Space-time Constraint Resources Modeling and Safety Verification Method for Automated Vehicles

Yi Zhu[1,2], Xiaoying Chen[1,*], and Yu Zhao[1]
[1]School of Computer Science and Technology, Jiangsu Normal University, XuZhou, China
[2]Key Laboratory of Safety-Critical Software, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
zhuy@jsnu.edu.cn, cxy@jsnu.edu.cn, zhaoyu@jsnu.edu.cn
*corresponding author

***Abstract*—Automated vehicle combines physics and computation on the basis of environment perception. It can realize intelligent interaction with the environment. Automated vehicle is a typical CPS. However, the continuous changes of driving physical space bring certain challenges to the safety of CPS resources. Therefore, how to solve this kind of CPS resource safety problems caused by space and time changes becomes the key. We propose a space-time constraint resource modeling and safety verification method for automated vehicle to solve this problem. Firstly, the physical topology model is proposed to model the physical topology space of CPS, which is able to describe the topology space. Secondly, the Resource-Space Time Communicating Sequential Process (RS-TCSP) is proposed by extending the resource vector on the basis of Time Communicating Sequential Process(TCSP) to describe the resources in CPS topology. Thirdly, the physical topology model and RS-TCSP are mapped to bigraphs and bigraphs reactive system, respectively. The safety of CPS resources is verified by BigMC, the verification tool of bigraphs, and the counterexample path is modified. Finally, a driving scene is given to verify the effectiveness of the proposes method.**

Keywords- cyber physical system; formal verification; process algebra; space-time constraint; resource safety

I. INTRODUCTION

CPS can be summed up as computation, communication and control. CPS is a controllable, credible and extensible networked physical device system that deeply integrates computing, communication and control capabilities on the basis of environmental awareness. Due to its human-computer interaction and the driving environment, CPS produces special resources: space-time constraint resources, such as: a parking space in a parking lot, a section of rail in a line, a data of the system, a message from a mobile device, etc. This kind of special resource is affected by time and physical topology space, and its safety affects the safety of automated vehicles.

In 2011, a bullet train collision occurred in Wenzhou, causing huge losses. The reason is that under the action of lightning strike, there are no vehicles occupying the area under the jurisdiction of the train control center of Wenzhou South Station. As a result, the train control center still displays the status of no vehicle occupied for control output when the actual vehicles occupy the area in the subsequent period. Therefore, the signal machine of the train control center is wrongly displayed as green, which leads to retail collision [1]. The unsafe occupation of railway resources has produced serious consequences. The safety of CPS resources may cause serious consequences, especially in some safety-critical CPS, such as the train control systems, automated vehicles, etc. As one of the influences of the safety of the CPS, the resource safety has been a hot issue in the research of CPS safety. As time and topology change, resource safety will be threatened and even cause serious consequences. Therefore, a safety verification method is urgently needed to ensure the safety of CPS resources. Therefore, how to verify the safety of space-time constraint resources to ensure the safety of CPS under the changes of time and topology space is the current challenge.

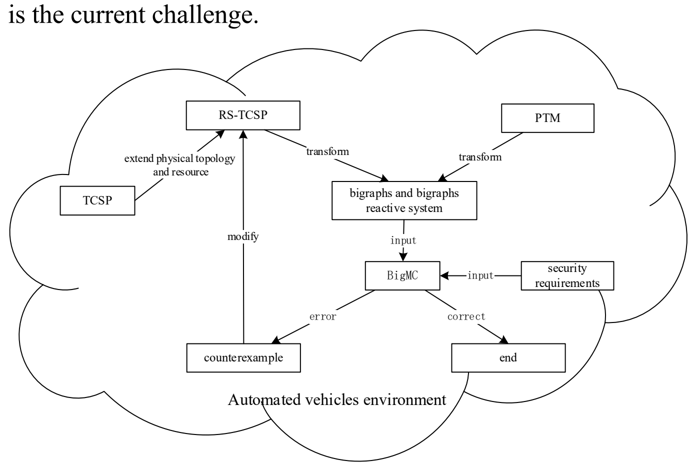


Fig 1: Technology roadmap.

In recent years, many achievements have been made in CPS safety verification. Reference [2] studied the real-time impact of environmental changes on system parameters. Reference [3] studied the impact of time and space consistency on CPS safety. These studies focus on non-functional attributes such as time, which is used as a resource to verify the system. Reference [4] proposes a CPS task-virtual resource scheduling mechanism based on intelligent planning. However, it implements scheduling of virtual resources without taking time and space into account. Reference [5-7] managed the energy in CPS to realize the energy consumption estimation in CPS. Reference [8] proposes the impact of topology space on CPS safety. However, it does not consider the impact of time on CPS safety. Reference [9] proposes a spatio-temporal access

2767-6684/22/$31.00 ©2022 IEEE
DOI 10.1109/DSA56465.2022.00112

control model of online social networks and its visual verification. Compared with the above work, we model and verify the space-time constraint resources to ensure the safety.

Communicating Sequential Process (CSP) is a formal method established in 1978 by Hoare [10] suitable for the specification and design of distributed concurrent software. In 1986, Oxford's Reed and Roscoe extended the CSP in real time and proposes Timed Communicating Sequential Process (TCSP) [11]. Process algebra is a formal method to solve the communication of concurrent systems. It can describe the problems of concurrency, synchronization, and asynchrony of events in CPS. However, the description ability for space of TCSP is limited, especially the physical topology space. In addition, TCSP also lacks the description ability of resources. Therefore, it is necessary to extend the description ability of the physical topology space and the resource for TCSP. So that the TCSP can describe the physical topology space of the CPS and the resources in cyber physical space, and then verify the safety of the resources corresponding to the space and time.

Based on the above analysis, we propose space-time constraint resources modeling and safety verification method for automated vehicles (as shown in Fig1). This method firstly models the physical environment in automated vehicles environment and proposes Physical Topology Model(PTM). Secondly, extend physical topology space and resources on TCSP and propose the RS-TCSP. Thirdly, the PTM and RS-TCSP are transformed into bigraphs and Bigraphs Reactive System(BRS) through the model transformation. The bigraphs tool BigMC is used to verify it. Counterexamples are used to modify RS-TCSP until the proposes safety requirements are satisfied. Finally, an example of a driving system scene is used to illustrate the effectiveness of the method in verifying the safety of CPS resources.

II. Background Knowledge

A. Bigraphs

Bigraphs is composed of a place graph and a link graph. The place graph is a forest with the number of regions as the root node, which can represent the nesting relationship between each node. The link graph is a hyper graph composed of the same set of nodes in the place graph and a set of edges. The connecting any number of nodes is used to represent the connection relationship between nodes. The place graph and the link graph are different results obtained from the observation of the same bigraphs. The related concepts are introduced according to Figure 2. Figure 2(a) is bigraphs *F*, Figure 2(b) and (c) are the place graph and link graph of the bigraphs *F* respectively.

There are two regions in Figure 2(a), which are represented by dashed boxes as 0, 1. V_0, V_1, and V_2 represent nodes. There is a nested relationship between V_1 and V_2, which is determined by the relationship between the modeling objects.

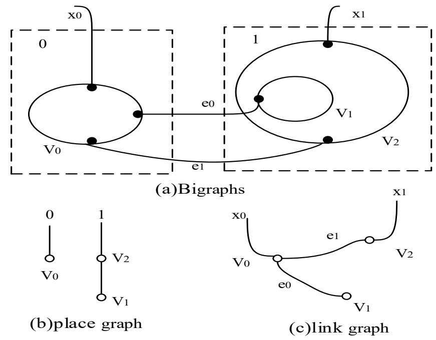


Fig 2: The anatomy of bigraphs.

The black dots in the figure are ports, and the ports can be connected by edges. Where e_0 and e_1 are closed links, x_0 and x_1 are open links.

B. Term language

Bigraphs describes the change of the physical position intuitively, but the change is hard for computers to understand. Milner et al. proposes an algebraic system to describe the bigraphs and the bigraphs reaction system. Table 1 shows part of the algebraic representation of the bigraphs and BRS [12].

Table 1. The bigraph symbol representation of PTM.

Term language representation	Meaning
R||T	Concatenation of roots
R|T	Concatenation of nodes
R∘T	Composition
R. T	Nesting
/x. R	R with outer name x replaced by an edge
x/y	Connection inner names y to outer name x

C. Bigraphs reactive system (BRS)

BRS's form can be expressed as *redex→reactum*. It reconstructs itself by defining reaction rules. Before the arrow is redex, after the arrow is the reactum. The bigraphs of redex are transformed into the bigraphs of reactum according to the reaction rules. As shown in Figure 3 is a reaction rule. The left and right sides are respectively redex and reactum. The reaction rule is expressed as: $C[x_0].(P) | D[x_1] \to C[x_0].(D[x_1] | P)$. It means that the object *D* with the connection x_1 enters the object *C* with the connection x_0. In the process of change, the connection relationship remains unchanged. If the bigraphs or part of the bigraphs matches the redex, the reactum will be replaced by reactum after the reaction rule.

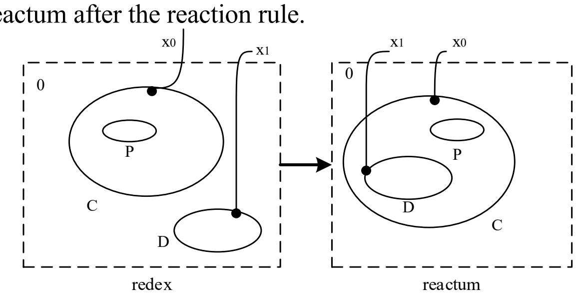


Fig3: Reaction rule.

There are many tools to support bigraphs and BRS, such as BigRed [13] and BigMC [14] etc. BigMC is a model checking tool that runs on the BRS. BRS is a formal tool developed by Robin Milner etc., emphasizing the orthogonality of locality and connectivity.

III. RELATED WORK

In recent years, the modeling and verification of CPS safety have made great progress. Reference [15] proposes a consistency verification method, which aimed to verify that the physical characteristics and time of this process will not cause conflicts. References [16-22] modeled how to safely interact between the various parts of the CPS under time constraints. Most of these traditional CPS modeling and verification are limited to the analysis in the time domain. Less consideration is given to the impact of physical topology changes on CPS. So some safety issues of CPS space-time resources exist.

Reference [23] proposes a new space-time language for CPS to support the unified modeling of the space-time property of CPS. And the language explained the topology space and natural numbers based on time. This part of work considers both time and space, conducting unified research on time and space. Reference [24] proposes a methodology and technical framework that supported the modeling of the evolving cyber physical space. The space of CPS is not only in the cyber space, but also in the physical space. Physical space is also an important factor affecting CPS safety.

For the study of physical space, the research results have been abundant in recent years. Reference [25] used BRS to model the topology of cyber space, physical space and its dynamics. Use this model to perform speculative threats through model checking analysis. It inferred the consequences of the evolution of topology deployment to satisfy the safety requirements. Reference [26] proposes a method for modeling the evolution of spatial scene snapshots and verifying the space-time model. Bigraphs were introduced into the topology space to define a novel topology map. It was used to study the expressibility and verifiability of modeling and analysis of space-time behavior. Reference [27] proposes a topology-aware network physical access control model (TA-CPAC). It can ensure the safety of the network and the physical world at the same time by dynamically adjusting the allocation of permissions. However, the focus of this research is on the formulation of access control policies. It takes little consideration of the resources in CPS and does not focus on the time. Reference [28] extends the time property on RBAC to study the access control model under the influence of time. Reference [29] studies RBAC under temporal and spatial constraints.

Automatic driving is the hot spot in recent years. Reference [30] proposes a new automatic annotation method to analyze road semantics, which treats the prior trajectory of vehicles as a multi-dimensional sequence and extends the traditional time series method to the spatial domain to process the data. Reference [31] minimizes the average travel time of all vehicles in the network relative to their respective travel deadlines to improve traffic throughput.A new approach to energy saving of intelligent transportation system (ITS) by using the delay constraint framework is proposed in reference[32].

These access control only study time and space factors, but does not focus on the resources corresponding to the space-time in cyber physical space. It cannot guarantee the safety of the space-time resources of the CPS in cyber physical space.

This article is a continuation of the existing work. In this paper, the physical topology space and resources are added to verify and modify the resources in the CPS. With changes in physical topology space and time, so as to ensure the safety of space-time resources for automated vehicles and realize trusted CPS. The method is an effective supplement to existing work.

IV. PTM AND RS-TCSP

A. PTM

If the building is a root node, and the rooms are taken as its child nodes, a tree describing the physical topology is formed. For each physical location domain set **POS**:=$\{p_1, p_2. . . , p_m \mid m \in N^+\}$, the nodes have a certain containment and proximity relationship. Cyber location domain set **CPOS**:=$\{cp_1, cp_2. . . , cp_n \mid n \in N^+\}$.

Definition 1. The inclusion relationship of the physical location domain.

If the location domain is in the hierarchical structure, and the node p_i is the parent node of the node p_j , then p_i includes p_j. It is denoted as $p_i(p_j)$.

Definition 2. The inclusion relationship between the physical location domain and the cyber location domain.

If the cyber domain $\{cp_k \mid \mathrm{cp}_k \in CPOS,\ k \in N^+,\ k \le \mathrm{m}\}$ is in the physical domain $\{p_r \mid \mathrm{p}_r \in POS,\ \mathrm{r} \in N^+,\ \mathrm{r} \le \mathrm{n}\}$. It is expressed as $p_r(cp_k)$.

B. RS-TCSP syntax

Definition 3. The RS-TCSP can be defined as:

$$P ::= STOP \mid SKIP \mid WAIT\ t \mid a \xrightarrow{(r,object)} P \mid P;Q \mid P \Box Q \mid P \sqcap Q \mid P \overset{d}{\triangleright} Q \mid f(P) \mid P \setminus A \mid P\ {}_A\|_B\ Q \mid P \mid\mid\mid Q \mid \mu X \cdot f(X) \mid Con >> P(Con ::= SPACE \mid TIME \mid RES \mid Con1 \wedge Con2 \mid Con1 \vee Con2 \mid true) \mid P \Box Fin\_Con(Fin\_Con ::= SPACE \mid TIME \mid RES \mid Con1 \wedge Con2 \mid Con1 \vee Con2 \mid false) \vdash Q$$

STOP is a process which will never engage in external communication, and it makes the process terminate;

SKIP is a process which does nothing except terminate, and is ready to terminate immediately;

WAIT t is a delay for skip. It does nothing, but is ready to terminate successfully after *t* time units;

$a \xrightarrow{(r,object)} P$ is the prefix operation, which means that the process *P* is executed after the event *a* is executed on the *object*. The resource vector ***r*** is changed, ***r***:=<PTM, (t, t_{wait}), *res*>. t and t_{wait} are the execution time and waiting time respectively. *res* represents the resource under the physical topology of PTM and time (t, t_{wait}). *res* can be empty. When *res* is empty, it can be omitted;

In the process $P;Q$, control is passed from process P to process Q if and when P performs the termination event. This event is not visible to the environment, and occurs as soon as P is ready to perform it. The sequential composition operator transfers control upon termination;

$P \square Q$ is an external choice between process P and Q. If the environment is prepared to cooperate with P but not Q, then the choice is resolved in favor of P;

$P \sqcap Q$ is an internal choice between P and Q, and the outcome of this choice is nondeterministic;

$P \overset{d}{\triangleright} Q$ represents timeout. If no communication occurs between the two processes within d, it is considered timeout and control is passed from P to Q;

$P \backslash A$ indicates that any events belonging to A in process P are not displayed;

The relabeled process $f(P)$ has a similar control structure to P, with observable events renamed according to function f;

In the hybrid parallel program $P_A \|_B Q$, components P and Q must synchronize according to events from set $A \cap B$, and they interleave on all other events;

$\mu X \cdot f(X)$: X is a process variable, $A = \alpha X$, a recursively defined process must immediately unwind before it is able to perform any visible action;

$$Con >> P(Con ::= SPACE \mid TIME \mid RES \mid Con1 \wedge Con2 \mid Con1 \vee Con2 \mid true)$$

It is called the space-time resource condition execution operator. When Con is satisfied, P is executed. Con includes three parts: the physical topology model $SPACE$, the time model $TIME$ and resource model RES. $SPACE=F_{judge}(F_{ptp}(x, y, z), l)$ is a physical location domain judgment function. It is a point-to-domain mapping function. $F_{ptp}(x, y, z)= l$ inputs points (x, y, z) and outputs the physical location domain of the object. $F_{judge}(F_{ptp}(x, y, z), l)$ judges whether the area of the current three-dimensional coordinate position is l. If the domain is l, it returns *true*. Otherwise, it returns *false*. The point (x, y, z) of the object can be mapped to the physical topology space area of the CPS. When the condition does not require space and time constraints, the condition is *true* by default. When the conditions are satisfied at the same time, use "$\wedge$". $TIME$ is a predicate verb, $TIME= (t_i, t_j)$, it represents the time period containing t_i, t_j. $TIME$ judges whether the current time is in (t_i, t_j). If the current time $t_{current} \in (t_i,t_j)$, $TIME$:=*true*. Otherwise, $TIME$:=*false*. RES is expressed as $res \equiv n$. res is the resource condition for executing the process, where n is a real number, $\equiv \in \{\geq,>,=,\leq,<\}$;

$$P \triangle Fin\_Con(Fin\_Con ::= SPACE \mid TIME \mid RES \mid Con1 \wedge Con2 \mid Con1 \vee Con2 \mid false) \vdash Q$$

$SPACE$, $TIME$ and RES are the same as the models in the space-time resource condition execution operator. The default is *false* and can be omitted. If the condition Fin_Con is satisfied, the interrupt on P can be executed and then execute process Q. Use "$\wedge$" when the conditions are satisfied at the same time, and use "$\vee$" if at least one of the conditions is satisfied;

The basic operation of TCSP $a?x$ means that the process receives the input of x through the channel a.

C. Algorithm for model checking

(1) Time verification

First, verify that resource safety is affected by time safety requirements. Time affects the safety of resources. If resources are used outside the allowed time range, resource safety may be compromised. Next, we verify the time requirement of resource safety through an algorithm for time property verification.

Algorithm 1 Algorithm for time property verification

```
abnormal:=∅; cur_path={N0}; totalt:=0; curr_t:=currentime
repeat
  ln:=last node in cur_path; //get the last node from current path
  if successor nodes of last node have been visited//delete visited nodes
    then delete last node of cur_path;
  else
    begin
      if time constraint(ti,tj) exists,totalt<ti or totalt>tj then result:=false;
//When the time value from the source node to the current node bn is not in
the (ti,tj), the result value is false
        then abnormal =abnormal∪{en}; //When the time value from
the source node to the current node bn is not in the (ti, tj) time period, write
down the abnormal node
      cur_path=cur_path∪{bn};
      end
  until cur_path=∅;
  if abnormal=∅ then
    return true;
  else return false;
```

The verification of time is the $TIME$ model in the verification condition. Algorithm for time property verification traverses the state space graph through a depth-first algorithm. Check whether the current time is within the time period (t_i, t_j). If the current time is satisfied. The time requirement returns *true*, otherwise it returns *false*.

(2) Get Deadlock

If the system is deadlocked due to time safety violations, the time transition system until each node meets the time safety requirements. For the modification of the system, the algorithm for finding the deadlock *getdeadlock()* is used to locate the nodes that violate the time safety requirements, and then the deadlock modification algorithm is used to modify the nodes.

Algorithm for finding the deadlock uses a depth-first algorithm to traverse the state space graph to find deadlock nodes. In the graph, deadlock nodes are nodes that do not contain child nodes. The node cannot continue to execute later. So it is necessary to locate the deadlock node to facilitate subsequent modification operations to the deadlock.

(3) Deadlock modification

The modification to the deadlock node includes three operations: adding an edge *ed* to the deadlock node; delete the node, that is, the scheme of this node is not selected; add an error handling node *en* and edge *ed* to fix the deadlock.

Algorithm 2 Algorithm for finding the deadlock *getdeadlock*()

```
deadlock:= ∅ ; cur_path={N0}
repeat
  ln:=last node in cur_path; //get the last node from the current path
  if successor nodes of last node have been visited//delete the visited nodes
    then delete last node of cur_path;
  else
    begin
      bn:=take a unvisited successor node of ln ; //take a child node bn that is not accessed by ln
      if bn=null;//The unreachable node does not have a child node and deadlock occurs
        deadlock=deadlock∪{ln}; cur_path=cur_path∪{bn};
      else
        cur_path=cur_path∪{bn};
    end
  until cur_path=∅ ;
  if deadlock=∅ then
    return true;
  else return false;
```

Algorithm 3 Algorithm for modifying the deadlock

```
G= current state transition graph; //G is the current state transition graph
deadlock=getdeadlock().deadlock; //get deadlock nodes from getdeadlock()
repeat
begin
  dn:=a node in deadlock; //get a node from current deadlock
  a= choose a to deal with the deadlock of dn; //choose the way to handle the deadlock
  switch(a):
  case 0:add an edge ed in graph G,G = G∪{ed}; break;
  case 1:delete the deadnode dn from graph G,G = G/{dn} ; break;
  case 2:add an error-handling node en and an edge ed in graph G,G = G∪{en}∪{ed}; break;
  delete node dn from deadlock; //the processed node is deleted from deadlock
end
until deadlock=∅ ; //handle all deadlock nodes
return G;
```

The RS-TCSP model meets the time safety requirements through the time safety requirement. Then model transformation will be carried out to verify the physical topology safety requirements. DFS is used in Algorithm 1-3, the time and space complexity is O(n).

V. Resource Safety Verification in the Physical Topology

In order to verify the physical topology safety requirements of CPS system resources, the RS-TCSP was converted into bigraphs and bigraphs reaction system, and the bigraphs tool BigMC was used for model detection to verify the safety of resources corresponding to space and time in the physical topology environment.

A. Mapping rules from PTM to bigraphs

The transformation from PTM to bigraphs is as follows:
(1) The physical locations in **POS** and **CPOS** are transformed into nodes **V**;
(2) The inclusion relationship $p_i(p_j)$ and $p_r(cp_k)$ are transformed into the nesting of nodes;
(3) The communication channels are transformed into related connection links;
(4) The changes of physical topology space and space-time constraint resource caused by $a \xrightarrow{(r,object)} P$ events in RS-TCSP are transformed into bigraphs reactions;
(5) The *SPACE* and *RES* of two extended operations are transformed into the redex of the bigraphs reaction rules.
(6) According to the specific process events, the event *a* in the event set **A** is transformed into the specific bigraphs reaction rule according to the change of its physical topology position, so as to realize the mapping from RS-TCSP to the bigraphs reaction rules.

Transformation rule 1. PTM to bigraphs reaction rules.

V: ***POS***、***CPOS*** $\Rightarrow$ *V*;//***POS*** location resource set and ***CPOS*** cyber resource set are transformed into node set **V**
ctrl: ***V→K***;//Nodes to controls mapping, ***K*** can be all entities in the CPS environment
prnt: $p_i(p_j) \Rightarrow p_j \to p_i$; //The inclusion relationship of the physical location is transformed to the nesting relationship between nodes
$p_r(cp_k) \Rightarrow cp_k \to p_r$;///The inclusion relationship between the physical location and the cyber location domain is transformed into a nested relationship between nodes
link: *channel* $\Rightarrow$ *link*; //*link* is the connection relationship of the communication *channel* between the processes
E:*link* connected edge set
m = *r*;//The number of sites in the actual CPS scene is *r*
n = *k*;//The number of regions in the actual CPS scene is *k*
X is the internal name of the CPS physical ports
Y is the external name of the CPS physical ports

The transformed bigraphs are symbolized as:

Table 2. The bigraph symbol representation of PTM.

The resource type	Node characteristics	Graphical representation
subject resource	active	
position resource	active	
cyber resource	active	
port	active	

VI. Case Study

Driving scenes and smart parking lots are both typical CPS. The following figure shows a physical deployment graph of a local city driving scene.

The figure 4 is a partial deployment structure graph of a city. The deployment structure graph shows the spatial structure of the city. The gray area is the road. There are three roads: *road1*, *road2* and *road3*. A *crosswalk* at the entrance of the school on *road1*. Blank areas are buildings in the city. For the convenience of description, this article lists four regional resources in local areas: *school*, shopping mall

(*mall*), parking lot (*parklot*) and *construction*. There are two signs on the road: *leftsign* and *parksign*.

In this example, the simplified parking space resources in the *parklot* are 6 parking spots: (*spot1*, *spot2*, *spot3*, *spot4*, *spot5*, *spot6*). The Intelligent parking management system (*IPMS*) is deployed on the *server* in the *guardroom* of the *parklot*. There is a record resource on the *IPMS*, so the *driver* can only enter the *guardroom* when the *guard* is present. It is not allowed to enter the *guardroom* alone to ensure the safety of the *record* resource. The area in the parking lot is the *mainarea*, and the parking lot is open from 5 to 20.

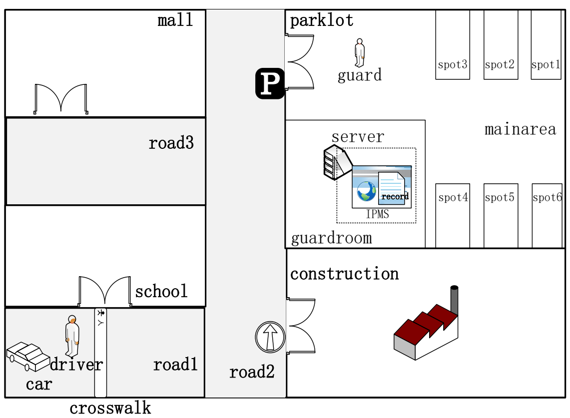


Fig 4: Local physical deployment of the city.

The working principle of IPMS is shown in the figure below. It is mainly divided into three modules: Data Collection Model (*DCM*), Design Model (*DM*) and Enforcement Model (*EM*). *DCM* includes some cameras, radars and other sensors data collection and related data preprocessing. Subsequently, the system sends the processed data to the *DM*. The *DM* through a series of data storage, data calculation and final decision. The result of the decision is input to the *EM* for the execution of related actions.

Arriving at the *parklot*, if the parking lot is during working hours and there are parking spots, the parking lot opens the gate, and the driver logs in to the IPMS to obtain relevant voice guidance and other prompts.

The existing *car* can go through the school and then through the *Road2* to finally arrive at the *parklot*, or through the *Road1* and then through the *Road2* to the *Road3* to the *mall* shopping and then to the *parklot*. It depends on the driver's goal choice. When the car performs these two goals, the two modules of the car need to work together: Speed Management Model (*SMM*) and Direction Management Model (*DMM*). *SMM* is connected to four units: Start-Stop Unit (*SSU*), Speed Notification Unit (*SNU*), Acceleration Unit (*AU*) and Brake Unit (*BU*). *DMM* is connected to two units: Steering Wheel (*SW*) and Direction Notification Unit (*DNU*).

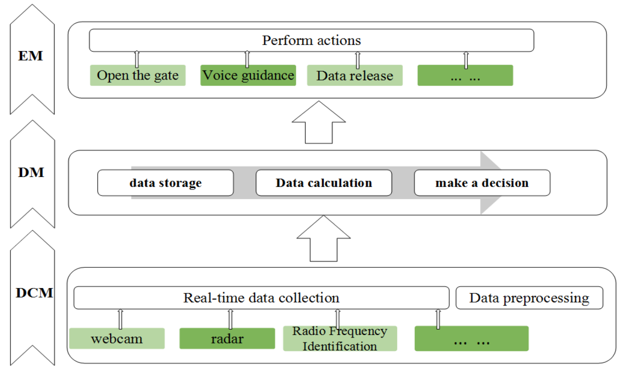


Fig 5: Intelligent parking management system architecture.

Reference [33] summarizes human information processing into a four-stage process: information acquisition, information analysis, decision-making and action selection, and action implementation. We simplify the process into three modules: information acquisition (*GET*), Goal (*G*) and Execution (*EXE*). *G* is the process of obtaining the final goal through human processing. In the current scene, the *driver* wants to park the *car* in *parklot* and enter *Guardroom* to read the *record*. Select *road1→road2→parklot→Guardroom* from the current location.

1) *Scene modeling*

(1)We use PTM and RS-TCSP to model the scene. First, the PTM is obtained from the local physical deployment graph.

POS:={p_{mall}, p_{school}, $p_{crosswalk}$, $p_{parklot}$, $p_{construction}$, p_{road1}, p_{road2}, p_{road3}, *server*, *leftsign*, *parksign*, *car*, *driver*, *guard*, *mainarea*, *spot*, *guardroom*}

CPOS:={*IPMS*, *DM*, *EM*, *DCM*, *record*, *GET*, *G*, *EXE*, *SMM*, *SSU*, *SNU*, *AU*, *BU*, *DMM*, SW, *DMU*}

The physical deployment relationship is:

PL(p_{mall}, p_{school}, $p_{parklot}$ (*mainarea*(*spot*, *guard*, *guardroom*(*server*))), $p_{construction}$, p_{road1} ($p_{crosswalk}$, *car*, *driver*), p_{road2} (*leftsign*, *parksign*), p_{road3})

Fig 6: Physical deployment relationship.

The cyber deployment relationship is:

IPMS(*DM*, *EM*, *DCM*, *record*)
driver(*GET*, *G*, *EXE*)
car(*SMM*, *SSU*, *SNU*, *AU*, *BU*, *DMM*, *SW*, *DMU*)

Fig 7: Cyber deployment relationship.

The set of communication channels for this scenario:

channel:={*pk, lt, cw, bu, au, sn, ss, name, st, sn, ac, br, dm, log, read, cname, dm, sw, work, logg, gname, sp1, sp2, sp3, sp4, sp5, sp6, re, login, city*}

Event set ***A***:={*in*, *out*, *accelerate*, *brake*, *enter*, *exit*, *login*, *loginout*, *turn*, *read*}

The initial process definitions of *DRIVER*, *CAR* and *IPMS* are as follow:

$DRIVER_{initial}$=*GET*||*G*||*EXE*|*name?→STOP*||*st?→STOP*||*sn?→STOP*||*ac?→STOP*||*br?→STOP*||*dm?→STOP*||*sn?→STOP*||*ac?→STOP*||*br?→STOP*||*dm?→STOP*|*log?→STOP*||*read?→STOP*||*get1!→STOP*||*get2!→STOP*||*exe2?→STOP*
GET=get1?→STOP||*get2?→STOP*
G=get2!→STOP||*exe1!→STOP*
EXE=exe1?→STOP||*exe2!→STOP*

Fig 8: Initial DRIVER process model.

$DRIVER_{initial}$ is the initial model in the current physical topology environment, and represents the concurrency of multiple processes. The initial model in the current physical topology, represented as the concurrency of multiple processes. It contains the interaction between the three modules of *DRIVER GET*, *G* and *EXE*. For example, if the *G* module generates a target and sends it to the *EXE* through the *exe1* channel, the *G* process contains *exe1!→STOP* concurrency, EXE contains *exe1?→STOP* concurrency to indicate the sending and receiving of the *goal*.

CAR=SMM||SSU||SNU||AU||BU||DMM||DMU||SW
SMM=ss2?→STOP||su2?→STOP||au2?→STOP||bu2?→STOP
SSU=ss1?→STOP||ss2!→STOP
SNU=su1?→STOP||su2!→STOP
AU=au1?→STOP||au2!→STOP
BU=bu1?→STOP|bu2!→STOP
DMM=sw2?→STOP||dmu2?→STOP
DMU=dmu1?→STOP||dmu2!→STOP
SW=sw1?→STOP|sw2!→STOP

Fig 9: CAR process model.

The *CAR* process contains the interaction of two modules and four units of *car*. So the *CAR* process is the concurrent process of these units. At the same time, the interaction between each module and unit is the concurrency of the process composed of the transceiver operation of the corresponding channel.

IPMS=DM||EM||DCM||record!→STOP
DCM=dm1!→STOP
DM=dm1?→STOP||dm2!→STOP
EM=dm2?→STOP||em!→STOP

Fig 10: IPMS process model.

IPMS process is the *DM, EM, DCM* and *record* send action related process concurrency. *DCM* sends the collected data through channel *dm1*, and *DM* receives the data sent by *DCM* through channel *dm1* for decision-making. Similarly, the *DM* sends decision data through channel *dm2*, and the *EM* receives decision data through channel *dm2* and sends execution commands through the *EM* channel.

DR_ENTER_ROAD2=

$$\mu X \bullet (in \xrightarrow{(PTM,(0.3,0.2),car)} accelerate \xrightarrow{(PTM,(0.2,0.2),car)} brake \xrightarrow{(PTM,(0.1,0.1),car)} enter \xrightarrow{(PTM,(0.2,0.3),road2)} X)$$

DRIVER=

$$\mu X \bullet (DR\_ENTER\_ROAD2;(5,20) \wedge (1 \le spots \le 6) >> enter \xrightarrow{(PTM,(0.2,0.3),parklot)} (F_{judge}(F_{ptp}(x,y,z),\ parklot) \wedge (5,20)) >> \log in \xrightarrow{(PTM,(0.2,0),IPMS)} enter \xrightarrow{(PTM,(0.1,0),guardroom)} read \xrightarrow{(PTM,(0.1,0),record)} X)$$

Fig 11: DRIVER process model.

In this scenario, *driver* enters *ROAD2* first, then *parklot*, and then *guardroom* to read *record*. *DR_ENTER_ROAD2* process is executed by a series of actions,*enter car→accelerate car →brake car→enter road2*. After the *DRIVER* process is *DR_ENTER_ROAD2*, *enter parklot→login IPMS→enter guardroom→read record.*

$GUARD_{initial}$*=logg!→STOP||gname?→STOP*

$$GUARD= \mu X \bullet (enter \xrightarrow{(PIM,(0.3,0.2),guardroom)} \log in \xrightarrow{(PIM,(0.2,0.2),IPMS)} exit \xrightarrow{(PIM,(0.1,0.1),guardroom)} X)$$

Fig 12: GUARD process model.

The initial *GUARD* process is the concurrency of the two processes that input the guard's name data through the channel *gname* and output login information through the *logg* channel. In the current scenario, *guard enter guardroom→login IPMS→exit guardroom*

MALL=work1!→STOP
PARKLOT=work2!→STOP
SPOT=sp1!→STOP||sp2!→STOP||sp3!→STOP||
sp4!→STOP||sp5!→STOP||sp6!→STOP
CROSSWALK=cw!→STOP
PARKSIGN=pk!→STOP
LEFTSIGN=lt!→STOP

Fig 13: Other process models.

MALL and *PARKLOT* output working data through channels, respectively. A *SPOT* process is a concurrency of six parking spaces sending data over a channel whether they are being used or not. In the same way, the *CROSSWALK*, *PARKSIGN*, and *LEFTSIGN* processes also send the used data through the channel.

$D1=DRIVER_{initial}||DRIVER$
$G1=GUARD_{initial}||GUARD$

$$ADS=D1 \underset{A}{||} G1 \underset{A}{||} CAR \underset{A}{||} IPMS \underset{A}{||} MALL \underset{A}{||} PARKLOT \underset{A}{||} SPOT \underset{A}{||} CROSSWALK \underset{A}{||} PARKSIGN \underset{A}{||} LEFTSIGN$$

Fig 14: ADS process model.

(2) Next, transform the model according to the transformation rule 1 in Section 5.1, and the transformation result is as follows:

Node set in the process **V**:={*mall*, *school*, *crosswalk*, *construction*, *parklot*, *road1*, *road2*, *road3*, *server*, *leftsign*, *parksign*, *car*, *driver*, *guard*, *mainarea*, *guardroom*, *IPMS*, *DM*, *EM*, *DCM*, *record*, *GET*, *G*, *EXE*, *SMM*, *SSU*, *SNU*, *AU*, *BU*, *DMM*, *SW*, *DMU*}

The containment relationship between the physical location domain and the containment relationship between the physical location domain and the cyber location domain are transformed into the nesting relationship. For the ***channel*** in the above model, it is transformed into the *port* in the bigraphs. The sending and receiving process of the same channel is mapped as the connection in the topology space. Such as *dm1!→STOP* of *DCM* and *dm1?→STOP* of *DM* is the sending and receiving process of the same channel *dm.* Then node *DCM* and *DM* will have a link. According to the transformation rules, the bigraphs of the scene are as follows:

The operation event set of the process ***event***:={*in*, *out*, accelerate, *brake*, *enter*, *exit*, *login*, *loginout*, *turn*, *read*}

The following transformation rules are used to transform the physical topology resource changes of the ***event*** in the scene into the bigraphs reaction rules. For different execution process subject to execute the same event corresponding to the change of different resource vector ***r***. For reasons of space, the reaction rules in this article will list only those that are relevant to the current scenario.

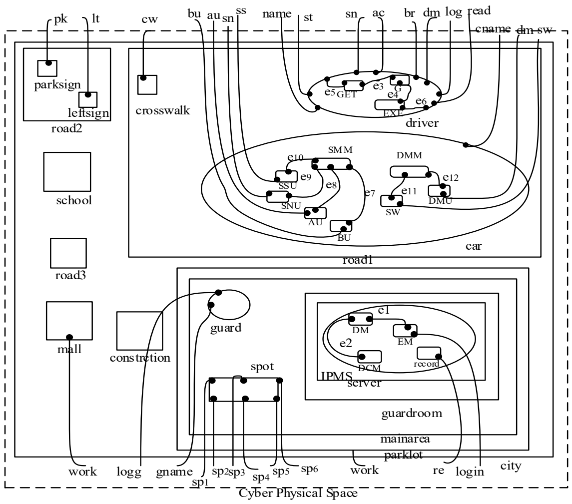


Fig 15: Bigraphs of local physical deployment of the city.

For space reasons, the following transformation rules only describe the mapping rules for entering the *car* event *in* $\xrightarrow{(r,car)}$ *STOP* and using process for parking space resources *enter* $\xrightarrow{(r,parklot)}$ *STOP* .

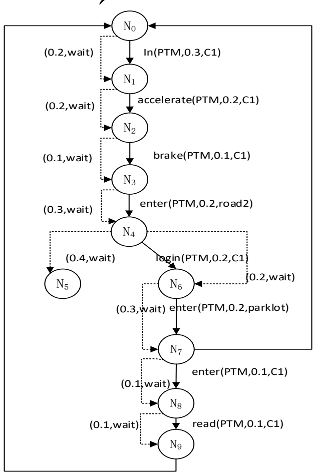


Fig 16: Graph of topology and time constrained state transition.

When the *enter* $\xrightarrow{(r,parklot)}$ *STOP* is executed, the *spot* resource condition for *parklot* is 1≤*spots*≤6. Therefore, only the remaining spots are between 1 and 6, the car can enter *parklot*. When entering the *parklot*, a *spot* enters the *used* state and the topology space description changes.

B. Model checking

(1)Time property verification

The symbol transition system of the ADS process in *Scene modeling* is shown in Fig 16. The time property of the process is verified by the verification algorithm of the time property in Section 4.3. To simplify the size of the state space, the graph of topology and time constrained state transition of *DRIVER* is drawn as Fig 16.

Perform the "Algorithm for time property verification" in Section 4.3 for this process. When nodes N_4 are reached, these two nodes have time constraints. The opening time of *parklot* is (5, 20) . When entering the car at time 18, the depth-first algorithm is executed. It is detected that there is a time constraint at the N_4. The current *totalt* is 2, and the previous execution time can just reach the *parklot* at 20.

Transformation rule 2. The events are mapped to the corresponding reaction rules.

Process Event	Topology space and resources changes
RULES1:= *in* $\xrightarrow{(r,car)}$ *STOP*	*driver[e5,e6,tom,-,-,-,-,-,-,-,-].$0|car[c1].$1|$2→* *car[c1].(driver[e5,e6,tom,-,-,-,-,-,-,-,-].$0|$1)|$2;*
RULES2:= $F_{judge}(F_{ptp}(x,y,z),$ *parklot)∧* *(5, 20)∧* *(1≤spots≤6)>>* *(enter* $\xrightarrow{(r,parklot)}$ *STOP)*	**(1)0 spot used(6 spots left)** *parklot[work].mainarea.($0|spot[-,-,-,-,-,-])|$1→* *parklot[work].mainarea.($0|spot[used,-,-,-,-,-]|$1;* **(2)1spot used(5 spots left)** *parklot[work].mainarea.($0|spot[used,-,-,-,-,-])|$1 →* *parklot[work].mainarea.($0|spot[used,used,-,-,-,-]|$1;* **(3)2 spots used(4 spots left)** *parklot[work].mainarea.($0|spot[used,used,-,-,-,-])|$1→* *parklot[work].mainarea.($0|spot[used,used,used,-,-,-]|$1;* **(4)3 spots used(3 spots left)** *parklot[work].mainarea.($0|spot[used,used,used,-,-,-])|$1→* *parklot[work].mainarea.($0|spot[used,used,used,used,-,-]|$1;* **(5)4 spots used(2 spots left)** *parklot[work].mainarea.($0|spot[used,used,used,used,-,-])|$1→* *parklot[work].mainarea.($0|spot[used,used,used,used,used,-]|$1;* **(6)5 spots used(1 spot left)** *parklot[work].mainarea.($0|spot[used,used,used,used,used,-])|$1→* *parklot[work].mainarea.($0|spot[used,used,used,used,used,used]|$1;*

(2)Deadlock status positioning and modification

For the above state transition graph, the deadlock node is located through the algorithm for finding the deadlock *getdeadlock*() and the deadlock node set ***Deadlock***:={N_5}. It is found that the N_5 is parked at the *parklot* without performing other actions, because *parklot* is open at (5, 20). Therefore, it is not open at other times and the deadlock state needs to be modified: add relevant edges to the node to make it or delete the node. Therefore, at N_5, it can delete the N_5. That is, find a parking lot that is open all day or perform fault-tolerant plans. The fault-tolerant processing measure of node N_{10} (such as roadside parking spaces) is added, and driver enters the *parklot* the next day. The following are the choices for deadlock: looking for a temporary parking space. The modified state space diagram is shown in Fig 17.

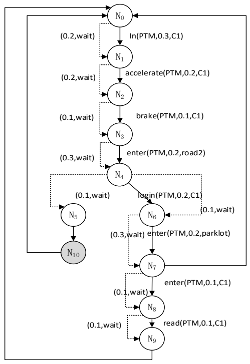


Fig 17: Modified graph of topology and time constrained state transition.

Then the RS-TCSP model of this process needs to be modified accordingly:

$brake\square F_{judge}(F_{ptp}(x,y,z),\ parklot)\wedge(5,20)\vdash error$ should be added to the DRIVER process for fault tolerance processing.

(3)Physical topology property verification

After we input the above model, rules and the following properties into the BigMC, we perform model checking. We enter a statement for the following properties into BigMC.

%property CppResourceSecue !matches (parklot[work].mainarea.(guard[jack,-,-]|guardroom. (driver[e5, e6, tom, -, -, -, -, -, -, -, -]. $1|$2)|$3)|$4)

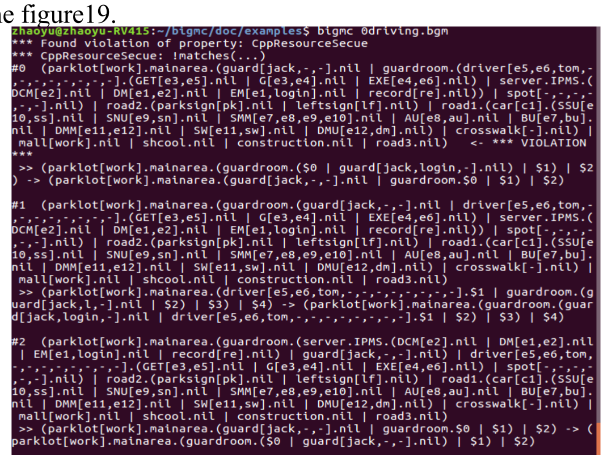
%property CppSpotsSecue !matches (parklot[work].mainarea.($0|spot[used,used,used,used,used,used]|car[c1].(driver[e5,e6,tom,s,-,-,-,-,-,-,-].$1)|$2)|$3).

Fig 18: The statements for checking the properties.

The first property *CppResourceSecue* is that the *driver* is not allowed to be alone in the *guardroom* to protect the safety of the resources in the *guardroom*.

The second property is that it is not allowed the *car* to enter the *mainarea* of the *parklot* when the spot resources are occupied.

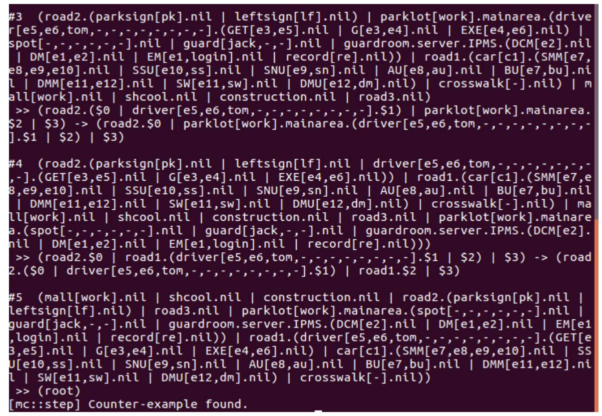
The initial state and the reaction rules are input into BigMC, resulting in a counterexample. The path is shown in the figure19.



```
zhaoyu@zhaoyu-RV415:~/bigmc/doc/examples$ bigmc 0driving.bgm
*** Found violation of property: CppResourceSecue
*** CppResourceSecue: !matches(...)
#0  (parklot[work].mainarea.(guard[jack,-,-].nil | guardroom.(driver[e5,e6,tom,-
,-,-,-,-,-,-,-].(GET[e3,e5].nil | G[e3,e4].nil | EXE[e4,e6].nil) | server.IPMS.(
DCM[e2].nil | DM[e1,e2].nil | EM[e1,login].nil | record[re].nil)) | spot[-,-,-,-
,-,-].nil) | road2.(parksign[pk].nil | leftsign[lf].nil) | road1.(car[c1].(SSU[e
10,ss].nil | SNU[e9,sn].nil | SMM[e7,e8,e9,e10].nil | AU[e8,au].nil | BU[e7,bu].
nil | DMM[e11,e12].nil | SW[e11,sw].nil | DMU[e12,dm].nil) | crosswalk[-].nil) |
 mall[work].nil | shcool.nil | construction.nil | road3.nil)   <- *** VIOLATION
***
 >> (parklot[work].mainarea.(guardroom.($0 | guard[jack,login,-].nil) | $1) | $2
) -> (parklot[work].mainarea.(guard[jack,-,-].nil | guardroom.$0 | $1) | $2)

#1  (parklot[work].mainarea.(guardroom.(guard[jack,-,-].nil | driver[e5,e6,tom,-
,-,-,-,-,-,-,-].(GET[e3,e5].nil | G[e3,e4].nil | EXE[e4,e6].nil) | server.IPMS.(
DCM[e2].nil | DM[e1,e2].nil | EM[e1,login].nil | record[re].nil)) | spot[-,-,-,-
,-,-].nil) | road2.(parksign[pk].nil | leftsign[lf].nil) | road1.(car[c1].(SSU[e
10,ss].nil | SNU[e9,sn].nil | SMM[e7,e8,e9,e10].nil | AU[e8,au].nil | BU[e7,bu].
nil | DMM[e11,e12].nil | SW[e11,sw].nil | DMU[e12,dm].nil) | crosswalk[-].nil) |
 mall[work].nil | shcool.nil | construction.nil | road3.nil)
 >> (parklot[work].mainarea.(driver[e5,e6,tom,-,-,-,-,-,-,-,-].$1 | guardroom.(g
uard[jack,l,-].nil | $2) | $3) | $4) -> (parklot[work].mainarea.(guardroom.(guar
d[jack,login,-].nil | driver[e5,e6,tom,-,-,-,-,-,-,-,-].$1 | $2) | $3) | $4)

#2  (parklot[work].mainarea.(guardroom.(server.IPMS.(DCM[e2].nil | DM[e1,e2].nil
 | EM[e1,login].nil | record[re].nil) | guard[jack,-,-].nil) | driver[e5,e6,tom,
-,-,-,-,-,-,-,-].(GET[e3,e5].nil | G[e3,e4].nil | EXE[e4,e6].nil) | spot[-,-,-,-
,-,-].nil) | road2.(parksign[pk].nil | leftsign[lf].nil) | road1.(car[c1].(SSU[e
10,ss].nil | SNU[e9,sn].nil | SMM[e7,e8,e9,e10].nil | AU[e8,au].nil | BU[e7,bu].
nil | DMM[e11,e12].nil | SW[e11,sw].nil | DMU[e12,dm].nil) | crosswalk[-].nil) |
 mall[work].nil | shcool.nil | construction.nil | road3.nil)
 >> (parklot[work].mainarea.(guard[jack,-,-].nil | guardroom.$0 | $1) | $2) -> (
parklot[work].mainarea.(guardroom.($0 | guard[jack,-,-].nil) | $1) | $2)
```



```
#3  (road2.(parksign[pk].nil | leftsign[lf].nil) | parklot[work].mainarea.(drive
r[e5,e6,tom,-,-,-,-,-,-,-,-].(GET[e3,e5].nil | G[e3,e4].nil | EXE[e4,e6].nil) |
spot[-,-,-,-,-,-].nil | guard[jack,-,-].nil | guardroom.server.IPMS.(DCM[e2].nil
 | DM[e1,e2].nil | EM[e1,login].nil | record[re].nil)) | road1.(car[c1].(SMM[e7,
e8,e9,e10].nil | SSU[e10,ss].nil | SNU[e9,sn].nil | AU[e8,au].nil | BU[e7,bu].ni
l | DMM[e11,e12].nil | SW[e11,sw].nil | DMU[e12,dm].nil) | crosswalk[-].nil) | m
all[work].nil | shcool.nil | construction.nil | road3.nil)
 >> (road2.($0 | driver[e5,e6,tom,-,-,-,-,-,-,-,-].$1) | parklot[work].mainarea.
$2 | $3) -> (road2.$0 | parklot[work].mainarea.(driver[e5,e6,tom,-,-,-,-,-,-,-,-
].$1 | $2) | $3)

#4  (road2.(parksign[pk].nil | leftsign[lf].nil | driver[e5,e6,tom,-,-,-,-,-,-,-
,-].(GET[e3,e5].nil | G[e3,e4].nil | EXE[e4,e6].nil)) | road1.(car[c1].(SMM[e7,e
8,e9,e10].nil | SSU[e10,ss].nil | SNU[e9,sn].nil | AU[e8,au].nil | BU[e7,bu].nil
 | DMM[e11,e12].nil | SW[e11,sw].nil | DMU[e12,dm].nil) | crosswalk[-].nil) | ma
ll[work].nil | shcool.nil | construction.nil | road3.nil | parklot[work].mainare
a.(spot[-,-,-,-,-,-].nil | guard[jack,-,-].nil | guardroom.server.IPMS.(DCM[e2].
nil | DM[e1,e2].nil | EM[e1,login].nil | record[re].nil)))
 >> (road2.$0 | road1.(driver[e5,e6,tom,-,-,-,-,-,-,-,-].$1 | $2) | $3) -> (road
2.($0 | driver[e5,e6,tom,-,-,-,-,-,-,-,-].$1) | road1.$2 | $3)

#5  (mall[work].nil | shcool.nil | construction.nil | road2.(parksign[pk].nil |
leftsign[lf].nil) | road3.nil | parklot[work].mainarea.(spot[-,-,-,-,-,-].nil |
guard[jack,-,-].nil | guardroom.server.IPMS.(DCM[e2].nil | DM[e1,e2].nil | EM[e1
,login].nil | record[re].nil)) | road1.(driver[e5,e6,tom,-,-,-,-,-,-,-,-].(GET[e
3,e5].nil | G[e3,e4].nil | EXE[e4,e6].nil) | car[c1].(SMM[e7,e8,e9,e10].nil | SS
U[e10,ss].nil | SNU[e9,sn].nil | AU[e8,au].nil | BU[e7,bu].nil | DMM[e11,e12].ni
l | SW[e11,sw].nil | DMU[e12,dm].nil) | crosswalk[-].nil))
 >> (root)
[mc::step] Counter-example found.
```

Fig 19: Modified graph of state space.

This counterexample path violates the *CppResourceSecue* property. After finding the counterexample, the BigMC finds all nodes from the initial node to the current violation node. The counterexample path is shown in the figure. For this counterexample, the path is 5(*root*)→4(*driver enters road2*)→3(*driver enters mainarea*)→2(*guard enters guardroom*)→1(*guard logins IPMS*)→0(*guard logouts and exits guardroom*). Therefore, we should specify that when the *guard* leaves the *guardroom*, the *guardroom* will not allow any more *drivers* in the room. Modify *GUARD* in *RS-TCSP* to:

GUARD=

$$\mu X \bullet (enter \xrightarrow{(PIM,(0.3,0.2),guardroom)} login \xrightarrow{(PIM,(0.2,0),IPMS)} (driver < 1) >> exit \xrightarrow{(PIM,(0.1,0.1),guardroom)} logout \xrightarrow{(PIM,(0.2,0.3),IPMS)} X)$$

Fig 20: Modified GUARD process.

Also, RULES was modified as figure 21.

RULES:=((*driver*<1)>>*exit* $\xrightarrow{(r,\text{mainarea})}$ *STOP*):*parklot*[work].*mainarea*.(*driver*[e5,e6,tom,-,-,-,-,-,-,-,-].$0|(*guardroom*.($1|*guard*[jack,login,-]))|$2)|$3 →*parklot*[work].*mainarea*.(*driver*[e5,e6,tom,-,-,-,-,-,-,-,-].$0|(*guard*[jack,-,-]|*guardroom*.$1|$2))|$3;.

Fig 21: Modified RULE.

As the physical topology space of the *driver* and *guard* changes, the physical space does not meet the resource safety requirements in the *guardroom*, and the modified process algebra model ensures the resource safety in the *guardroom*.

VII. Conclusion

The safety of automated vehicles has been widely concerned by people. To solve the safety problem of space-time constraint resource of automated vehicles, an algebraic model for space and time resource constraints of CPS and its verification method are proposes in this paper.

However, the scene of automated vehicles is complex, and we have only made brief modifications to the problems such as deadlock and unreachable in the model detection. We will continue to study how to modify the model according to the scene of automated vehicles.

ACKNOWLEDGMENT

The authors acknowledge the support from the National Natural Science Foundation of China under Grant No.62077029; the CCF-Huawei Populus Grove Fund under Grant No.CCF-HuaweiFM202209; the Applied Basic Research Program of Xuzhou under Grant No.KC19004; the Open Project Fund of Key Laboratory of Safety-Critical Software Ministry of Industry and Information Technology under Grant No.NJ2020022;

REFERENCES

[1] A Particularly Serious Railway Traffific Accident on the NingboWenzhou Line. Accessed: Dec. 7, 2015. [Online]. Available: https://baike.so.com/doc/5381626-5617962.html

[2] Luo CX, Wang R, Guan Y, Li XJ, Shi ZP, Song XY. CPS integrated modeling method for real-time data[J]. Journal of Software, 2019, 30(07):1966-1979.

[3] Chen XY,Zhu Y,Zhao Y,Wang JY.Hybrid AADL Modeling and Model Transformation for CPS Time and Space Properties Verification[J]. Journal of software,2021,32(06):1779-1798.

[4] Xu JQ,Guo XJ,Wang JF,Li HQ,Zhao H.Research on CPS Resource Service Model and Resource Scheduling[J].Journal of Computer Science,2018,41(10):2330-2343.

[5] Orumwense E F, Abo-Al-Ez K M. Energy management in a cloud-based cyber-physical system[J]. IET Cyber-Physical Systems: Theory and Applications. 2021, 6(2): 93-103.

[6] Apat H K, Bhaisare K, Sahoo B, et al. Energy Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System[C]. In: Gunupur, India: Institute of Electrical and Electronics Engineers Inc., 2020.

[7] Zhu Y,Xiao FX,Zhou H,Zhang GQ.A method of energy consumption modeling and analysis for embedded real-time system software is presented[J].Computer research and development. 2014,51(04):848-855.

[8] Cao Y, Huang Z, Kan S, et al. Specification and verification of a topology-aware access control model for cyber-physical space[J]. Tsinghua science and technology. 2019, 24(5): 497-519.

[9] Zhang, Lanfang,et al. "A Novel Spatio-Temporal Access Control Model for Online Social Networks and Visual Verification." IJCAC vol.11, no.2 2021: pp.17-31. http://doi.org/10.4018/IJCAC.2021040102

[10] C. A. R. Hoare, "Communicating sequential processes, ", Communications of the ACM, vol. 21, pp.666-677, 1978.

[11] G. M. Reed and A. W. Roscoe, "A timed model for communicating sequential processes,", Theor. Comput. Sci., vol. 58, pp. 249-261, January, 1988.

[12] Milner R,The Space and Motion of Communicating Agents[M]. Cambridge, UK: Cambridge University Press, 2009

[13] Faithfull A J,Perrone G,Hildebrandt T T.Big red:A development environment for bigraphs[J].Electronic Communications of the Easst,2013,61:1-10

[14] Perrone G,Debois S,Hildebrandt T T.A verification environment for bigraphs[J].Innovations in Systems and Software Engineering,2013,9 (2) :95-104

[15] I. Graja, S. Kallel, N. Guermouche, S. Cheikhrouhou and A. H. Kacem, "Modelling and verifying time-aware processes for cyber-physical environments," IET Softw., vol. 13, pp. 36-48, January, 2019.

[16] J. Zhang, Y. Zhu and F. Xiao, "Modelling and analysis of real-time and reliability for WSN-based CPS,", International Journal of Internet Protocol Technology, vol. 12, pp. 76-84, May, 2019.

[17] Q. Su, T. Wang, T.M. Chen, R.R. Chen, "CPS Security Modeling and Validation Based on Time Automaton, ", Information Security Research., vol. 3, pp.601-609, July, 2017.

[18] M. U. Tariq, J. Florence and M. Wolf, "Improving the safety and security of wide-area cyber-physical systems through a resource-aware, service-oriented development methodology," P. IEEE, vol. 106, pp. 144-159, January, 2018.

[19] H. Tran, L. V. Nguyen, P. Musau, W. Xiang and T. T. Johnson, "Decentralized Real-Time Safety Verification for Distributed Cyber-Physical Systems,", Kongens Lyngby, Denmark, vol. 11535 LNCS, Springer Verlag, pp. 261-277, January, 2019.

[20] C. Sun, S. Cheng, K. Yuan, J. Sun, Y. Song, Z. Wu, X. Yang and Y. Shi, "Real Time Simulation Platform of Power Cyber-physical System Based on Node Mapping Model,", Dianwang Jishu/Power System Technology, vol. 43, pp. 2368-2375, January, 2019.

[21] M. Ring, F. Bornebusch, C. Luth, R. Wille and R. Drechsler, "Verification Runtime Analysis: Get the Most out of Partial Verification,", Grenoble, France, Institute of Electrical and Electronics Engineers Inc., pp. 873-878, March, 2020.

[22] R. Wang, Y. Guan, X. Li and R. Zhang, "Formal Verification of CAN Bus in Cyber Physical System,", Macau, China, Institute of Electrical and Electronics Engineers Inc., pp. 249-255, December, 2020.

[23] J. Liu, JY. Wang, ZW. Li, HY. Sun, YJ. W, DH. Du, XH. Chen and MS Chen. "ST-LUSTRE: A Novel Spatio-Temporal Language Towards Safety-Critical Cyber-Physical Systems, ", International Journal of Performability Engineering, pp. 1219-1232, June, 2017.

[24] C. Tsigkanos, T. Kehrer and C. Ghezzi, "Modeling and verification of evolving cyber-physical spaces,", Paderborn, Germany, vol. Part F130154, Association for Computing Machinery, pp. 38-48, January, 2017.

[25] C. Tsigkanos, L. Pasquale, C. Ghezzi and B. Nuseibeh, "On the Interplay Between Cyber and Physical Spaces for Adaptive Security,", IEEE T. Depend. Secure, vol. 15, pp. 466-480, 2018.

[26] T. Li, X. Chen, H. Sun, J. Liu, J. Yang, C. Yang and J. Sun, "Modeling and verification of spatio-temporal intelligent transportation systems,", Guangzhou, China, Institute of Electrical and Electronics Engineers Inc., pp. 568-575, January, 2020.

[27] Y. Cao, Z. Huang, S. Kan, D. Fan, and Y. Yang, "Specification and verification of a topology-aware access control model for cyber-physical space,", Tsinghua Science and Technology, vol. 24, pp. 497-519, January, 2019.

[28] S. Mondal, S. Sural, and V. Atluri, "Security analysis of GTRBAC and its variants using model checking,", Comput. Secur., vol. 30, pp. 128-147, 2011.

[29] M. Toahchoodee, I. Ray, and V. Atluri, "On the formalization and analysis of a spatio-temporal role-based access control model,", Journal of computer security, vol. 19, pp. 399-452, January, 2011.

[30] H. Ma, Y. Wang and R. Xiong, "From Time to Space: Automatic Annotation of Unmarked Traffic Scene Based on Trajectory Data," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2018, pp. 1177-1182.

[31] G. Dai, P. K. Paluri, T. Carmichael, A. M. K. Cheng and R. Miikkulainen, "Work-in-Progress: Leveraging the Selfless Driving Model to Reduce Vehicular Network Congestion," 2019 IEEE Real-Time Systems Symposium (RTSS), 2019.

[32] M. P. I. Dias, E. Grigoreva, C. M. Machuca, L. Wosinska and E. Wong, "Delay-Constrained Framework for Road Safety and Energy-Efficient Intelligent Transportation Systems," 2017 European Conference on Optical Communication (ECOC), 2017.

[33] R.Parasuraman,T.B.Sheridan,and C.D.Wickens, A model for types and levels of human interaction with automation[J]. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans. 2000, 30(3): 286-297.