paper utilizes SPZN [33], a formal modeling tool proficient in capturing concurrency and randomness.

2.2 Stochastic Petri Net and Z (SPZN)

A Petri Net is defined as a four-tuple $N = (P, T, F, M_0)$ [34-36].

(1) $P = \{p_1, p_2, \dots, p_n\}$ is a non-empty finite set containing all the place elements in the Petri Net, and usually the place is represented in the Petri Net in the form of a circle;

(2) $T = \{t_1, t_2, \dots, t_m\}$ is a non-empty finite set containing all the transition elements in the Petri Net, and usually the transition is represented in the Petri Net in the form of a rectangle;

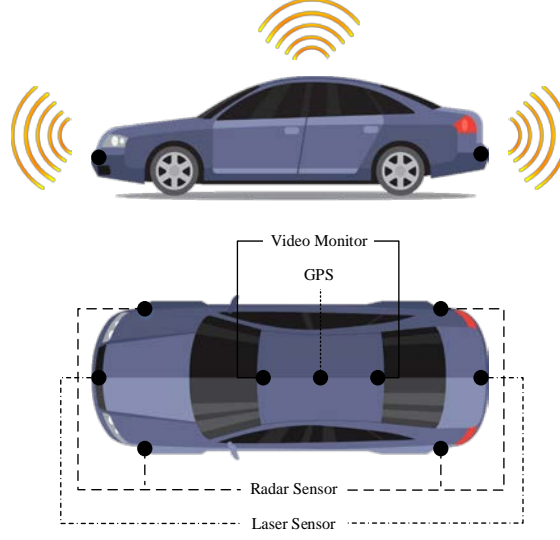


Fig. 2. Distribution of common on-board sensors in intelligent networked vehicles.

(3) $F = \{P \times T\} \cup \{T \times P\}$ is the set of directed arcs in the Petri Net, connecting the place and the transition, and there is no directed arc between any two places or two transitions;

(4) $M_0 = \{m_0(p_1), m_0(p_2), \dots, m_0(p_n)\}$ is the initial marking, which describes the initial marking of the Petri Net.

For the above four-tuple, the following conditions should be satisfied:

$$P \cap T = \emptyset \quad (1)$$

$$P \cup T \neq \emptyset \quad (2)$$

Stochastic Petri Net (SPN) [37] is an advanced Petri Net that can effectively model the description of uncertainty systems by adding the transition implementation rate λ to the Petri Net, so it is defined as a five-tuple $N = (P, T, F, M_0, \lambda)$, Where $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$ is the set of transition implementation rates, λ_i is a non-negative real number, and there exists a transition implementation rate λ_i corresponding to each transition t_i . The transition implementation rate is opposed to the transition trigger delay, which is also called the transition service time, and if the transition trigger delay is expressed by T_{delay} , then the transition implementation rate and the transition trigger delay satisfies the following equation:

$$T_{\text{delay}} \times \lambda = 1 \quad (3)$$

The trigger rules of SPN are defined as follows: when there exists a transition t_i to satisfy the trigger condition, then after the time delay of the transition trigger, the token flows from the place p_a before t_i to the library p_b after t_i . Meanwhile, the reachable marking M before the trigger also changes with the transition t_i to produce a new reachable marking M' , and the passing rules of token satisfies the following equation:

$$m'(p_a) = m(p_a) - Pre(p_a, t_i) + Post(p_a, t_i) \quad (4)$$

$$m'(p_b) = m(p_b) - Pre(p_b, t_i) + Post(p_b, t_i) \quad (5)$$

The meaning of $Pre(p_a, t_i)$ is the number of output tokens in the place p_a required for the trigger of transition t_i , and the meaning of $Post(p_b, t_i)$ is the number of input tokens after the trigger of transition t_i .

SPZN combines Stochastic Petri Net with the Z language to enhance the descriptive power of the model based on Stochastic Petri Net, which is defined as a nine-tuple $N = (P, T, F, M_0, \lambda, Z_P, Z_T, S, C)$, where

- (1) $Z_P = \{Z_{p_1}, Z_{p_2}, \dots, Z_{p_n}\}$ is a set of Z that each place is based on.
- (2) $Z_T = \{Z_{t_1}, Z_{t_2}, \dots, Z_{t_m}\}$ is a set of Z that each transition is based on.
- (3) $S: P \rightarrow Z_P$ is a set of the one-to-one map relationship between P and Z_P .
- (4) $C: T \rightarrow Z_T$ is a set of the one-to-one map relationship between T and Z_T .
- (5) $PN = (P, T, F, M_0)$ is a Petri Net.
- (6) $SPN = (P, T, F, M_0, \lambda)$ is a Stochastic Petri Net.
- (7) $PZN = (P, T, F, Z_P, Z_T, S, C)$ is a Petri Net and Z (PZN),

As shown in the Fig. 3, the correspondence between SPZN-SPN and SPZN-Z in SPZN can be clearly seen. In Z_{t_i} , the assertions are divided into pre-assertion and post-assertion, which correspond to the pre-condition and post-conditions of the transition t_i , respectively.

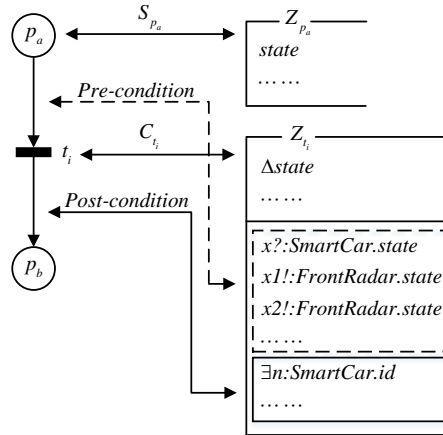


Fig. 3. The relationship between SPN and Z in SPZN.

In SPN, it has been proved that its marking graph is isomorphic to Markov Chain (MC), and each marking in SPN corresponds to the state in MC [38]. For a MC that is isomorphic to a SPN, where the MC has l states, we can define a transition matrix Q of $l \times l$ order.

When $c \neq d$, if $\exists t_i \in T$ that makes $M_c[t_i] > M_d$, then

$$q_{cd} = \left. \frac{d(1 - e^{-\lambda_i \tau})}{d\tau} \right|_{\tau=0} = \lambda_i \quad (6)$$

else,

$$q_{cd} = 0 \quad (7)$$

When $c = d$, then

$$q_{cd} = \left. \frac{d(e^{-\tau \sum_i \lambda_i})}{d\tau} \right|_{\tau=0} = - \sum_i \lambda_i \quad (8)$$

where λ_i is the average implementation rate of transition t_i .

By constructing the reachable marking graph of the SPN, we can obtain that the reachable marking graph and its isomorphic MC have l states. The steady-state probability from marking

M_0 to marking M_{l-1} is an n-dimensional vector P , then $P = (P(M_0), P(M_1) \cdots P(M_{l-1}))$, where $P(M_i)$ is the steady-state probability of marking M_i . According to the Markov process, there are the following equations. By solving this system of equations we can obtain the steady-state probability $P(M_i)$ for each reachable marking M_i .

$$\begin{cases} P \times Q = 0 \\ \sum_{i=0}^{l-1} P(M_i) = 1 \end{cases} \quad (9)$$

2.3 Determination of the transition trigger delay

Liu et al. [39] made significant advancements in determining the transition trigger delay. They introduced the use of triangular membership functions and adopted the regional center method to determine the interval for the transition trigger delay. These two methods enable the quantitative representation of expert knowledge and the standardization of different expert opinions.

However, the aforementioned method exhibits a high level of complexity and is deemed unsuitable for intelligent networked vehicle systems that have a substantial number of nodes, and is limited to triangular and symmetric cases for the membership functions. When the trigger delay intervals provided by different experts do not overlap, the above method cannot effectively solve the problem. To address these issues, this study proposes the method of weighted average after the interval average. This approach effectively resolves the aforementioned limitations. Additionally, when different experts possess different specialties, we can evaluate and assign corresponding weights to them. Then, we take the average value of the interval provided by each expert and calculate the final result using the weighted average method.

Assuming that expert u estimates the interval of the trigger delay for transition v as $[x, y]$, calculating the average trigger delay for this transition can be accomplished easily, as shown in (10), $TA_{u,v}$ is the average trigger delay estimate of expert u for transition v . Subsequently, the average trigger delays of u experts for v transitions can be represented by a matrix, as indicated in (11),

$$TA_{u,v} = \frac{x + y}{2} \quad (10)$$

$$TA = \begin{pmatrix} TA_{1,1} & \cdots & TA_{1,v} \\ \vdots & \ddots & \vdots \\ TA_{u,1} & \cdots & TA_{u,v} \end{pmatrix} \quad (11)$$

Weight vectors for the u experts are obtained following evaluation, considering their professionalism and reliability, as shown in (12),

$$F = (f_1, f_2, \cdots, f_u) \quad (12)$$

Equation 13 of the weighted average algorithm allows us to readily derive (14), where $TA_{transition}$ represents the vector of weighted average time delays across all transitions.

$$\bar{x} = \frac{x_1 f_1 + x_2 f_2 + \cdots + x_k f_k}{\sum_1^k f_i} \quad (13)$$

$$TA_{transition} = \frac{F * TA}{\sum_1^u f_i} = \left(\frac{f_1}{\sum_1^u f_i}, \frac{f_2}{\sum_1^u f_i}, \cdots, \frac{f_u}{\sum_1^u f_i} \right) \begin{pmatrix} TA_{1,1} & \cdots & TA_{1,v} \\ \vdots & \ddots & \vdots \\ TA_{u,1} & \cdots & TA_{u,v} \end{pmatrix} = (TA_1, TA_2, \cdots, TA_v) \quad (14)$$

3. Methodology

In response to the state explosion problem that arises when using Petri Net for formal modeling and analysis in complex systems, this section proposes an abstraction and refining method based on SPZN. On the premise of the safety, boundedness and reachability of the model, according to different stages and different needs, choose an abstract method or a refining method. Using the abstract method, the key elements in each sub-model are extracted and abstracted into a simplified and complete system model, and then the overall function of the system is analyzed according to the model. Using the refining method, each sub-model is refined separately, the details of the workflow in the sub-module are analyzed, and the functions in the sub-model are analyzed and refined. Fig. 4 clearly and completely shows the work flow of the abstract method and the refined method. In this section, we propose five refinement and abstraction methods for SPZN, each of which is reversible.

- (1) The abstract method of the place and the refinement method of the transition.
- (2) The abstract method of transition and the refinement method of place.
- (3) Abstract method and refinement method for multiple places.
- (4) Abstract method and refinement method for multiple transitions.
- (5) Mixed refinement methods and abstract methods.

As shown in Fig. 5, this figure is the first method, which abstracts a place p_a and two transitions t_i, t_j into a transition t^* . After t_i meets the trigger condition, the token flows into p_a , and t_j triggers after the trigger condition of t_j is met, and the token in p_a flows out to the subsequent part. The place p_a has one and only one input arc connected to t_i and one output arc connected to t_j . Suppose the number of input arcs and output arcs of transition t_i is I_{t_i}, O_{t_i} , the number of input arcs and output arcs of transition t_j is I_{t_j}, O_{t_j} , and the number of input arcs and output arcs of transition t^* is I_{t^*}, O_{t^*} , then this method satisfies the following equation:

$$\begin{cases} I_{t_i} + I_{t_j} = I_{t^*} + 1 \\ O_{t_i} + O_{t_j} = O_{t^*} + 1 \end{cases} \quad (15)$$

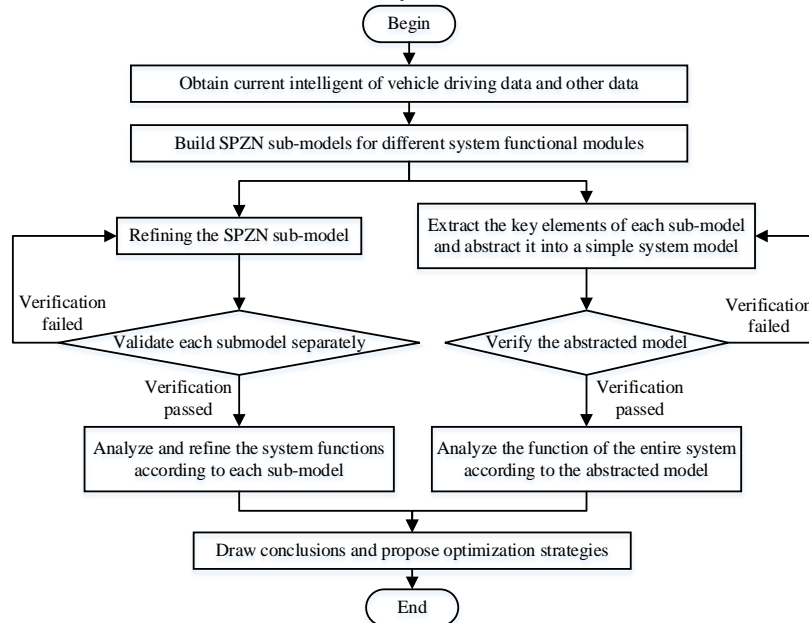


Fig. 4. Workflow diagram of the refinement and abstraction method based on SPZN.

While abstracting, the Z language will also change accordingly. After the abstraction, the Z language contains all the content of place and transition before the abstraction. Therefore, there is a set relationship $Z_{t^*} = \{Z_{t_i}, Z_{t_j}, Z_{p_a}\}$ correspond to this abstraction.

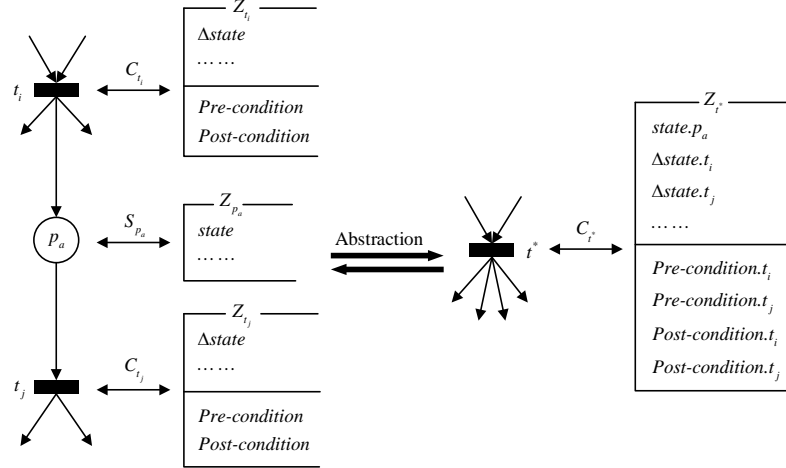


Fig. 5. The abstract method of the place and the refinement method of the transition.

Fig. 6 shows the second method, which abstracts two places p_a , p_b and a transition t_i into one place. After t_i meets the trigger condition, the token in p_a flows into p_b through t_i . Transition t_i has one and only one input arc connected to p_a and one output arc connected to p_b . Suppose the number of input arcs and output arcs of place p_a is I_{p_a} , O_{p_a} , the number of input arcs and output arcs of place p_b is I_{p_b} , O_{p_b} , and the number of input arcs and output arcs of place p^* is I_{p^*} , O_{p^*} , then this method also satisfies the following equation:

$$\begin{cases} I_{p_a} + I_{p_b} = I_{p^*} + 1 \\ O_{p_a} + O_{p_b} = O_{p^*} + 1 \end{cases} \quad (16)$$

While abstracting, the Z language will also change accordingly. After the abstraction, the Z language contains all the content of place and transition before the abstraction. Therefore, there is a set relationship $Z_{p^*} = \{Z_{p_a}, Z_{p_b}, Z_{t_i}\}$ correspond to this abstraction.

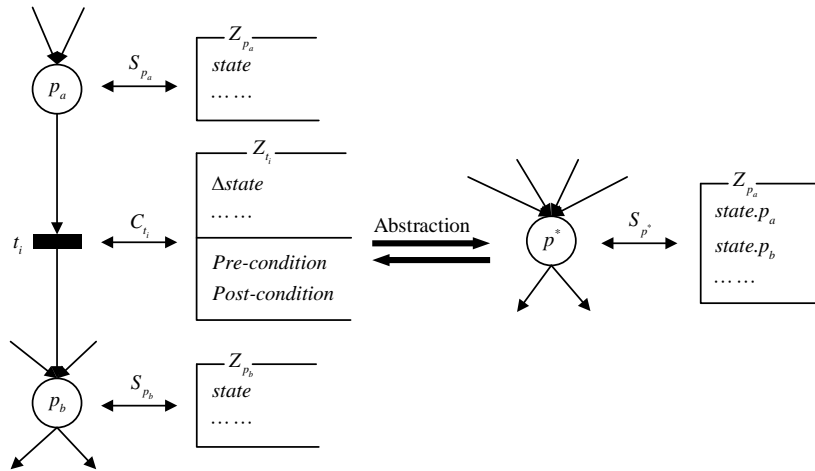


Fig. 6. The abstract method of the transition and the refinement method of the place.

The third method is shown in Fig. 7. This method abstracts multiple places into one place and satisfies the set relationship $Z_{p^*} = \{Z_{p_a}, \dots, Z_{p_b}\}$. Assuming that there are c places abstracted into one place, the number of output arcs of transition t_i is O_{t_i} , and the number of input arcs of transition t_j is I_{t_j} . Then in the fourth method, after the abstraction, both the output arc of t_i and the input arc of t_j will reduce by c . The number of output arcs of t_i is $O_{t_i} - k$, and the number of input arcs of t_j is $I_{t_j} - k$.

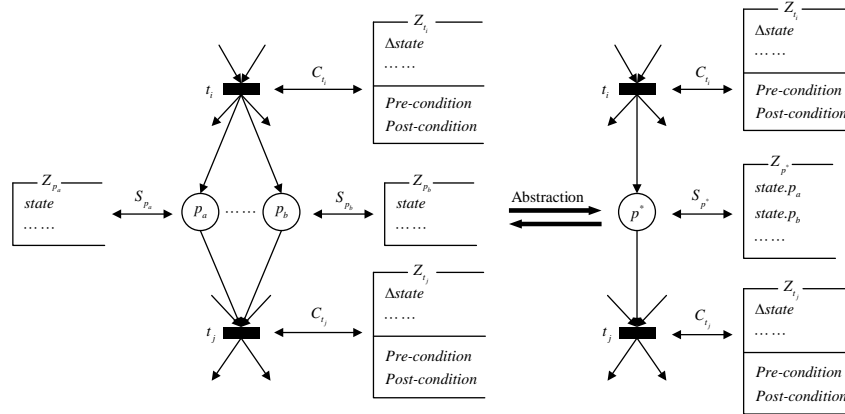


Fig. 7. Abstract method and refinement method for multiple places.

The fourth method is shown in Fig. 8. This method abstracts multiple transitions into one transition, and satisfies the set relationship $Z_{t^*} = \{Z_{t_i}, \dots, Z_{t_j}\}$. Assuming that k transitions are abstracted into one transition, the number of output arcs of place p_a is O_{p_a} , and the number of input arcs of place p_b is I_{p_b} . Then in the fourth method, after abstraction, the output arcs of p_a and the input arcs of p_b will be reduced by k . The number of output arcs of p_a is $O_{p_a} - k$, and the number of input arcs of p_b is $I_{p_b} - k$.

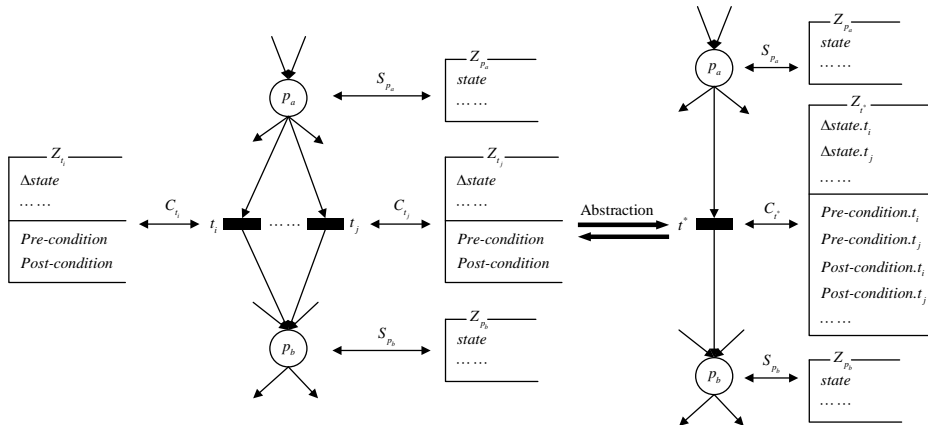


Fig. 8. Abstract method and refinement method for multiple transitions.

In the fifth mixed refinement method and abstraction method is shown in Fig. 9, this method abstracts two places and two transitions into one place and one transition, and the corresponding set relationship satisfies: $Z_{t^*} = \{Z_{t_i}, Z_{t_j}\}$, $Z_{p^*} = \{Z_{p_a}, Z_{p_b}\}$. Suppose the number of input arcs of place p_a is I_{p_a} and the number of input arcs of place p_b is I_{p_b} , and the number of input arcs of place p^* is I_{p^*} . Place p_a has one and only one output arc connected to

transition t_i , and place p_b has one and only one output arc connected to transition t_j . Both transition t_i and transition t_j have one and only one input arc and output arc, and both output arcs are connected with place p_c . After abstraction, the transition t^* has one and only one input arc connected to p^* , and one output arc connected to p_c , then this method satisfies the following formula:

$$I_{p_a} + I_{p_b} = I_{p^*} \quad (17)$$

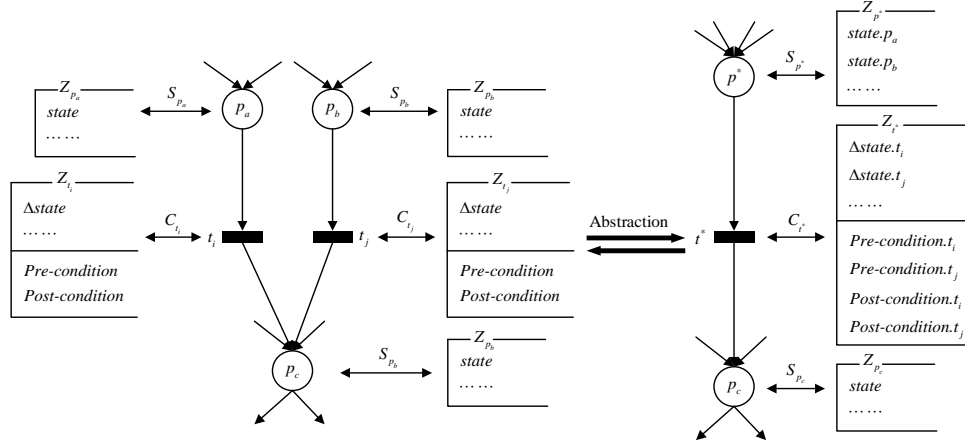


Fig. 9. Mixed refinement methods and abstract methods.

4. Modeling analysis

4.1 Reachability

Reachability is the most basic dynamic property of Petri Net, and other properties need to be defined through reachability.

Definition 1. Let $\Sigma = (P, T, F, M_0)$ be a Petri Net, if there is $\exists t \in T$ that makes $M[t > M']$, then M' is called directly reachable from M . If there are transition sequences t_0, t_1, \dots, t_{k-1} and marking sequences M_0, M_1, \dots, M_k , such that $M_0[t_0 > M_1[t_1 > M_2 \dots M_{k-1}[t_{k-1} > M_k]$, then M_k is called reachable from M_0 . The set of all markings reachable from M_0 is denoted as $R(M_0)$.

According to the description of Definition 1, the algorithm for verifying the reachability of Petri Net as shown in Algorithm 1 can be obtained.

Algorithm 1: Algorithm for Verifying the Reachability of Petri Net Σ

Input: Σ .

Output: Reachability of Petri Net $R(\Sigma)$.

if $\exists t \in T : M_0[t > M$

 | $M \in R(M_0)$
 | $R(M_0) = \{M\}$

else if $\exists t_0, t_1, \dots, t_{k-1} \in T : M_0[t_0 > M_1[t_1 > M_2 \dots M_{k-1}[t_{k-1} > M_k]$

 | $M_1, M_2, \dots, M_k \in R(M_0)$
 | $R(M_0) = \{M_1, M_2, \dots, M_k\}$

else

 | $R(M_0) = \{\emptyset\}$

 | Traverse all elements of $R(M_0)$.

if $\exists i : R(M_i) \notin R(M_0)$

 | $R(\Sigma) = 0$

else

 | $R(\Sigma) = 1$

4.2 Boundedness and safety

According to the definition of reachability, boundedness and safety are very necessary in Petri Net. When a certain part of the system does not satisfy the safety requirements, relevant solutions should be taken to prevent unsafe conditions. If the system satisfies safety, it means that the probability of the system being unsafe is very small, but it does not mean that the system will always maintain a safe state.

Definition 2. Let $\Sigma = (P, T, F, M_0)$ be a Petri Net, if $\exists B \in \mathbb{Z}^+$ such that $\forall M \in R(M_0): M(p) \leq B$, then the place p is called bounded, and the smallest positive integer B that satisfies this condition is called the bound of place p , denoted as $B(p)$,

$$B(p) = \min\{B | \forall M \in R(M_0): M(p) \leq B\} \quad (18)$$

when $B(p) = 1$, the place p is said to be safe.

Definition 3. Let $\Sigma = (P, T, F, M_0)$ be a Petri Net, if any place is bounded, it is called a bounded Petri Net.

$$B(\Sigma) = \max\{B(p) | p \in P\} \quad (19)$$

we call $B(\Sigma)$ is the bound of Σ . When $B(\Sigma) = 1$, the Petri Net Σ is called be safe.

According to the description of Definition 2 and Definition 3, the algorithm for verifying the boundedness and safety of Petri Net shown in Algorithm 2 can be obtained. Among them, $S(p)$ and $S(\Sigma)$ respectively represent the safety of place p and Petri Net Σ . If $S(p) = 0$ or $S(\Sigma) = 0$, then the place p or the Petri Net Σ does not satisfy safety. If $S(p) = 1$ or $S(\Sigma) = 1$, then the place p or the Petri Net Σ satisfies safety.

Algorithm 2: Algorithms for Verifying the Boundedness and Safety of Petri Net Σ

Input: Σ .

Output: Boundedness of Petri Net $BD(\Sigma)$ and Safety of Petri Net $S(\Sigma)$.

```

if  $\forall M \in R(M_0) : M(p) \leq B$ 
   $B(p) = \min\{B | \forall M \in R(M_0) : M(p) \leq B\}$ 
  if  $B(p) = 1$ 
     $S(p) = 1$ 
  else
     $S(p) = 0$ 
  if  $\forall p \in P, \exists B(p)$ 
     $BD(\Sigma) = 1$ 
     $B(\Sigma) = \max\{B(p) | p \in P\}$ 
    if  $B(\Sigma) = 1$ 
       $S(\Sigma) = 1$ 
    else
       $S(\Sigma) = 0$ 
  else
     $BD(\Sigma) = 0$ 
else
   $BD(\Sigma) = 0$ 
   $S(\Sigma) = 0$ 

```

5. Case study

In this chapter, considering the extensiveness and complexity of the intelligent networked vehicle system in actual application scenarios, we give a simple application of intelligent networked vehicle in smart parking scenarios, as shown in Fig. 10. In this application scenario, the intelligent networked vehicle and the intelligent parking system communicate with the cloud respectively, and the cloud plans the parking location according to the remaining parking

spaces, and communicates with the intelligent networked vehicle and the intelligent parking system to complete smart parking. Fig. 11 clearly describes the process of smart parking.

According to the flow chart shown in Fig. 4, we first establish corresponding SPZN-SPN sub-models for different functional modules in the above scenarios. In this scenario, it mainly involves three modules: smart car, cloud and smart parking system. After the sub-models of the above three modules are established, according to the business relationship and workflow among the sub-modules of the system, the connection between the modules is established, as shown in Fig. 12. The meanings of corresponding places and transitions are shown in Table 1 and Table 2.

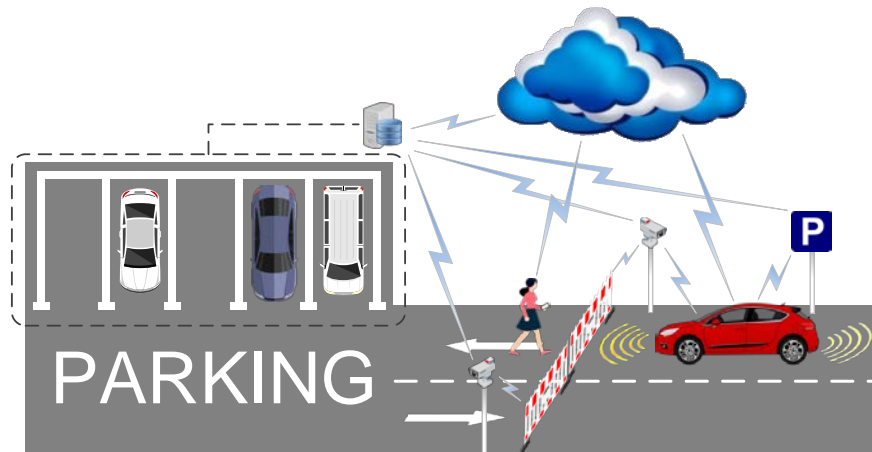


Fig. 10. Smart parking scene.

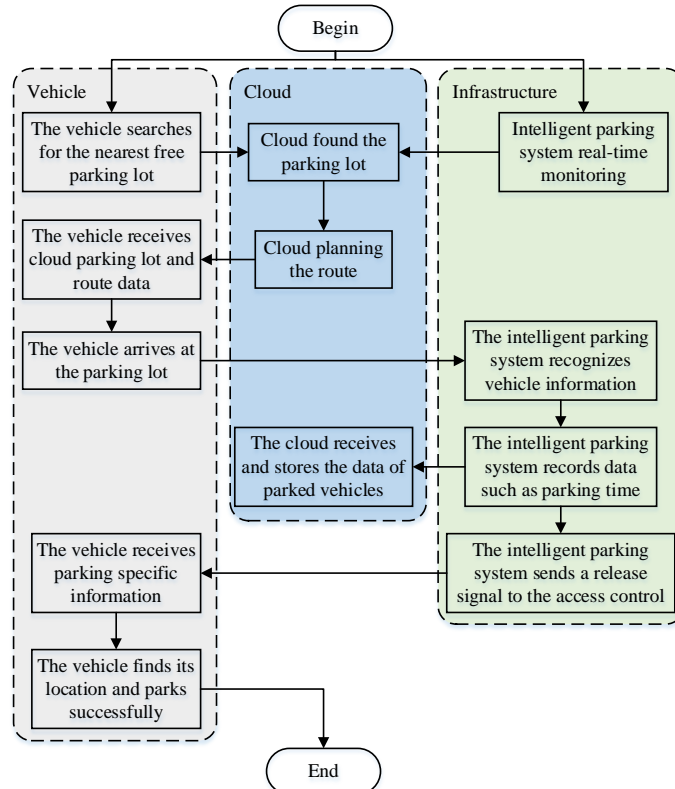


Fig. 11. Smart parking application process.

Table 1. Meanings of places.

Place	Meaning	Place	Meaning
p_1	Vehicle start status	p_{16}	Environment perception module
p_2	Vehicle parking status	p_{17}	Vehicle communication module
p_3	Front laser sensor	p_{18}	System monitoring status
p_4	Rear laser sensor	p_{19}	Monitoring module
p_5	Front left radar sensor	p_{20}	Data statistical analysis module
p_6	Right front radar sensor	p_{21}	Payment and settlement module
p_7	Left rear radar sensor	p_{22}	User service module
p_8	Right rear radar sensor	p_{23}	Cloud communication API
p_9	Front video monitor	p_{24}	Cloud data receiving module
p_{10}	Rear video monitor	p_{25}	Data processing and analysis module
p_{11}	Positioning receiver	p_{26}	Vehicle management module
p_{12}	Laser sensing subsystem	p_{27}	Traffic management module
p_{13}	Radar sensing subsystem	p_{28}	Application module
p_{14}	Video surveillance subsystem	p_{29}	Decision module
p_{15}	Positioning subsystem		

Table 2. Meanings of transition.

Transition	Meaning	Transition	Meaning
t_1	The vehicle issues a parking command	t_{12}	Send data to vehicle management module
t_2	The parking command is passed on to the sensors	t_{13}	Send data to traffic management module
t_3	Send the sensing data to the laser subsystem	t_{14}	Send data to application module
t_4	Send the sensing data to the radar subsystem	t_{15}	Send data to decision module
t_5	Send the sensing data to the video surveillance subsystem	t_{16}	Send data to decision module
t_6	Send the sensing data to positioning subsystem	t_{17}	Send data to decision module
t_7	Each subsystem sends data to the environment perception module	t_{18}	Send decision to vehicle perception module
t_8	Send data to the vehicle communication module	t_{19}	Upload sensing data to the cloud
t_9	Smart parking system sends signal	t_{20}	Upload system data to the cloud
t_{10}	Send data to cloud communication API	t_{21}	Send decision to smart parking system
t_{11}	Store and send data to data processing module		

Table 3. The set of reachable marking $R(M_0)$,

Marking	Corresponding place	Marking	Corresponding place
M_1	(2)	M_{15}	(9,10,12,13,15)
M_2	(3,4,5,6,7,8,9,10,11)	M_{16}	(3,4,13,14,15)
M_3	(5,6,7,8,9,10,11,12)	M_{17}	(12,13,14,15)
M_4	(3,4,9,10,11,13)	M_{18}	(16)
M_5	(3,4,5,6,7,11,14)	M_{19}	(17)
M_6	(3,4,5,6,7,10,15)	M_{20}	(24)
M_7	(9,10,11,12,13)	M_{21}	(25)
M_8	(5,6,7,8,11,12,14)	M_{22}	(26)
M_9	(5,6,7,8,9,10,12,15)	M_{23}	(27)
M_{10}	(3,4,11,13,14)	M_{24}	(28)
M_{11}	(3,4,9,10,13,15)	M_{25}	(29)
M_{12}	(3,4,5,6,7,8,14,15)	M_{26}	(18)
M_{13}	(11,12,13,14)	M_{27}	(19,20,21,22)
M_{14}	(5,6,7,8,12,14,15)	M_{28}	(23)

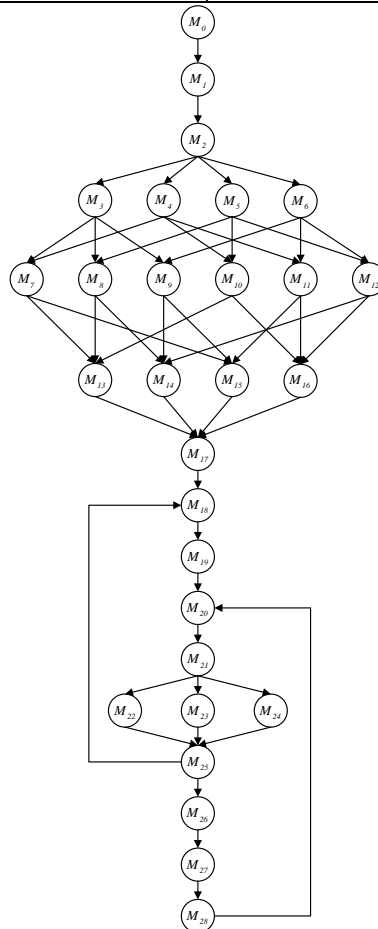


Fig. 14. The reachable marking tree.

It can be seen that the number of intelligent networked vehicle terminal nodes is large, the number of states is increasing exponentially, the complexity of the SPZN model is high, and the reachable marking tree constructed is large, which hinders the analysis and verification of the subsequent model. According to the method of refinement and abstraction based on the SPZN formal model proposed in Section 3 of this paper, each module is abstracted, and essential elements are extracted from each module to form concise and complete system modules, as depicted in Fig. 15. The meanings of places and transitions are shown in Table 4. After abstraction, the set relationship corresponding to Z language is shown in Table 5.

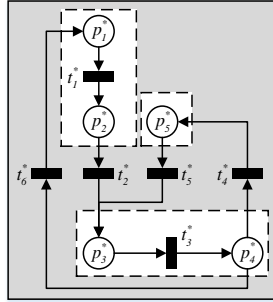


Fig. 15. Use the abstract method to abstract the SPZN model.

Table 4. Meanings of places and transitions.

Place	Meaning	Transition	Meaning
p_1	Vehicle perception module that receives parking command	t_1	Send the parking command to the vehicle communication module
p_2	Vehicle communication module	t_2	Upload sensing data to the cloud
p_3	Cloud data receiving module	t_3	The data is processed and sent to the decision module
p_4	Decision module	t_4	Send decision to smart parking system
p_5	Smart parking system	t_5	Upload system data to the cloud
		t_6	Send decision to vehicle perception module

Table 5. The set relationship corresponding to the Z language.

Abstracted	Original
$Z_{p_1}^*$	$\{Z_{p_1}, Z_{p_2}, Z_{p_3}, Z_{p_4}, Z_{p_5}, Z_{p_6}, Z_{p_7}, Z_{p_8}, Z_{p_9}, Z_{p_{10}}, Z_{p_{11}}, Z_{p_{12}}, Z_{p_{13}}, Z_{p_{14}}, Z_{p_{15}}, Z_{p_{16}}, Z_{t_1}, Z_{t_2}, Z_{t_3}, Z_{t_4}, Z_{t_5}, Z_{t_6}, Z_{t_7}\}$
$Z_{p_5}^*$	$\{Z_{p_{18}}, Z_{p_{19}}, Z_{p_{20}}, Z_{p_{21}}, Z_{p_{22}}, Z_{p_{23}}, Z_{t_9}, Z_{t_{10}}\}$
$Z_{t_3}^*$	$\{Z_{p_{25}}, Z_{p_{26}}, Z_{p_{27}}, Z_{p_{28}}, Z_{t_{11}}, Z_{t_{12}}, Z_{t_{13}}, Z_{t_{14}}, Z_{t_{15}}, Z_{t_{16}}, Z_{t_{17}}\}$

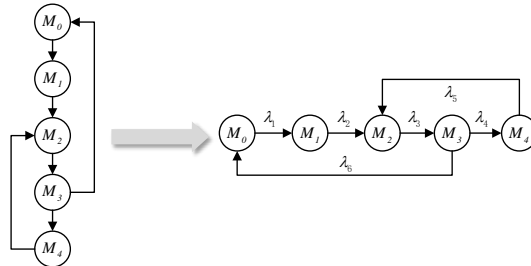


Fig. 16. Markov Chain with reachable marking graph isomorphism.

After abstracting the SPZN model, assuming that the initial marking is $(1,0,0,0,0)$, the corresponding isomorphic MC chain can be easily obtained according to the triggering rules, as shown in **Fig. 16**. Consequently, we apply the method described in Section 2.3 for determining the transition trigger delay, and integrate the transition trigger delay intervals provided by various experts. **Table 6** displays the trigger delay intervals provided by the six experts.

Table 6. Transition implementation delay interval. (unit: ms)

Transition	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Expert 6
t_1	[5,10]	[5,7]	[4,9]	[5,9]	[4,10]	[6,9]
t_2	[20,25]	[17,30]	[15,20]	[22,25]	[18,31]	[15,25]
t_3	[50,75]	[40,55]	[50,65]	[55,73]	[40,50]	[50,60]
t_4	[10,13]	[5,15]	[5,10]	[10,18]	[5,10]	[8,10]
t_5	[30,35]	[28,33]	[25,35]	[30,33]	[28,38]	[30,38]
t_6	[40,45]	[38,55]	[35,45]	[42,47]	[38,53]	[35,48]

According to (10), a matrix depicting the average trigger delay for six experts across six transitions can be obtained, as demonstrated in (20), considering the varying professional levels of these six experts, a weight vector $F = (0.23, 0.08, 0.13, 0.2, 0.18, 0.18)$ is derived after evaluation. Subsequently, utilizing (14), the weighted average time delay vector for the six transitions can be determined, as presented in (21). The implementation rate of each transition can be easily obtained by considering the reciprocal relationship between the transition implementation rate and the transition trigger delay, as illustrated in **Table 7**.

$$TA^T = \begin{pmatrix} 7.5 & 6 & 6.5 & 7 & 7 & 7.5 \\ 22.5 & 23.5 & 17.5 & 23.5 & 29.5 & 20 \\ 62.5 & 47.5 & 57.5 & 64 & 45 & 55 \\ 11.5 & 10 & 7.5 & 14 & 7.5 & 9 \\ 32.5 & 30.5 & 30 & 31.5 & 33 & 34 \\ 42.5 & 46.5 & 40 & 44.5 & 45.5 & 41.5 \end{pmatrix} \quad (20)$$

$$TA_{transition} = (7.060, 22.940, 56.450, 10.190, 32.175, 43.255) \quad (21)$$

Table 7. Transition implementation rate.

Transition	Transition implementation rate	Transition	Transition implementation rate
λ_1	0.141643	λ_4	0.098135
λ_2	0.043592	λ_5	0.031080
λ_3	0.017715	λ_6	0.023119

The steady-state probability for various markings can be calculated using (9), **Table 8** presents the steady-state probabilities for each flag displayed in **Fig. 16**, considering the transition implementation rates from **Table 7**.

Based on the algorithms for reachability, boundedness, and safety described in Section 4, we can verify the abstraction process detailed above. **Table 9** displays the verification results for both the original SPZN model and the abstracted SPZN model.

Table 8. The steady-state probability for each marking.

Marking	Steady-state probability
$M_0(1,0,0,0,0)$	0.0140
$M_1(0,1,0,0,0)$	0.0453
$M_2(0,0,1,0,0)$	0.5852
$M_3(0,0,0,1,0)$	0.0855
$M_4(0,0,0,0,1)$	0.2700

It is important to acknowledge that the model can be refined or abstracted as long as the conditions for refinement and abstraction are met. The selection of refined and abstract methods depends on specific usage requirements. In this chapter, only abstract methods within the refinement and abstraction methods are presented. If a more thorough examination of each functional module in the smart parking scenario is desired, it is possible to refine or abstract each sub-model depicted in **Fig. 12**. However, due to space limitations, this paper does not delve into an in-depth discussion.

Table 9. Verification results of reachability, safety, and boundedness.

Property	Reachability	Boundedness	Safety
Original	√	√	√
Abstracted	√	√	√

6. Results and discussions

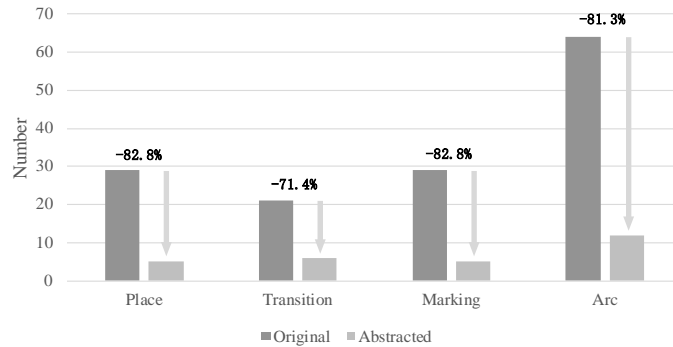
In the study by Liu et al. [33], the SPZN model has been proven to have good performance, as shown in **Table 10**. It can be seen from the case in Section 5 that after refining the established SPZN model, the number of places, transitions, and reachable marking is significantly reduced, relevant data are shown in **Table 11** and **Fig. 17**. When the SPZN model is applied to large and complex systems, the corresponding model size becomes larger and the complexity will increase accordingly. It will become very difficult to verify the model by applying relevant mathematical rules. According to different requirements, we can analyze a large and complex system model by using refined and abstract methods. And according to the definition of the nature of the model and mathematical rules, we can verify that the model is safe and stable before and after refinement and abstraction.

Table 10. Difference between PN, PZN, SPN, and SPZN.

	Framework	Dynamic	Randomness	Abstraction
PN	√	√	×	×
PZN	√	√	×	√
SPN	√	√	√	×
SPZN	√	√	√	√

Table 11. The number of places, transitions, and markings before and after abstraction.

	Place	Transition	Marking	Arc
Original	29	21	29	64
Abstracted	5	6	5	12

**Fig. 17.** The quantity trends of place, transition, marking and arc after abstraction.

Based on the transition implementation rates determined by the method proposed in this paper as shown in [Table 7](#), we changed all the transition implementation rates one by one, and calculated the corresponding steady-state probabilities, and obtained the result trend as shown in [Fig. 18](#). It should be noted that when changing the implementation rate of a certain transition, other conditions remain unchanged. By changing the transition implementation rate, we have the ability to dynamically analyze the abstracted SPZN. This enables us to observe the correlation between the different modules within the entire intelligent networked vehicle system. To enhance the security and stability of the system, it is recommended to undertake the following measures:

- (1) Regular maintenance should be conducted on intelligent connected vehicles to enhance their safety.
- (2) The performance of the cloud module should be improved to enhance its data processing capability.
- (3) Establishing a more stable and faster data transmission channel can significantly improve the stability and efficiency of data transmission between modules.
- (4) It is crucial to maintain the intelligent parking system to ensure real-time performance of data transmission.

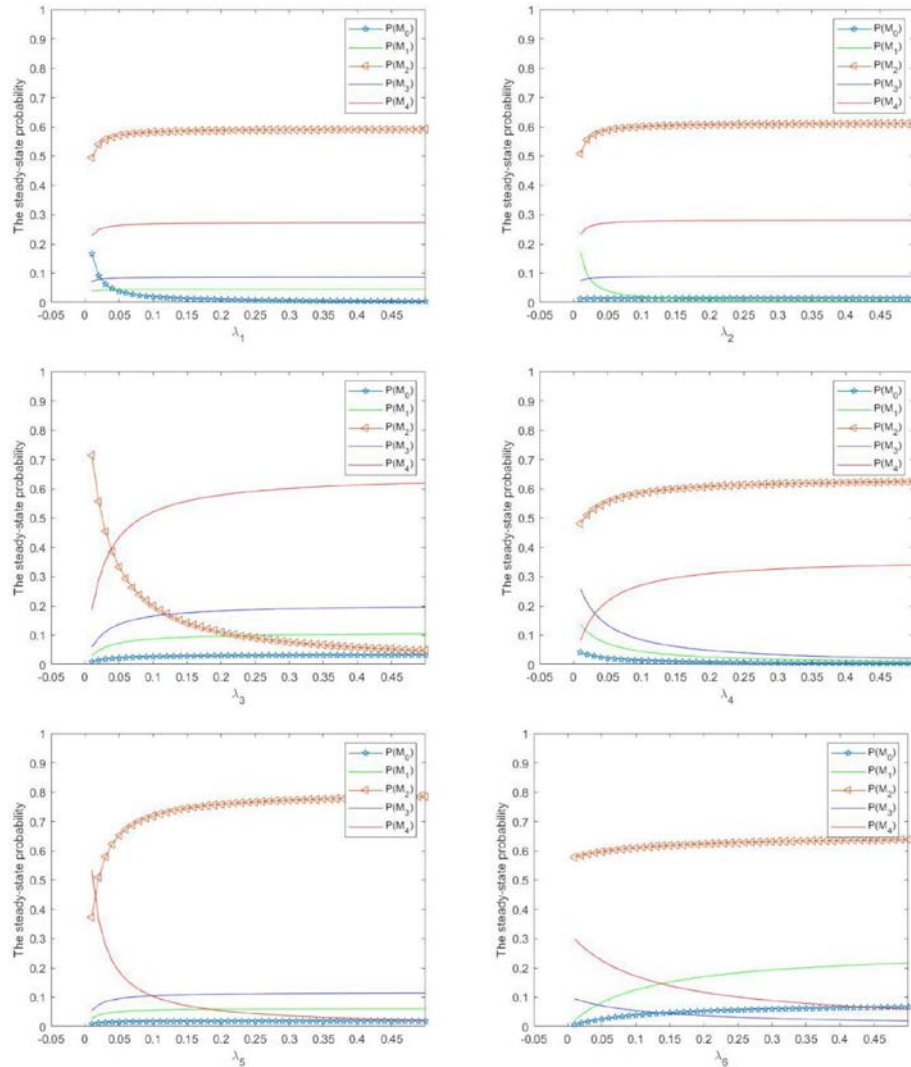


Fig. 18. Dynamic analysis of steady-state probability.

7. Conclusions

This paper proposes a method for refinement and abstraction based on SPZN. Leveraging the advantages of randomness and abstraction inherent in SPZN, this method aims to enhance its descriptive and abstract capabilities. Depending on distinct usage scenarios and requirements, the choice between refinement and abstraction methods is provided. The abstraction method effectively addresses the issue of state explosion in Petri Net by offering a macroscopic perspective of the entire large-scale complex system. This macro-level view significantly reduces the number of places, transitions, and markings, thereby facilitating subsequent analysis and verification through formal verification methods. On the other hand, the refinement method enables a detailed analysis of local aspects of SPZN, contributing to a deeper comprehension of the underlying workflow and structure of large-scale systems. It is worth noting that the SPZN model and its refinement and abstraction methods are primarily tailored to a single agent in an intelligent networked vehicle, lacking an in-depth exploration

of multiple identical agents.

Additionally, this paper introduces a novel approach for determining transition implementation rates. This approach combines interval averaging with the expert system and employs a weighted average method. Compared to alternative methods, this approach is characterized by its simplicity, making it applicable in situations where there is a significant disparity in professional expertise among experts, and their opinions do not fully overlap. However, the presented method has its limitations. Firstly, expert systems rely on expert knowledge, and the accuracy of these systems is contingent upon the knowledge and rules provided by experts. The subjective nature of the weighted average method introduces variability, and different weight assignments may yield disparate results. Furthermore, expert systems lack adaptability to new changes; when the scene changes, the expert system may struggle to provide accurate solutions.

In conclusion, future work will focus on optimizing the SPZN model, refining its abstraction methods, and exploring the applicability of this approach in diverse scenarios and for various usage requirements. Concurrently, research into collaborative control of intelligent networked vehicles will be a pivotal area of investigation.

Acknowledgement

This work was supported by the Chongqing Municipal Natural Science Foundation General Project CSTB2023NSCQ-MSX0441, and the National Scholarship Council for Study Abroad[2020]50, and the National Natural Science Foundation of China Project NSFC 61903053 and 62073051, and the Science and Technology Research Project of the Chongqing Municipal Education Commission of P.R.China under Grants KJZD-K 201900704.

References

- [1] Zhang, Hong, and Xinxin Lu, "Vehicle communication network in intelligent transportation system based on Internet of Things," *Computer Communications*, vol. 160, pp. 799-806, 2020. [Article \(CrossRef Link\)](#)
- [2] Ren, Yuzheng, et al, "Green intelligence networking for connected and autonomous vehicles in smart cities," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1591-1603, 2022. [Article \(CrossRef Link\)](#)
- [3] Wu, Yulei, et al, "A survey of intelligent network slicing management for industrial IoT: Integrated approaches for smart transportation, smart energy, and smart factory," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1175-1211, 2022. [Article \(CrossRef Link\)](#)
- [4] Panigrahy, Saroj Kumar, and Harika Emany, "A survey and tutorial on network optimization for intelligent transport system using the internet of vehicles," *Sensors*, vol. 23, no. 1, pp. 555, 2023. [Article \(CrossRef Link\)](#)
- [5] Sun, Erxin, Zongjuan Chen, and Jianjun Cai, "Cloud control platform of vehicle and road collaborative and its implementation on intelligent networked vehicles," in *Proc. of 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*, IEEE, 2021. [Article \(CrossRef Link\)](#)
- [6] Jia, Yunhui, et al, "Research on Information Security Protection System of Intelligent Networked Vehicles under Computer Artificial Intelligence Technology," in *Proc. of the 7th International Conference on Cyber Security and Information Engineering*, pp. 991-996, 2022. [Article \(CrossRef Link\)](#)
- [7] Chen, Guang, et al, "NeuroIV: Neuromorphic vision meets intelligent vehicle towards safe driving with a new database and baseline evaluations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1171-1183, 2022. [Article \(CrossRef Link\)](#)

- [8] Kolekar, Suresh, et al, “Behavior prediction of traffic actors for intelligent vehicle using artificial intelligence techniques: A Review,” *IEEE Access*, vol. 9, pp. 135034-135058, 2021. [Article \(CrossRef Link\)](#)
- [9] Baruah, Barnana, and Subhasish Dhal, “A security and privacy preserved intelligent vehicle navigation system,” *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 944-959, 2023. [Article \(CrossRef Link\)](#)
- [10] Raman, Ramakrishnan, Nikhil Gupta, and Yogananda Jeppu, “Framework for Formal Verification of Machine Learning Based Complex System-of-Systems,” *Insight*, vol. 26, no. 1, pp. 91-102, 2023. [Article \(CrossRef Link\)](#)
- [11] Almkhour, Mouhamad, et al, “A formal verification approach for composite smart contracts security using FSM,” *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no.1, pp.70-86, 2023. [Article \(CrossRef Link\)](#)
- [12] Mehdipour, Noushin, et al, “Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges,” *Automatica*, vol. 152, pp. 110692, 2023. [Article \(CrossRef Link\)](#)
- [13] Liu, Fei, Monika Heiner, and David Gilbert, “Fuzzy Petri nets for modelling of uncertain biological systems,” *Briefings in bioinformatics*, vol. 21, no. 1, pp. 198-210, 2020. [Article \(CrossRef Link\)](#)
- [14] Assaf, George, Monika Heiner, and Fei Liu, “Coloured fuzzy Petri nets for modelling and analysing membrane systems,” *Biosystems*, vol. 212, pp. 104592, 2022. [Article \(CrossRef Link\)](#)
- [15] Wiśniewski, Remigiusz, Marcin Wojnakowski, and Zhiwu Li, “Design and Verification of Petri-Net-Based Cyber-Physical Systems Oriented toward Implementation in Field-Programmable Gate Arrays—A Case Study Example,” *Energies*, vol. 16, no. 1, pp. 67, 2023. [Article \(CrossRef Link\)](#)
- [16] Kumar, Kuldeep, et al, “Predicting reliability of software in industrial systems using a Petri net based approach: A case study on a safety system used in nuclear power plant,” *Information and Software Technology*, vol. 146, pp. 106895, 2022. [Article \(CrossRef Link\)](#)
- [17] Yin, Xiaohong, Lin Li, and Qiang Liu, “A Study on the Vulnerability Cascade Propagation of Integrated Energy Systems in the Transportation Industry Based on the Petri Network,” *Energies*, vol. 15, no. 12, pp. 4320, 2022. [Article \(CrossRef Link\)](#)
- [18] Huang, Bo, et al, “Scheduling of resource allocation systems with timed Petri nets: A Survey,” *ACM Computing Surveys*, vol. 55, no. 11, pp. 1-27, 2023. [Article \(CrossRef Link\)](#)
- [19] Kaid, Husam, et al, “An improved synthesis method based on ILPP and colored Petri net for liveness enforcing controller of flexible manufacturing systems,” *IEEE Access*, vol. 10, pp. 68570-68581, 2022. [Article \(CrossRef Link\)](#)
- [20] Zhao, Tong, et al, “Formal certification methods for automated vehicle safety assessment,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 232-249, 2023. [Article \(CrossRef Link\)](#)
- [21] Luckcuck, Matt, “Using formal methods for autonomous systems: Five recipes for formal verification,” in *Proc. of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 237, no. 2, pp. 278-292, 2023. [Article \(CrossRef Link\)](#)
- [22] Wang, Feipeng, Diana Filipa Araújo, and Yan-Fu Li, “Reliability assessment of autonomous vehicles based on the safety control structure,” in *Proc. of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 237, no. 2, pp. 389-404, 2023. [Article \(CrossRef Link\)](#)
- [23] Qi, Hongda, et al, “Probabilistic Reachability Prediction of Unbounded Petri Nets: A Machine Learning Method,” *IEEE Transactions on Automation Science and Engineering*, pp. 1-13, 2023. [Article \(CrossRef Link\)](#)
- [24] He, Zhou, et al, “Path planning for automated guided vehicle systems with time constraints using timed Petri nets,” *Measurement and Control*, vol. 53, no. 9-10, pp. 2030-2040, 2020. [Article \(CrossRef Link\)](#)
- [25] Latorre-Biel, Juan I., et al, “Combining simheuristics with Petri nets for solving the stochastic vehicle routing problem with correlated demands,” *Expert Systems with Applications*, vol. 168, no. 15, pp. 114240, 2021. [Article \(CrossRef Link\)](#)

- [26] Yan, Rundong, Sarah J. Dunnett, and Lisa M. Jackson, "Model-based research for aiding decision-making during the design and operation of multi-load automated guided vehicle systems," *Reliability Engineering & System Safety*, vol. 219, pp. 108264, 2022. [Article \(CrossRef Link\)](#)
- [27] Yoshizawa, Takahito, et al, "A survey of security and privacy issues in v2x communication systems," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1-36, 2023. [Article \(CrossRef Link\)](#)
- [28] Chen, Shanzhi, et al, "A vision of C-V2X: Technologies, field testing, and challenges with Chinese development," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3872-3881, 2020. [Article \(CrossRef Link\)](#)
- [29] Gyawali, Sohan, et al, "Challenges and solutions for cellular based V2X communications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 222-255, 2021. [Article \(CrossRef Link\)](#)
- [30] Al-Turjman, Fadi, and Joel Poncha Lemayian, "Intelligence, security, and vehicular sensor networks in internet of things (IoT)-enabled smart-cities: An overview," *Computers & Electrical Engineering*, vol. 87, pp. 106776, 2020. [Article \(CrossRef Link\)](#)
- [31] Fayyad, Jamil, et al, "Deep learning sensor fusion for autonomous vehicle perception and localization: A review," *Sensors*, vol. 20, no. 15, pp. 4220, 2020. [Article \(CrossRef Link\)](#)
- [32] Wang, Zhangu, et al, "A review of vehicle detection techniques for intelligent vehicles," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3811-3831, 2023. [Article \(CrossRef Link\)](#)
- [33] Liu, Yang, et al, "Formal Model and Analysis for the Random Event in the Intelligent Car with Stochastic Petri Nets and Z," *Security and Communication Networks*, vol. 2022, pp. 1-18, 2022. [Article \(CrossRef Link\)](#)
- [34] Castellanos Contreras, Jose Ulises, and Leonardo Rodríguez Urrego, "Technological Developments in Control Models Using Petri Nets for Smart Grids: A Review," *Energies*, vol. 16, no. 8, pp. 3541, 2023. [Article \(CrossRef Link\)](#)
- [35] Liu, GaiYun, and Kamel Barkaoui, "A survey of siphons in Petri nets," *Information Sciences*, vol. 363, pp. 198-220, 2016. [Article \(CrossRef Link\)](#)
- [36] Giua, Alessandro, and Manuel Silva, "Petri nets and automatic control: A historical perspective," *Annual Reviews in Control*, vol. 45, pp. 223-239, 2018. [Article \(CrossRef Link\)](#)
- [37] Taleb-Berrouane, Mohammed, Faisal Khan, and Paul Amyotte, "Bayesian Stochastic Petri Nets (BSPN)-A new modelling tool for dynamic safety and reliability analysis," *Reliability Engineering & System Safety*, vol. 193, pp. 106587, 2020. [Article \(CrossRef Link\)](#)
- [38] Molloy, "Performance analysis using stochastic Petri nets," *IEEE Transactions on computers*, vol. C-3, no. 9, pp. 913-917, 1982. [Article \(CrossRef Link\)](#)
- [39] Liu, Shuanglei, et al, "Modeling and performance analysis of gas leakage emergency disposal process in gas transmission station based on Stochastic Petri nets," *Reliability Engineering & System Safety*, vol. 226, pp. 108708, 2022. [Article \(CrossRef Link\)](#)



Yang Liu was born in 1984. She received the Ph.D. degree in computer software and theory from the University Electronic Science and Technology of China, Chengdu, China, in 2013. Her research interests include formal verification, information safety, data processing, and fault-tolerant control.



Yingqi Fan was born in 1999, graduated from Chongqing Jiaotong University with a bachelor's degree in computer science and technology in 2021. Currently studying for a master's degree in the School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing. His research interests include formal verification and information safety.



Ling Zhao received the B.S., M.S., and Ph.D. degrees in mechanical and electronic engineering from Chongqing University, in 2002, 2005, and 2010, respectively. Since 2020, she has been a Professor with the College of Information Science and Engineering, Chongqing Jiaotong University, Chongqing, China. Her research interests include fault diagnosis and fault-tolerant control of dynamic systems, fault diagnosis of complex systems, and reliability engineering.



Bo Mi was born in 1982. He received the Ph.D. degree in computer system architecture from Chongqing University, China, in June 2009. Since 2011, he has been an Associate Professor with the College of Information Science and Engineering, Chongqing Jiaotong University, China. His current research interests include intelligent transportation, vehicular ad-hoc networks, and cryptography.