Chinese Journal of Aeronautics, (2016), 29(4): 1027–1041

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

A hazard analysis via an improved timed colored petri net with time–space coupling safety constraint

Li Zelin, Wang Shihai *, Zhao Tingdi, Liu Bin

Science and Technology on Reliability and Environmental Engineering Laboratory, School of Reliability and Systems Engineering, Beihang University, Beijing 100083, China

Received 15 September 2015; revised 16 February 2016; accepted 19 March 2016
Available online 23 June 2016

KEYWORDS

Petri nets;
Real-time systems;
Resource allocation;
System modeling;
Time–space coupling safety constant

Abstract Petri nets are graphical and mathematical tools that are applicable to many systems for modeling, simulation, and analysis. With the emergence of the concept of partitioning in time and space domains proposed in avionics application standard software interface (ARINC 653), it has become difficult to analyze time–space coupling hazards resulting from resource partitioning using classical or advanced Petri nets. In this paper, we propose a time–space coupling safety constraint and an improved timed colored Petri net with imposed time–space coupling safety constraints (TCCP-NET) to fill this requirement gap. Time–space coupling hazard analysis is conducted in three steps: specification modeling, simulation execution, and results analysis. A TCCP-NET is employed to model and analyze integrated modular avionics (IMA), a real-time, safety-critical system. The analysis results are used to verify whether there exist time–space coupling hazards at runtime. The method we propose demonstrates superior modeling of safety-critical real-time systems as it can specify resource allocations in both time and space domains. TCCP-NETs can effectively detect underlying time–space coupling hazards.

1. Introduction

With the development of computer and software technology and incremental increases in reliability, availability, and safety requirements for safety-critical real-time systems, the concept of partitioning in time and space domains is proposed in avionics application standard software interface (ARINC 653).[1] Multiple applications of different software levels share resources and are hosted on one common hardware platform. Applications are isolated by time and space partitioning to efficiently prevent fault propagation. Current operating systems with the ARINC 653 standard include: VxWorks653, LynxOS-178, and Integrity-178B. This paper will introduce time–space coupling safety issues in resource configurations mainly based on an integrated modular avionics (IMA) system.

An IMA architecture provides a shared platform with reusable and flexible hardware and software resources.[2] By replacing numerous separate, centralized common processing modules, IMA architectures benefit from low

* Corresponding author. Tel.: +86 10 82313598.
E-mail address: wangshihai@buaa.edu.cn (S. Wang).

Peer review under responsibility of Editorial Committee of CJA.

http://dx.doi.org/10.1016/j.cja.2016.04.016
1000-9361 © 2016 Chinese Society of Aeronautics and Astronautics. Production and hosting by Elsevier Ltd.

power consumption and maintenance savings, but bring with them potential safety issues.[3] The platform can host avionics functions of various safety levels while ensuring the integrity of the system through its robust partitioning mechanism. Time and space partitioning is the core concept of an IMA system. This type of partitioning requires adequate temporal resources (time slots) and spatial resources (memory space) to be allocated to each partition in the design phase in order to ensure proper execution and satisfy real-time constraint requirements. System designers usually configure time slots and memory space of each partition separately, ignoring time–space dynamic connection requirements between partition resources, which are heavy, complicated, and error-prone.[4] Spatial resource requirements of applications vary from and can be affected by their allocated temporal resources. The interactive relationship between temporal and spatial resources introduces new safety issues.

Petri nets (PNs) were first introduced in the doctoral dissertation, "Communication with Automata", of Carl Adam Petri.[5] PNs are an information flow model of network structure with parallelism, uncertainty and synchronism. PNs provide formal methods to establish mathematical models that can describe system behaviors and also provide a graphical interface that helps system modeling and analysis.[6] PNs have been proven to be effective graphical, mathematical modeling and analysis tools that are widely used to model asynchronous, concurrent computer systems. PNs have been expanded and contain different features and functions for specific modeling purposes such as timed PNs[7], colored PNs[8], and hierarchy PNs.[9] For analyzing time–space coupling hazards in safety-critical real-time systems, all of these current methods seem unsatisfactory. To deal with resource coupling in a time–space domain, this paper proposes a time–space coupling safety constraint. Furthermore, a new, timed colored Petri net with time–space coupling safety constraints (TCCP-NET) is introduced and employed for time–space coupling hazard analysis.

This paper is organized as follows: Section 2 – a brief introduction of classical PNs, colored PNs, timed PNs, their modeling and analysis methods, and limitations. An IMA system, as an example, is introduced to illustrate the concept of time–space partitioning where time–space coupling hazards are possibly introduced; Section 3 – specifications of time–space coupling and time–space coupling safety constraints; Section 4 – a new, timed colored PN is proposed with time–space coupling safety constraints. Its modeling process and analysis methods are introduced; Section 5 – a case study and discussion are used to demonstrate the effectiveness of the proposed modeling and analysis methods; Section 6 – conclusion and future work.

2. Background

2.1. PN

2.1.1. Classical PN

A classical PN[10] has two different types of nodes: places (circles), transitions (rectangles). The different nodes are connected by directed arcs (arrows). A place can contain any number of tokens, and the distribution of tokens over places is called a marking, which represents the allocation of resources. If all input places connected to a transition have more than one token, that transition can be fired. Tokens in input places are removed and tokens are generated in output places. Any transition in a PN may be fired concurrently if it is enabled. Due to uncertainty and concurrency, there are many distributions of tokens that represent various markings.

Table 1 Interpretations of places and transitions.[11]

Input place	Transition	Output place
Required resources	Tasks	Freed resources
Input data	Computations	Output data
Input signals	Signals processing	Output signals
Buffers/registers	Processors	Buffers/registers

In the modeling process, the state of a system is generally denoted by places, and behaviors that change system state are denoted by transitions. Some typical interpretations of transitions and their input places and output places are shown in Table 1.

Complex behaviors in the system can be modeled by a classical PN. The model is then used to analyze behavioral and structural properties of the system.[12] However, a number of limitations exist in this type of PN:

(1) All tokens are identical and descriptions of resource types are too simple.
(2) It is difficult for existing attributes in PNs to describe additional system properties.
(3) The concept of time is not taken into consideration in the modeling and simulation process.

2.1.2. Timed PN

The timed PN (TP-NET)[13] is derived from classic PNs. It models interactions between activities, taking into account time properties. Time is introduced into a TP-NET in different ways:

(1) Time associated with tokens. Each token is associated with a time value that indicates when the token is available to fire a transition.
(2) Time associated with arcs. Each arc is associated with a delay t, which indicates a token takes t time to travel between two nodes.
(3) Time associated with transitions. Each transition can be associated with a delay t or delay interval [start, end], which represents time required to fire a transition.

With such time properties, TP-NETs have been widely used for modeling and performance evaluation, especially in real-time systems.[14–16]

The structure of a TP-NET is $N_{\text{timed}} = (P, T, A, W, M_0, I)$, where $N = (P, T, A, W, M_0)$, a marked PN. Symbols P, T, A, and W represent places, transitions, arcs and initial marking respectively. $I(t)$ denotes firing time of a transition and is called the firing time function.

The TP-NET specifies how much time an individual operation takes and how long it is necessary wait before it is ready. In TP-NETs, powerful time properties can help model time-dependent system behaviors that can be used in simulation to analyze problems in a time domain.

2.1.3. Colored PN

A colored PN (CP-NET) is a tuple $N_{\text{colored}} = (P, T, A, \Sigma, C, N, E, G, I)$[17], which preserves existing properties of PNs and adds modeling formalism to distinguish between different tokens. Definition of these symbols can refer to Ref.[17] CP-NETs provide graphical notations and basic primitives to construct concurrent, synchronous, and communicating models.[18] Tokens in places can be assigned a data value consisting of a simple number, string, or data structure. This value is called token color and its type is called a color set, which it is identical to data type in programming languages.

When a PN is used to model a system, tokens often represent objects or resources in those systems. These resources may have attributes that are not easily represented by a simple token. For example, Fig. 1 shows simple resource utilization. The first place p_1 has three resources (Resource 1, Resource 2, and Resource 3). Although both Channels A and B have the same model, and the three resources are similarly identical, they are distinct in that Resource 1 must use Channel A, while Resource 3 must use Channel B. Resource 2 can use either Channel A or Channel B.

A CP-NET can model similar systems according to their specification. System behaviors can be analyzed either by means of simulation or occurrence graphs. It is very efficient to model resource allocation and analyze problems in a space domain using CP-NETs.

In general, TP-NETs are used to analyze the time-dependent behaviors of a system, such as scheduling analysis in real-time systems. In Ref.[19], PNs with imposed timing constraints are employed to analyze schedulability with the assumption that resources are always available and adequate upon request. CP-NETs are used to analyze problems in resource allocations, especially for shared or dedicated resources.[6] It is worth noting that current PNs are only used to analyze either time-dependent behaviors of a system or resource allocations. They appear to be incapable of analyzing the problems caused by interactions between temporal resources and spatial resources, called time–space coupling safety issues.

2.2. IMA system

2.2.1. IMA architectures

Fig. 2 shows the typical three-layer structure defined by ARINC 653:[20] the application layer, operating system layer, and support layer. ARINC 653 defines the application interface (API) and partitioning to decouple shared process platforms, namely, application software and operating system. Operating systems and hardware can be isolated.

Essentially, the main objective of the IMA system is to extend the flexibility of a distributed architecture to support different safety-critical programs. The task scheduling of distributed real-time system not only satisfies the basic requirements of distributed system scheduling, but also ensures each task can be completed at a determined time. Based on this IMA structure, a hierarchical scheduling model is introduced for resource partitioning and task scheduling in IMA systems.[21] In the operating system layer, time slice rotation is employed to activate each partition. All processes of a partition are scheduled according to their pre-determined task priorities within the time slots of each partition, as shown in Fig. 3. In this model, there are two schedulers of different layers. One is responsible for the cyclic scheduling of partitions in IMA systems, and the other is used to schedule tasks in an activated partition within assigned time slots.[22]

2.2.2. Time and space partitioning

In the ARINC 653 standard, the partition is the core concept and the unit of scheduling and resource allocation, including time and space partitioning.[23,24] The computing resources are shared between applications via time and space partitioning. Partitions with different critical levels can be executed in the same module due to a robustly partitioned architecture.[25]

Space partitioning provides a runtime environment where different critical applications can run concurrently on the same computing platform without spatial resource conflicts.[26] Memory areas in different partitions are protected from unauthorized access from other partitions by space partitioning mechanisms. In this framework, the memory space of each partition has its own logical address into which processes, data, and resources are placed. They cannot be occupied or interrupted by applications running in other partitions.

With time partitioning, a guaranteed portion of CPU time can be allocated to each partitioned application running in the

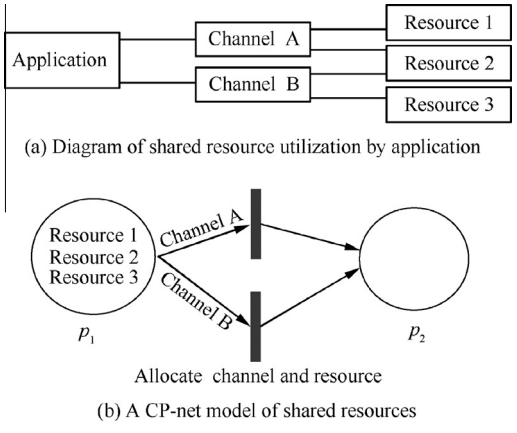

Fig. 1 Using a CP-NET to model shared resource utilization.

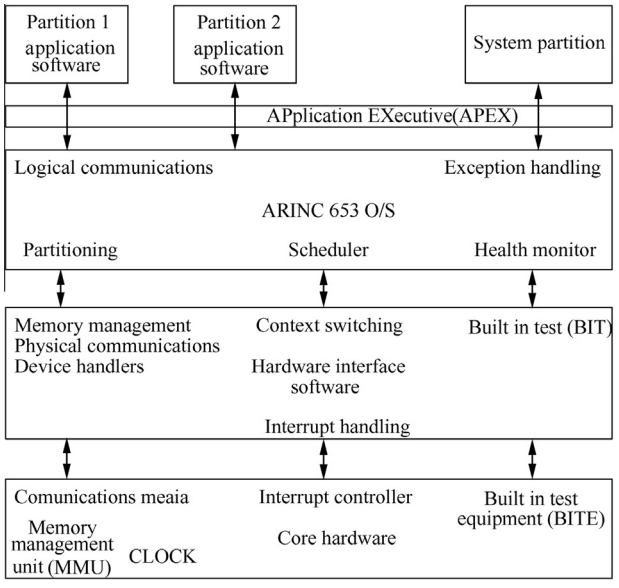

Fig. 2 IMA architecture (ARINC 653).

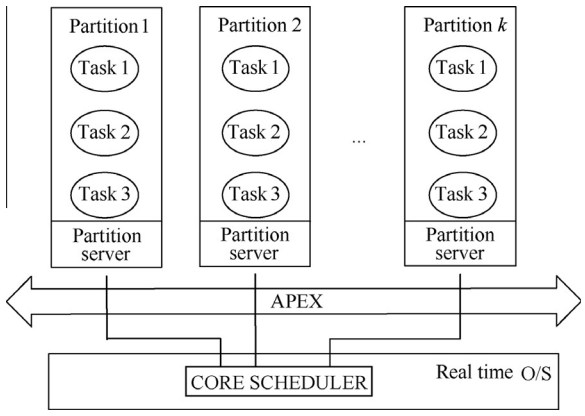

Fig. 3 A hierarchical scheduling model.

same module, without regard to other partitions. Each partition can only acquire computing resources within its time slots. The system will continue to run processes in other partitions according to schedule, even if errors occur. The sum of time slots allocated to each partition in one cycle forms a major time frame[27] (MTF) as shown in Fig. 4.

Space partitioning implies that a partition has exclusive access to its own resources, like memory and buffers. With space partitioning, an application can be protected from any erroneous behavior of other applications while sharing the same resources. Time partitioning guarantees a partition's monopoly on the use of a pre-allocated processing time without any interference from other partitions. The expected benefits of time and space partitioning are explained as follows:

(1) Applications can be developed separately, regardless of the underlying hardware; different safety-critical software can be integrated into a common processing platform, which does not reduce system reliability or safety.
(2) Resources (computing, communication and input/output) are shared by hosted functions and utilization is improved, thus weight, power consumption, and volume are significantly reduced.

Applications in partitions communicate with each other whether they run on the same module or not. A message can be sent from a single partition to one or more others through channels, logical links between communication partitions. A channel has one or more distinct end points called ports. Each individual port has two modes of operation: sampling mode or queuing mode.

In sampling mode, a message in the source port is refreshed by updated data with identical structure, and remains until transmitted or overwritten by a new occurrence of the message. Destination ports can only access the latest message received. In queuing mode, a message will not be overwritten during transmission by any new message sent on the same channel. When a message reaches the destination port, it is not consumed by the partition, but stored in a message queue. Messages should be transmitted from source port to destination port without data loss.

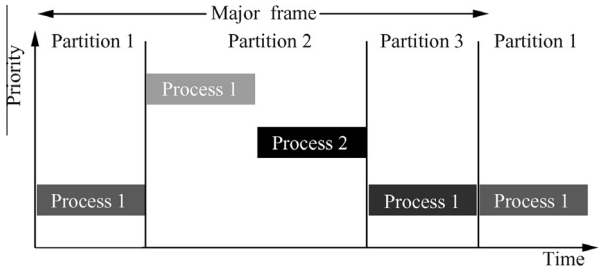

Fig. 4 Time partitioning structure.

In order to have successful communication between partitions, attributes of each port should be configured properly. Port attributes are shown in Table 2. The name of each port is unique to distinguish between them. Data in a message cannot exceed the maximum message size otherwise the integrity and correctness of the data are not fully guaranteed. All ports can operate in only one direction and cannot be used for bidirectional transmission. In queuing mode, the number of messages should be less than the maximum indicated.

In all cases, applications should allocate enough memory space to ensure maximum receipt of messages regardless of application protocol. Each communication node is considered to be configured to define resource allocation and operational mode of the port. It is worth noting that attributes and operational mode of each port cannot be changed at run-time.

2.2.3. Configuration methods

Resource allocation in IMA systems consists of spatial resources (memory, bandwidth) and temporal resources (time slots).

Allocation policy of temporal resources is derived from task scheduling algorithms. Scheduling is the essence of temporal resource allocation, and the time constraints of a task are emphasized in real-time task scheduling systems. In general, real-time scheduling algorithms are classified according to different criteria; Table 3 shows one such classification for real-time systems.

Task scheduling algorithms widely used in real-time systems include rate-monotonic (RM), earliest deadline first (EDF), and least slack time (LST) scheduling.[28] A set of tasks is considered to be schedulable only if all tasks can be invoked and executed within the given time. For the set of tasks $T_{\text{task}} = \{t_{\text{task1}}, t_{\text{task2}}, \ldots, t_{\text{taskn}}\}$, Eq. (1) is the precondition for being schedulable. Otherwise, the set of tasks T_{task} cannot be scheduled correctly using a scheduling algorithm.

$$U = \sum_{i=1}^{n} (c_i / T_{ri}) \leqslant 1 \tag{1}$$

where U is processor utilization, c_i is computation time, and T_{ri} is release period.

Table 2 Port attributes.

Port type	Name	Definition
Sampling port	Name	The name of the port
	Max message size	Maximum message size
	Direction	Source or destination
	Name	The name of the port
Queuing port	Max message size	Maximum message size
	Direction	Source or destination
	Max num message	Maximum number of messages

Table 3 Real-time scheduling algorithm classification.

Classification criteria	Classification result
The order of task executions	Static or dynamic
Whether the task can be preempted	Pre-emptive or non-pre-emptive
The system operating environment	Uniprocessor or multiprocessor
System architecture	Centralized or distributed

RM is an algorithm in a static-priority scheduling class.[29] RM scheduling algorithms assign tasks to a fixed priority: tasks with the shortest cycle duration receive the highest priority. RM has been proven to be an optimal static priority algorithm. A set of tasks, which can be scheduled properly using arbitrary fixed priority scheduling algorithms, must also be schedulable using RM algorithms. Because RM scheduling algorithms are static, they have weak adaptability and flexibility at runtime.

EDF is a dynamic priority scheduling algorithm, also known as the deadline driven scheduling algorithm.[30] EDF assigns a priority to a task based on its deadline. When a new task is ready, the priorities of other tasks may need to be adjusted to ensure the task with the closest deadline can be executed first. EDF scheduling algorithms have proven to be optimal for dynamic scheduling on a uniprocessor system where processor utilization may be up to 100%. When a system becomes overloaded instantaneously, however, system behaviors are unpredictable due to missing prioritization of tasks. As a result, industrial real-time systems rarely use this algorithm.

A dynamic priority scheduling algorithm, LST, assigns a dynamic priority to a task based on its slack time, as introduced in Ref.[31]. The shorter the slack time of a task, the higher its priority. LST scheduling algorithms are widely used in embedded systems.

In Refs.[32,33], the scheduling algorithms mainly focus on an appropriate allocation of resources in the temporal domain. However, partition memory should also be taken into consideration. Partition memory resources are usually allocated in terms of engineering experience, and then modified repeatedly until debugging is successful. In Ref.[34], the constraint-based allocation approach is proposed to help allocate avionics resources for system implementation while ensuring safety requirements. In this approach, resource allocation is considered as a traditional combination of software and hardware, and the interactive relationship between temporal and spatial resources is not pointed out as a safety constraint.

When an integrator completes the configuration of time slots and partition memory, the partition applications are considered to be running properly. If there exists a high degree of data sharing, such as inter-partition communications, temporal resources (time slots) and spatial resources (partition memory) interact, and should not be analyzed separately with respect to resource allocation. Although partition configurations are considered appropriate, they may actually raise a time–space coupling hazard in safety-critical software systems. None of the reviewed literature discussed an analysis of time–space dynamic connections in the allocation of temporal spatial resources. Therefore, at the conclusion of this paper, we propose a modeling and analysis approach with time–space coupling safety constraints to improve hazard analysis of partition configurations.

3. Time–space coupling safety constraints

Real-time operating systems compliant with ARINC 653 are an important part of safety-critical real-time systems. The system architecture supplies a shared and common computing platform that hosts multiple applications of different criticalities. It provides benefits of flexibility and scalability and allows applications to be developed independently. Complexity is increased due to high integrity and the sharing of resources. Robust partitioning can effectively limit or isolate different avionics applications. Interface control helps us thoroughly understand interactions between applications that are not eliminated by partitioning. Robust partitioning and interface control facilitate application integration and hazard analysis techniques.

Although interactions between applications and data sharing are a necessary and desirable property of IMA systems, they introduce some safety issues as well. Spatial resource requirements of applications vary from and may be affected by their allocated temporal resources. The interaction between the temporal spatial resources is called resource coupling in a time–space domain. In systems theory, safety is considered as a system control problem rather than a component reliability problem.[35] Accidents resulting from interactions among system components occur due to a lack of adequate constraints imposed on components and system behaviors. In Ref.[36], segregation constraints (spatial and temporal) and a mixed-integer linear programming formulation are employed to obtain suitable allocation of resources. The current methods only focus on either temporal resources or spatial resources, which is limiting with respect to time–space dynamic connections. We propose a time–space coupling safety constraint and then employ a TCCP-NET with it to improve hazard analysis.

The time–space coupling safety constraint imposed on the allocation of resources in real-time systems is:

The spatial resources allocated to a system or application should satisfy the combined requirements of the time resources and internal properties of the system, for instance, the average speed of data generation.

The constraint can be expressed by formula:

$$T_{\text{tem } i} \otimes k_i \leqslant S_{\text{spa } i}, \forall \bar{p}_i \in \bar{P} \tag{2}$$

where the ith application $\bar{p}_i$ is allocated temporal resources $T_{\text{tem } i}$ and spatial resources $S_{\text{spa } i}$. The effect on spatial resources from temporal resources is set to the impact factor k_i. $\bar{P}$ is a set of applications $\bar{p}_i$. The combined relationship of the time resources and internal properties of the system is denoted by the $\otimes$ operator.

This constraint reveals the relationship between the spatial and temporal resources. With the time–space coupling safety constraint in a TCCP-NET, an improved method can be used to analyze resource configurations concerning time–space

dynamics. For a better understanding of this constraint, an IMA system example is given to illustrate its specific forms.

We assume that each task in every partition arrives periodically and needs to send data at a certain rate. Each partition $\bar{p}_i$ has invocation period $T_{\text{per } i}$, and execution duration D_i. Each channel is assigned to a memory size M_{ij}. Internal property k_i is replaced with data generation rate V_{ij}. i and j are the source and destination partition of the channel respectively.

Each partition $\bar{p}_i$ is considered to produce data incessantly within its execution duration D_i. Memory size M_{ij} should satisfy the requirements of the data amount generated within execution duration D_i. The $\otimes$ operator is instantiated by multiplication, so Eq. (2) is accommodated as follows:

$$D_i \times V_{ij} \leqslant M_{ij} \tag{3}$$

Eq. (3) is an exact constraint for IMA system resource configurations. It will be added into a TCCP-NET as a guarding function to help analyze potential time–space coupling hazards. For other real-time systems, the internal properties of the system may change. Eq. (2) should be modified to accommodate such properties.

4. Modeling and analysis using TCCP-NET

A TCCP-NET based on an extension of colored PNs is proposed that can specify the configurations on temporal and spatial resources in terms of time slots and physical memory characteristics. Based on this model, an improved safety analysis method with a coupling safety constraint in time and space domains is introduced. An example IMA system demonstrates how to analyze time–space coupling hazards by using the TCCP-NET. Notably, time–space coupling hazard analysis of IMA systems is about temporal and spatial resource allocation of partitions, and its impact on inter-partition behaviors.

In TCCP-NET models, a global clock is introduced to represent the model time, which may be either discrete or continuous. More specifically, the basic properties of TCCP-NET include colored tokens, arc expression functions, guarding functions, and time attributes. Each token has a color and a time value. Colored tokens can distinguish different resource instances, such as data storage space and channel memory size. A time value of each token represents the earliest time when the token is ready. The routing of different tokens is controlled by arc expression and guarding functions. Delay time interval required for implementing a transition is denoted by the time attribute in each transition.

In this section, the definition and the modeling process of TCCP-NET are discussed in detail.

4.1. TCCP-NET definition

A TCCP-NET is a multi-element structure defined as TCCP-NET $= (R, P, T, A, N, C, G, E, \text{TF}, I)$, where

(1) R is the type of non-empty finite set, called a color set, which indicates the type of tokens in each place;
(2) $P = \{p_1, p_2, p_3, \ldots, p_n\}$. Where P is a collective notation of a finite set of places, p_n denotes nth place;
(3) $T = \{t_1, t_2, \ldots, t_m\}$. T is a finite set of transitions, m is the number of transitions, $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$;
(4) $A \subseteq (P \times T) \cup (T \times P)$. A is a finite set of directed arcs (flow relations) and $P \cap A = T \cap A = \varnothing$;
(5) $N : A \to (P \times T) \cup (T \times P)$. N is a node function. Directed arcs are mapped from A to $(P \times T) \cup (T \times P)$;
(6) $C : P \to R$ or $T \to R$. C is a color function. Each place p_i is mapped to a type $C(p_i)$, and each transition t_i is mapped to a type $C(t_i)$ by color function, which means that each place or transition should have its own set of colors;
(7) $G : T \to$ Expressions, $\exists t, t \in T$, then $\text{Type}(G(t)) = B \cap \text{Type}(\text{Var}(G(t)))$. G a is guarding function, $\text{Type}(v)$ is the type of variable v, $\text{Var}(\text{Expr})$ is variable set of the expression and B is a boolean function;
(8) $E : A \to \text{Expression}. \forall a, a \in A$, then $\text{Type}(E(a)) = C(p_i)_{\text{MS}} \cap \text{Type}(\text{Var}(E(a)))$. E is an arc function, p_i is the place of $N(a)$ and $C(p_i)_{\text{MS}}$ is all the sets of $C(p_i)$;
(9) $\text{TF}(t_i) = \{T_{del1}, T_{del2}, \ldots, T_{\text{del } i}\}$. TF is a time delay set of corresponding transitions, $T_{\text{del } i}$ is the time delay of transition t_i;
(10) $I : P \to$ startedExpression. I is an initial function. Each place p_i is mapped into an expression by the initial function I;

In a TCCP-NET, time concepts and guarding functions are employed. The conditions where transition t_i can be enabled are modified and specific requirements are as follows:

(1) The total number of tokens in place Input should not be less than the number of tokens required by binding elements. $\forall p_i \in \bullet t, E(p_i, t_i)\langle \text{binding} \rangle \leqslant M(p_i)$.
(2) The timestamp should be ready. The value of timestamps carried by binding tokens should be less than or equal to the current model time.
(3) Guarding function is satisfied. $G(t_i)\langle b \rangle = \text{true}$.

In our model, there are some modifications on our TCCP-NET from the original CP-NET that improve the ability of TCCP-NET to analyze resource configuration in real-time systems. More details are as follows:

(1) Directed arcs are mapped from A to $(P \times T) \cup (T \times P)$. Each arc is mapped into a pair, where the first element is called the source node, and the second element is called the destination node. The source node and destination node must be different types (places or transitions). More than one arc between the same pair of nodes can exist in a TCCP-NET as multi-arcs. This helps model resource utilizations in different channels or modes.
(2) Each transition t is mapped to an expression by guarding functions. All variable types in the expression are part of the color set R, and the result of the expression is boolean. Transitions can be implemented only when the guarding function returns a true value. Resource constraints (temporal or spatial) can be added into guarding functions to restrict system behaviors. The time–space coupling safety constraint is added into TCCP-NET as a guarding function.
(3) Each arc is mapped into an expression by an arc function. All variable types in the expression are part of the color set R, and the value of the expression is a multi-set. Arc expression determines the token types

and the conditions in which tokens can be transferred. The @ + operator in arc expressions specify a delay time required to transfer between nodes. In general, arc expressions are essential.

(4) Each transition t_i is given a transition delay $\mathrm{TF}(t_i) = T_{\mathrm{del}\ i}$. When the transition is ready, $T_{\mathrm{del}\ i}$ is needed before implementing the transition t_i. For $t_i \in T$, if $\mathrm{TF}(t_i) = T_{\mathrm{del}\ i} = 0$, there is no time delay needed for a ready transition. Such transitions are called instant transitions and other transitions are referred to as timed transitions. A resource occupancy time interval can be denoted by a transition delay.

The execution of a TCCP-NET model is time driven. System time starts at 0 and all transitions are searched for those that can fire. When these transitions are executed completely, the system clock is increased by a time unit. The next step is to search for transitions that can fire at the current time and execute them, and then increase the system clock by a time unit again. The above steps are repeated until the system clock is equal to the set value. In the end, the developed model is simulated; Time–space coupling hazards can be detected through simulation results.

In the TCCP-NET, a value is attached to every token, the token color, which distinguishes them from each other. In addition, the time concept is involved in tokens, transitions, and arcs. The time stamp of each token indicates the time at which the token is ready to be consumed. The delay time in both transitions and arcs denotes the time interval required for corresponding operations. Guarding functions are used to further constrain firing rules. A transition is considered to be enabled when it satisfies the requirements of the general enabling rules. In addition, the binding element also must be ready (i.e. the time stamp of corresponding tokens should not exceed the current model clock) and the value of guarding function must be true. Hence we are able to utilize these extended features to model a system with coupling in time and space domains, such as partition configurations in IMA systems.

4.2. Modeling process based on TCCP-NET

To demonstrate the modeling approach for TCCP-NET clearly, an example of a simple IMA configuration is employed. IMA architecture may consist of one or more modules connected with each other using a time division multiplexing communication bus such as ARINC 659 and AFDX. Each core module has one or more processors. Several execution partitions, $\bar{P} = \{\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n\}$, run on a processor and are scheduled on statically predefined and fixed time slots within a major time frame. Applications are allocated to corresponding partitions. Each partition $\bar{p}_i$ has a period $T_{\mathrm{per}\ i}$, duration D_i, and offset O_i relative to the start of the MTF. Each partition invocation must complete its computation and send out the resulting messages before the next invocation of the same partition.

Details of the partition configuration are given in Table 4. Each partition $\bar{p}_i$ has an invocation period $T_{\mathrm{per}\ i}$, duration D_i, and offset O_i. Each channel is assigned to a memory size M_{ij} and data generation rate V_{ij}. Partition $\bar{p}_1$ has two channels. One is Channel ch12 ($\bar{p}_1 \rightarrow \bar{p}_2$) in which partition $\bar{p}_1$ sends data to partition $\bar{p}_2$ at a rate V_{12}. The other is channel ch13 ($\bar{p}_1 \rightarrow \bar{p}_3$) in which partition $\bar{p}_1$ sends data to partition $\bar{p}_3$ at a rate V_{13}.

As Fig. 5 illustrates, three partitions communicate with each other. The lines represent communication channels and the arrows show data flow direction, intuitively distinguishing between the source and destination of a channel. For example, Partition 1 has two channels. Both of their sources are Partition 1, but one destination is Partition 2 and the other is Partition 3.

The length of the MTF is defined as a product of the least common multiple (LCM) of all partition periods on one processor,

$$\mathrm{MTF} = k \cdot \mathrm{LCM}(T_{\mathrm{per}\ 1}, T_{\mathrm{per}\ 2}, \ldots, T_{\mathrm{per}\ n}), k \in \mathbf{N}^{+} \tag{4}$$

In this example, k is assigned 1. Thus, MTF is 120 ms calculated by Eq. (4). If an appropriate time slot configuration policy exists for a set of partitions, $\bar{P} = \{\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n\}$, the following must be satisfied:

$$\sum_{1 \leqslant i \leqslant n} D_i / T_{\mathrm{per}\ i} \leqslant 1 \tag{5}$$

According to Eq. (5)

$$\sum_{1 \leqslant i \leqslant 3} D_i / T_{\mathrm{per}\ i} = \frac{4}{20} + \frac{2}{30} + \frac{3}{40} \approx 0.342 < 1$$

The timing requirement is satisfied. This MTF structure of these partitions is shown in Fig. 6. Within this MTF, partitions $\bar{p}_1$, $\bar{p}_2$, and $\bar{p}_3$ are invoked six, four, and three times respectively. The three partitions are invoked properly in a time domain as seen in Fig. 6.

To facilitate modeling of partition temporal and spatial resource configurations, the following rules are proposed:

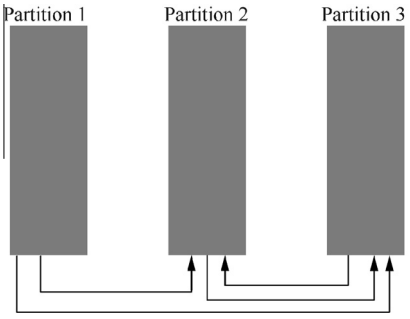

Fig. 5 Partition configuration.

Table 4 Partition parameter set.

Partition	Period (ms)	Duration (ms)	Offset (ms)	Channel	Memory size	Data generation rate
$\bar{p}_1$	20	4	0	ch12: $\bar{p}_1 \rightarrow \bar{p}_2$ ch13: $\bar{p}_1 \rightarrow \bar{p}_3$	M_{12} M_{13}	V_{12} V_{13}
$\bar{p}_2$	30	2	4	ch23: $\bar{p}_2 \rightarrow \bar{p}_3$	M_{23}	V_{23}
$\bar{p}_3$	40	3	8	ch32: $\bar{p}_3 \rightarrow \bar{p}_2$	M_{32}	V_{32}

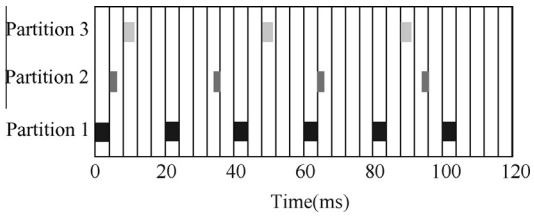

Fig. 6 A major time frame (MTF).

(1) The places represent states of system behaviors that can be divided into two parts. The first is state of system scheduling, including Inactive, Dispatched, Compute, and Finish. The second is state of system communications, including Unused, Send, and Arrival. Defined color sets indicate the resource type of each place (i.e. partition or message) and colored tokens describe specific resource information in a place. The time stamp of each token in one place represents the time when resources arrive in this place, and also indicates the earliest time the token can be used. Initial function I defines initial time stamp invoked in place, which represents the offset of partitions (i.e. O_1, O_2, O_3).

(2) Directed Arcs connecting places and transitions represent flow direction of tokens. An arc expression defines resource types that can go through channels. A delay time (the @+ operator) in arc expression represents the time interval for its next invocation in a corresponding partition (i.e. partition invocation period, $T_{\text{per }1}, T_{\text{per }2}, T_{\text{per }3}$).

(3) Transitions represent behaviors that impel state changes. A transition duration (the @+ operator) denotes partition execution duration D_i.

(4) The time–space coupling safety constraint (3) is added by guarding functions into the transition that denotes communication behavior.

A graphic model for an IMA system is shown in Fig. 7 based on the proposed rules of modeling. The four types of

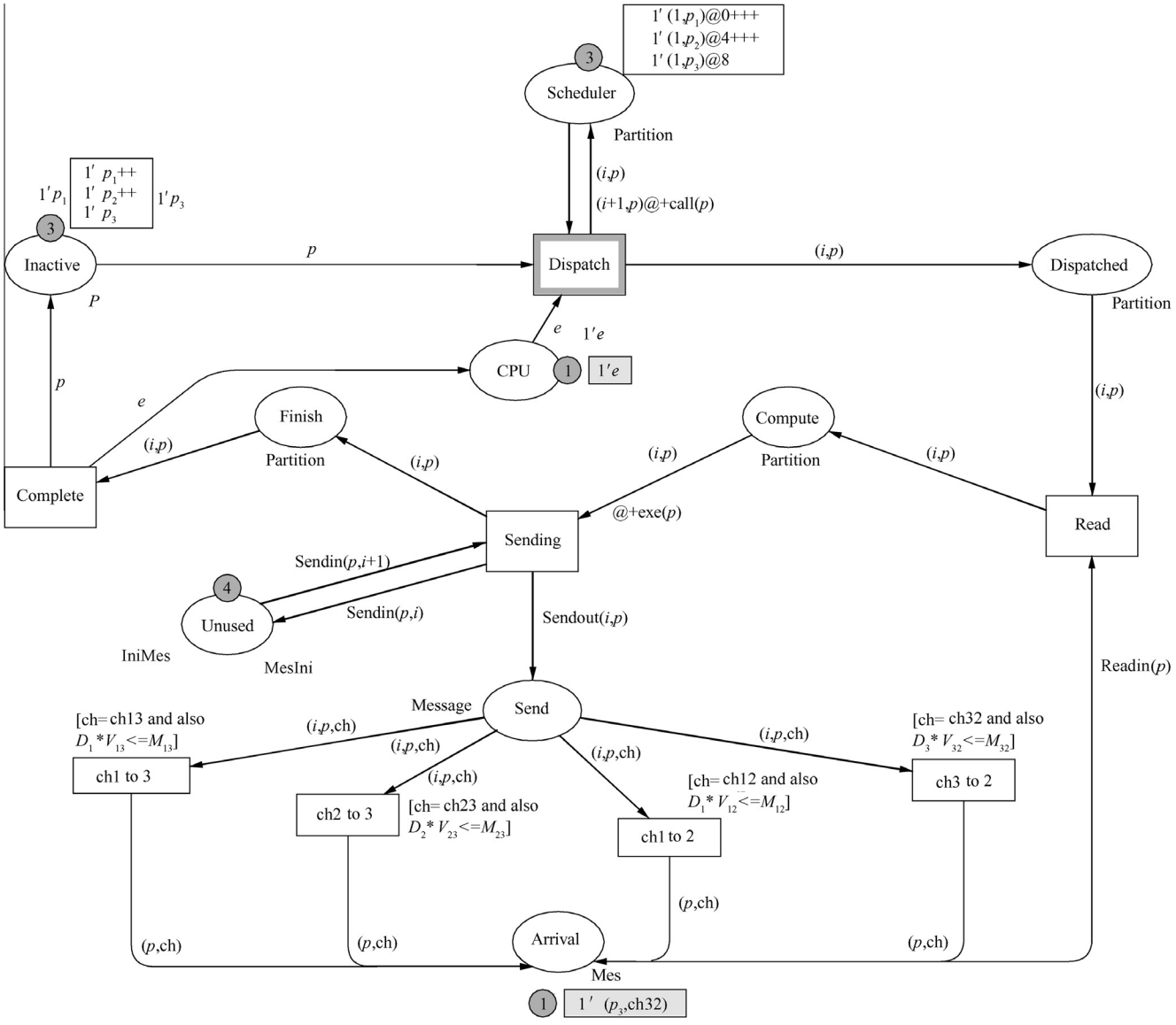

Fig. 7 A partition configuration model.

elements in a TCCP-NET are: places (drawn as ellipses), transitions (drawn as rectangular boxes), arcs (drawn as directed lines), and tokens (drawn in rounded rectangles).It has thirteen places and eight transitions, and can be divided into upper and lower parts. The upper part refers to the temporal resources of each partition and the lower part describes inter-partition communication behavior. The time–space coupling safety constraint is added into transitions as a guarding function. The color set, which is marked around each place, expresses the type of tokens in the place. The color set is defined as a group of the form $\{\langle i, p, \text{ch}, x\rangle, \langle i, p, x\rangle\}$. Color symbol p, an enumerated type, represents individual partition $\bar{p}_k$ in the IMA system. Color symbol i, an integral type, denotes the number of times partition $\bar{p}_k$ has been invoked. The time of arrival at each intermediate state of partition $\bar{p}_k$ is recorded by color symbol x, a real type. Color symbol ch represents the channel in which data can be transmitted. The token in color set Partition represents the number of invocations and current time of partition $\bar{p}_k$, while the token in color set Mestime records the arrival time of the message generated by the ith invocation of partition $\bar{p}_k$. Diverse tokens in a place represent different types of resources. Transitions with a time function represent the time required for transition between states, and the binding elements in each transition only allow particular tokens to be passed. The time–space coupling safety constraint (3) is added to guarding functions as a boolean expression, such as $G(\text{ch1 to 3}) : D_1 \times V_{13} \leqslant M_{13}$. Transitions with bolded marks indicate that these transitions can be fired at the current time.

Initially there are three partitions ready for invocation, represented by three tokens (p_1, p_2, and p_3) in place Inactive. The place Invocator, which has three tokens with different time stamps, activates invocations of the three partitions. For instance, the token with color (1, p_1) and time stamp 0 denotes that partition $\bar{p}_1$ is to be invoked when the system clock is at 0 and it is time for the first invocation. Once activated, a partition receives data as input from place Arrival and then produces data in place Send as output after computation. Guarding functions are added into transitions that connect place Send and place Arrival. Arc expressions, and binding elements restrict the tokens that go through these transitions. Transition ch1 to 3, for instance, has the binding element ch = ch13, which denotes that only the tokens that contain the color ch_{13} are allowed to pass. In addition, tokens are further restricted in terms of data size by the time–space coupling safety constraint, $D_1 \times V_{13} \leqslant M_{13}$ added into transition ch1 to 3, as guarding function. Places with color set MesTime record all data used to analyze the time–space coupling hazards in communication behaviors including arriving time, arrival times, and data size. A partition that is invoked and running on a platform has sole access to computing resources. Thus a place with color set CPU is introduced to represent computing resources, which can help verify correctness of temporal resources for each partition.

5. A case study of TCCP-NET

5.1. Case description

Here, a case study on an IMA system will be introduced. Aircraft sensor management (ASM) regulates and directs airborne sensors, including processing of millimeter-wave radar, infrared and other sensor state information, cockpit controls, and commands to relevant sensors. ASM, with limited sensor resources, can satisfy multiple demands for targeting and scanning of space to obtain the optimum value of specific features (such as the detection probability, probability of intercept, track precision or loss probability), greatly reducing both the physical and psychological burden of pilots. ASM systems consist of five applications and each application runs on an individual partition. There are many data and control signal flow transitions between the five applications. Fig. 8 shows an intuitive ASM structure, including ports, channels, and data flow directions. In Fig. 8, partition $\bar{p}_1$ receives sensor data from partitions $\bar{p}_2$ and $\bar{p}_3$, and then sends status data to the bus. Partition $\bar{p}_4$ sends control data to partition $\bar{p}_2$ and partition $\bar{p}_3$. Partition $\bar{p}_5$ receives sensor data from partitions $\bar{p}_2$ and $\bar{p}_3$, and then sends target detection data to the bus.

In configuration tables, every application is pre-assigned to an appropriate temporal resource and spatial resource, and allocated resources cannot be changed at runtime. For example, invocation period $T_{\text{per }1}$ and execution duration D_1 of partition $\bar{p}_1$ are set to 100 ms and 20 ms respectively. The port, which transfers data from partition $\bar{p}_1$ to the bus at the average speed of data generation V_1, is assigned to port memory size M_1. Resource allocation of all partitions is given in Table 5.

In the ideal case, each partition is expected to run without interruption or preemption to reduce time spent repeatedly switching between application pending and reuse. For limited allocated temporal resources, applications cannot be finished at once, but have to be divided into several slots to meet real-time requirements. In this case, the duration of partition $\bar{p}_1$ is divided into two 10 ms slots. Offset allocation to time windows for the five partitions is shown in Table 6.

The Gantt chart in Fig. 9 is drawn to describe partition schedulability according to allocated temporal resources of each partition. Partition $\bar{p}_1$ has two slots for each period and four slots within the major time frame. Partitions $\bar{p}_2$, $\bar{p}_3$ and $\bar{p}_4$ have two slots and partition $\bar{p}_5$ has one slot within the major time frame.

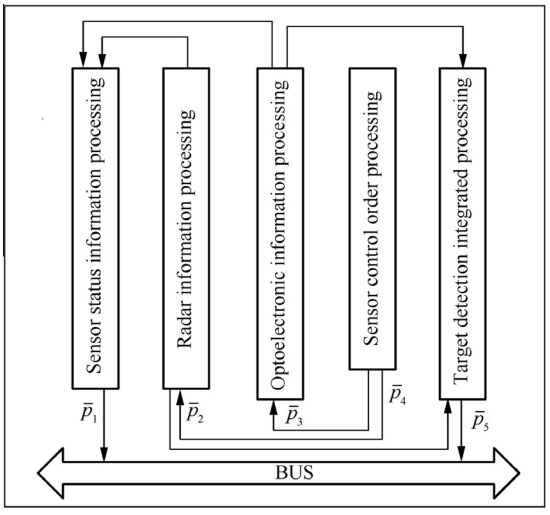

Fig. 8 ASM structure.

Table 5 Resource allocation for ASM.

Partition	Period (10^{-3} s)	Duration (10^{-3} s)	Channel	Memory size (MB)	Data generation rate (10^3 MB/s)
$\bar{p}_1$	100	20	ch1B: $\bar{p}_1 \rightarrow$ BUS	$M_1 = 10$	$V_1 = 0.45$
$\bar{p}_2$	100	10	ch21: $\bar{p}_2 \rightarrow \bar{p}_1$ ch25: $\bar{p}_2 \rightarrow \bar{p}_5$	$M_{21} = 4$ $M_{25} = 4$	$V_{21} = 0.15$ $V_{25} = 0.20$
$\bar{p}_3$	100	30	ch31: $\bar{p}_3 \rightarrow \bar{p}_1$ ch35: $\bar{p}_3 \rightarrow \bar{p}_5$	$M_{31} = 16$ $M_{35} = 16$	$V_{31} = 0.50$ $V_{35} = 0.65$
$\bar{p}_4$	100	20	ch42: $\bar{p}_4 \rightarrow \bar{p}_2$ ch43: $\bar{p}_4 \rightarrow \bar{p}_3$	$M_{42} = 6$ $M_{43} = 6$	$V_{42} = 0.20$ $V_{43} = 0.25$

Table 6 Specific time slots for each partition.

Partition	Offset(ms)	Duration(ms)
$\bar{p}_4$	0	20
$\bar{p}_1$	20	10
$\bar{p}_2$	30	10
$\bar{p}_3$	40	30
$\bar{p}_1$	70	10
$\bar{p}_4$	100	20
$\bar{p}_1$	120	10
$\bar{p}_2$	130	10
$\bar{p}_3$	140	30
$\bar{p}_1$	170	10
$\bar{p}_5$	180	20

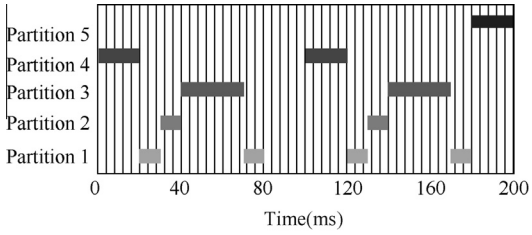

Fig. 9 An MTF for ASM.

5.2. Modeling and simulation

After an IMA function specification is given, resource scheduling in a temporal domain should be analyzed first. According to Eq. (5), the precondition to be schedulable is:

$$\sum_{1 \leqslant i \leqslant 3} D_i / T_{\text{per } i} = \frac{10}{60} + \frac{5}{50} + \frac{5}{50} + \frac{4}{40} + \frac{10}{60} \approx 0.633 < 1$$

This partition time allocation satisfies the precondition for feasible partition scheduling. Using existing scheduling analysis tools[29,30] for integrated modular avionics systems, these partition applications can be scheduled easily.

Following the proposed modeling rules, we can use the TCCP-NET to model this IMA function according to specifications. Here, the TCCP-NET model is built up as shown in Fig. 10. From the TCCP-NET model, transitions in bold indicate that their corresponding tokens can fire. The time–space coupling safety constraint from Eq. (3) is added as guarding function in corresponding transitions. In the upper part of this model, places refer to system intermediate states, including Inactive, Dispatched, Compute, Suspend, Run and Finish. An additional place, Suspend, is used to describe the mechanism of suspend and resume. The color set of these places is defined as a group of the form $\{\langle i, p, x\rangle\}$. Color symbol p (p_1, p_2, p_3, p_4, p_5) represents different partitions $\bar{p}_k$ ($\bar{p}_1$, $\bar{p}_2$, $\bar{p}_3$, $\bar{p}_4$, $\bar{p}_5$) in the ASM system and color symbol i denotes the number of times partition $\bar{p}_k$ has been invoked. The time of arrival at each intermediate states of partition $\bar{p}_k$ is recorded by color symbol x. In the lower part of this model, places describe the states where data is generated, sent and received. The color set of these places is defined as a group of the form $\{\langle i, p, \text{ch}, x\rangle\}$. Color symbol ch limits the channel used to transmit data. The places Unused and MesTime record arguments that data is generated and received. Initially there are five tokens in place Inactive and eight tokens in place Unused.

Variable types in this model, which denote different types of resources in each partition, are defined by color sets and marked around the places. All of the main parameters in this model are given in Table 7.

When we finish modeling a real-time system function by transforming its specifications along with the time–space coupling safety constraint into a TCC-PN model, the next step is to simulate the reachability of the model in order to verify whether there exist time–space coupling hazards at runtime for given resource allocations (temporal resource and spatial resource). Because all the released partition invocations within each MTF have finished before its next MTF, allocation of time slots in each MTF is exactly the same. Time–space coupling hazard analysis can be conducted only in one MTF. The length of the MTF is 200 ms, as calculated by Eq. (4).

$$\text{MTF} = 1 \cdot \text{LCM}(100, 100, 100, 100, 200) = 200 \text{ ms}$$

The amount of simulation time is set to 200 ms and the remaining constant values (port memory size M_{ij} and data generation rate V_{ij}) in the TCCP-NET model are given in Table 8.

5.3. Simulation results

For this TCCP-NET model, simulation experiments were performed on CPN Tools,[37,18] and system marking after simulation execution for 200 ms are shown in Fig. 11. The model in Fig. 11 is very similar to that in Fig. 10 because one model shows the end state of the first MTF and while the other displays the initial state of the second MTF. But, tokens in place for the two models have changed. The attributes of each place consist of token type, the number of tokens, and time stamp (message arrival time). All token records are shown in Table 10. Tokens in places P_1, P_2, P_3, P_5, and BUS record information

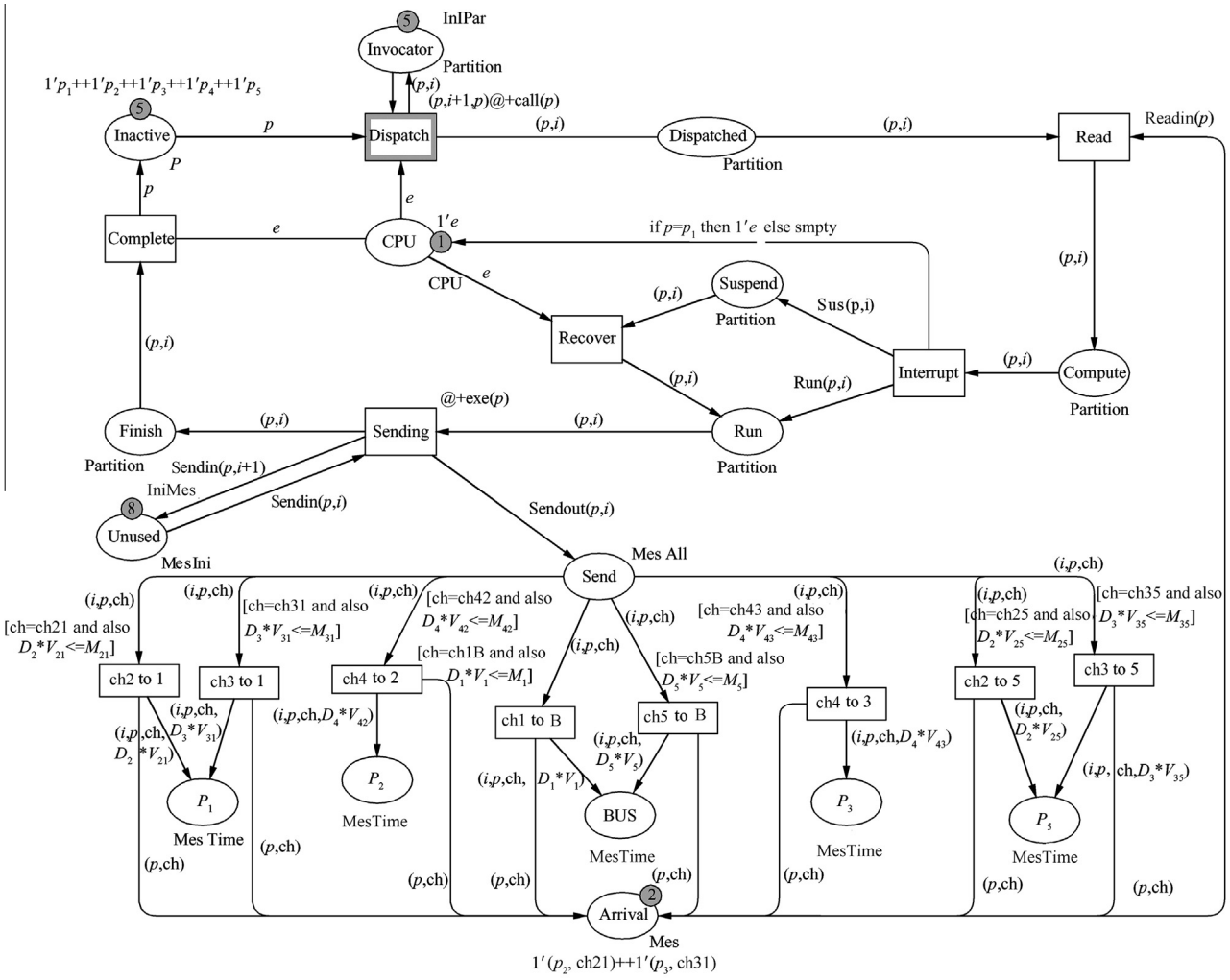

Fig. 10 A TCCP-NET model for ASM.

Table 7 Declaration of color set and parameters.

Colset P = with p_1 | p_2 | p_3 | p_4| p_5
Closet CH = with BUS | ch42 | ch43 | ch21 | ch31 | ch35 | ch25 | ch1B | ch1B
Colset Partition = product INT * P timed;
Colset Mes = product P * CH;
Colset MesAll = product INT * P * CH timed;
Colset MesTime = product INT * P * CH * INT timed;
Colset CPU = with e;
Val IniMes =
1′(0, p_4, ch42)++1′(0, p_4, ch43)++1′(0, p_2, ch21)++1′(0, p_3, ch31)
++1′(0, p_3, ch35)++1′(0, p_2, ch25)++1′(0, p1, ch1B)++1′(0, p_5, ch5B);
Val IniPar =
1′(0, p_1)@20++1′(0, p_2)@30++1′(0, p_3)@40++1′(0, p_4)@0
++1′(0, p_5)@180;
Guarding functions
$G_1 = D_2 * V_{21} <= M_{21}$; $G_2 = D_3 * V_{31} <= M_{31}$;
$G_3 = D_4 * V_{42} <= M_{42}$; $G_4 = D_1 * V_1 <= M_1$;
$G_5 = D_5 * V_5 <= M_5$; $G_6 = D_4 * V_{43} <= M_{43}$;
$G_7 = D_2 * V_{25} <= M_{25}$; $G_8 = D_3 * V_{35} <= M_{35}$;
Var *p*: P Var ch: CH Var *i*: INT

Table 8 Constant values in TCCP-NET model.

Memory size(MB)	Data generation rate(10^3 MB/s)
$M_1 = 10$	Val $V_1 = 0.45$
$M_{21} = 4$	Val $V_{21} = 0.15$
$M_{25} = 4$	Val $V_{25} = 0.20$
$M_{31} = 16$	Val $V_{31} = 0.50$
$M_{35} = 16$	Val $V_{35} = 0.65$
$M_{42} = 6$	Val $V_{42} = 0.20$
$M_{43} = 6$	Val $V_{43} = 0.25$
$M_5 = 10$	Val $V_5 = 0.65$

about data that is successfully received. For example, P_1 has 2 tokens with color $(p_2, \text{ch21})$ transferred from P_2 via the transition ch2 to 1, and 2 tokens with color $(p_3, \text{ch31})$ transferred from P_3 via the transition ch3 to 1. The arrival time of the two tokens with color $(p_2, \text{ch21})$ are 40 and 140 respectively. Tokens in place unused record the number of times data has been sent. Tokens in place scheduler record the number of times partitions have been successfully invoked. Partition $\bar{p}_5$, for instance, was successfully invoked once and then sent data to the BUS once. In addition, in an MTF, the number of

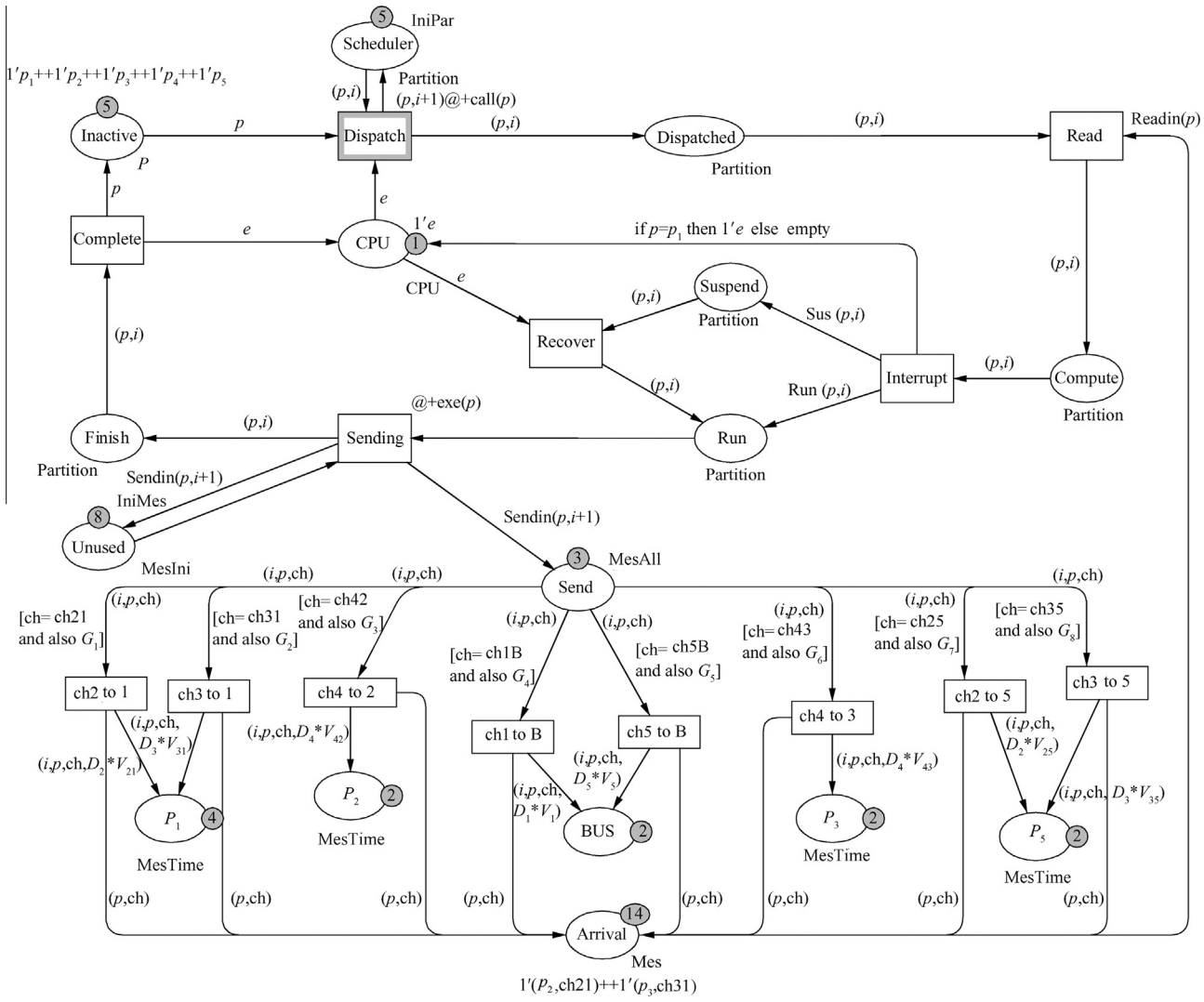

Fig. 11 System marking after simulation execution.

partition invocations is easily calculated and invocation times of each partition are listed in Table 9. For example, partition $\bar{p}_1$ had been invoked and then sent data to the BUS 2 times. So place BUS should have 2 tokens with color (p_1, BUS).

By comparing token numbers in places Unused and Scheduler in Table 10 and invocation times of each partition in Table 9, it can be concluded that all of partitions were scheduled as expected and data were successfully sent from each partition. It also verifies the correctness of temporal resource allocation. On this basis, token attributes in places P_1, P_2, P_3, P_5, and Bus are used to analyze time–space coupling hazards. For example, partition $\bar{p}_2$ sent data to partitions $\bar{p}_1$ and $\bar{p}_5$ via two separate channels 2 times in an MTF. Partition $\bar{p}_1$ has 2 tokens with color (p_2,ch21) and partition $\bar{p}_5$ has 2 tokens with color (p_2,ch25). Data were successfully transmitted from partition $\bar{p}_2$ to partitions $\bar{p}_1$ and $\bar{p}_5$. It can be concluded that resource allocation (temporal resources and spatial resources) of partition $\bar{p}_2$ was appropriate. The other four partitions can be analyzed in the same way. Results show that resource allocations of partitions $\bar{p}_1$ and $\bar{p}_4$ were appropriate, but the resource allocations of partitions $\bar{p}_3$ and $\bar{p}_5$ were inadequate. Partition $\bar{p}_3$ had two channels and sent data to partition $\bar{p}_1$ and $\bar{p}_5$ respectively 2 times in an MTF. Partition $\bar{p}_1$ had 2 tokens with color (p_3,ch31) and partition $\bar{p}_5$ had 0 tokens with color (p_3,ch35). Partition $\bar{p}_1$ received a correct number of data. Data were successfully transmitted from partition $\bar{p}_3$ to partition $\bar{p}_1$ via the channel ($\bar{p}_3 \rightarrow \bar{p}_1$). However, partition $\bar{p}_5$ could not receive messages transmitted from partition $\bar{p}_3$ via the channel ($\bar{p}_3 \rightarrow \bar{p}_5$). Messages were not successfully transmitted

Table 9 Number of partition invocations.

Partition	Period(ms)	Invocation times	Channel
$\bar{p}_1$	100	2	ch1B: $\bar{p}_1 \rightarrow$ BUS
$\bar{p}_2$	100	2	ch21: $\bar{p}_2 \rightarrow \bar{p}_1$ ch25: $\bar{p}_2 \rightarrow \bar{p}_5$
$\bar{p}_3$	100	2	ch31: $\bar{p}_3 \rightarrow \bar{p}_1$ ch35: $\bar{p}_3 \rightarrow \bar{p}_5$
$\bar{p}_4$	100	2	ch42: $\bar{p}_4 \rightarrow \bar{p}_2$ ch43: $\bar{p}_4 \rightarrow \bar{p}_3$
$\bar{p}_5$	200	1	ch5B: $\bar{p}_5 \rightarrow$ BUS

Table 10 Simulation results for each place.

Place	Token type	Token number i	Time stamp
P^1	(p_2,ch21)	2	@40,140
	(p_3,ch31)	2	@70,170
P_2	(p_4,ch42)	2	@20,120
P_3	(p_4,ch43)	2	@20,120
P_5	(p_2,ch25)	2	@40,140
BUS	(p_1,ch1B)	2	@80,180
Unused	(p_5,ch5B)	1	
	(p_1,ch1B)	2	
	(p_2,ch21)	2	
	(p_2,ch25)	2	
	(p_3,ch31)	2	
	(p_3,ch35)	2	
	(p_4,ch42)	2	
	(p_4,ch43)	2	
Scheduler	p_1	2	
	p_2	2	
	p_3	2	
	p_4	2	
	p_5	1	

from partition $\bar{p}_3$ to partition $\bar{p}_5$ via the channel ($\bar{p}_3 \rightarrow \bar{p}_5$) and these data were lost since resource allocation of partition $\bar{p}_3$ did not satisfy the time–space coupling safety constraint (i.e. G_8 was not satisfied. The formula, $D_3 \times V_{35} <= M_{35}$ did not work). Partition $\bar{p}_5$ had only one channel and sent data to the bus one time in an MTF. The place BUS has 0 tokens with color (p_5,ch5B). The bus could not receive data transmitted from partition $\bar{p}_5$ via the channel ($\bar{p}_5 \rightarrow$ BUS). Resource allocation of partition $\bar{p}_5$ did not satisfy the time–space coupling safety constraint (i.e. G_5 was not satisfied).

5.4. Discussion

Through analysis of the simulation results, it can be concluded that the resource allocations (temporal resources and spatial resources) of partition $\bar{p}_2$, partition $\bar{p}_1$, and partition $\bar{p}_4$ are appropriate. However, resource allocations of partition $\bar{p}_3$ and partition $\bar{p}_5$ are inadequate, and a time–space coupling hazard occurs at runtime.

The technique proposed in Ref.[33] presents an optimal allocation policy under the assumption of adequate spatial resources. It can validate temporal resource allocations (time slot allocations of partitions) for ASM by simulating the time slot allocation within one MTF. A constraint-based allocation approach[34] further addresses avionics temporal resource allocation implementation with practical constraints. Still, the impact of temporal resources on spatial resource requirements is never taken into consideration, and resource coupling hazards in the time–space domain can barely be detected by existing methods.

Although temporal resource allocations for ASM have proven to be feasible and all partitions can be scheduled properly, due to insufficient spatial resource allocation, some underlying problems still occur. These issues usually cause data loss and communication obstruction, which seriously affect the integrity of safety-critical software systems. To solve this problem,

Table 11 Simulation results for each place after modifications.

Place	Token type	Token number (i)	Time stamp
P^1	(p_2,ch21)	2	@40,140
	(p_3,ch31)	2	@70,170
P_2	(p_4,ch42)	2	@20,120
P_3	(p_4,ch43)	2	@20,120
P^5	(p_2,ch25)	2	@40,140
	(p_3,ch35)	2	@70,170
BUS	(p_1,ch1B)	2	@80,180
	(p_5,ch5B)	1	@200
Unused	(p_5,ch5B)	1	
	(p_1,ch1B)	2	
	(p_2,ch21)	2	
	(p_2,ch25)	2	
	(p_3,ch31)	2	
	(p_3,ch35)	2	
	(p_4,ch42)	2	
	(p_4,ch43)	2	
Scheduler	p_1	2	
	p_2	2	
	p_3	2	
	p_4	2	
	p_5	1	

we propose using a time–space coupling safety constraint then employing a TCCP-NET to improve hazard analysis. Two solutions emerge: If sufficient spatial resources are available, system specifications can be modified to satisfy the requirements at runtime. If spatial resources are inadequate or limited, the amount of generated data should be reduced (e.g. shortened duration of partition execution) to reduce spatial resource requirements.

In this case of ASM, the first solution is employed, which increases corresponding spatial resources. Memory size M_5 and M_{35} in Table 8 are adjusted. The values of M_5 and M_{35} are both set as 20 M. System marking after simulation execution for 200 ms is shown in Table 11.

An identical analysis process was conducted once again. Partition $\bar{p}_5$ has 2 tokens with color (p_3,ch35), which indicates that partition $\bar{p}_5$ successfully received messages transmitted form partition $\bar{p}_3$ via the channel ($\bar{p}_3 \rightarrow \bar{p}_5$). Resource allocation of partition $\bar{p}_5$ was appropriate. The results show that all of the partitions can be scheduled properly, and messages can be successfully transmitted from source to destinations via the corresponding channels. It can be concluded that both temporal resource and spatial resource allocations for ASM are suitable and no time–space coupling hazards exist.

6. Conclusion and future work

In this paper, a time–space coupling safety constraint and a new TCCP-NET modeling method are proposed for safety-critical real-time systems with partitioning in time and space domains. Current configuration methods focus only on temporal resources or spatial resources independently, which limits the analysis of time–space dynamic connection requirements. Time–space coupling hazard analysis is conducted in three

steps: specification modeling, simulation execution, and results analysis. The process of modeling and analysis using a TCCP-NET is demonstrated in detail through the modeling of a safety-critical function and analysis of underlying time–space coupling hazards. The results of an ASM case study show the feasibility and effectiveness of this approach.

The TCCP-NET modeling and analysis method can be applied to other real-time systems that contain additional safety constraints (temporal and spatial) on components and system behaviors. However, when an established model becomes large and complex, state space size increases dramatically. A corresponding state space explosion would impede model analysis capability. Therefore, while ongoing work moves toward expanding potential applicability by adding more safety constraints into the TCCP-NET model, it should also strive to simplify the model and minimize state space size. Finally, a software analysis tool shall be developed for more users.

Acknowledgements

This work was supported by grants from the National Basic Research Program of China (No. 2014CB744904) and the National Natural Science Foundation of China (No. 61300069).

References

1. Spitzer CR, Ferrell U, Ferrell T, Prisaznuk PJ. *ARINC specification 653, avionics application software standard interface*. Boca Raton: CRC Press; 2014. p. 625–32.
2. Watkins CB, Walter R. Transitioning from federated avionics architectures to integrated modular avionics. In: *AIAA/IEEE digital avionics systems conference*; 2007 Oct 21–25; Dallas, USA. Piscataway, NJ: IEEE Press; 2009. p. 2.A.1-1-10.
3. Shortle J, Zhang YM. Safety comparison of centralized and distributed aircraft separation assurance concepts. *IEEE Trans Reliab* 2014;**63**(1):259–69.
4. Ammar HH, Nikzadeh T, Dugan JB. Risk assessment of software-system specifications. *IEEE Trans Reliab* 2001;**50**(2):171–83.
5. Billington J, Christensen S, Van Hee K, Kindler E, et al. *The petri net markup language: concepts, technology, and tools*. Berlin: Springer; 2003. p. 483–505.
6. David R, Alla H. Petri nets for modeling of dynamic systems: a survey. *Automatica* 1994;**30**(2):175–202.
7. Wang J. *Timed petri nets: theory and application*. Berlin: Springer Science & Business Media; 2012. p. 63–123.
8. Jensen K. *Coloured petri nets: basic concepts, analysis methods and practical use*. Berlin: Springer Science & Business Media; 2013. p. 1–19.
9. Jensen K, Rozenberg G. *High-level petri nets: theory and application*. Berlin: Springer Science & Business Media; 2012. p. 215–43.
10. Murata T. Petri nets: properties, analysis and applications. *Proc IEEE* 1989;**77**(4):541–80.
11. Zurawski R, Zhou MC. Petri nets and industrial applications: a tutorial. *IEEE Trans Industr Electron* 1994;**41**(6):567–83.
12. Ze T, Xie LY, Liang D. Controller design of DES Petri nets with mixed constraint. *Chin J Aeronaut* 2005;**18**(3):283–8.
13. Zuberek WM. Timed petri nets definitions, properties, and applications. *Microelectron Reliab* 1991;**31**(4):627–44.
14. Berthomieu B, Diaz M. Modeling and verification of time dependent systems using time petri nets. *IEEE Trans Software Eng* 1991;**17**(3):259.
15. Silva JR, del Foyo PMG. *Timed petri nets*. Rijeka: INTECH Open Access Publisher; 2012. p. 359–78.
16. Malhotra M, Trivedi KS. Power-hierarchy of dependability-model types. *IEEE Trans Reliab* 1994;**43**(3):493–502.
17. Jensen K. Coloured petri nets*Discrete event systems: a new challenge for intelligent control systems, IEE colloquium on, London*. p. 5/1–3.
18. Jensen K, Kristensen LM, Wells L. Coloured petri nets and CPN tools for modelling and validation of concurrent systems. *Int J Softw Tools Technol Transfer* 2007;**9**(3–4):213–54.
19. Tsai JJP, Yang SJ, Chang YH. Timing constraint petri nets and their application to schedulability analysis of real-time system specifications. *IEEE Trans Software Eng* 1995;**21**(1):32–49.
20. Santos S, Rufino J, Schoofs T, Tatibana C. A portable ARINC standard interface. In: *Digital avionics systems conference*; 2008 Oct 26–30; Crowne Plaza St., USA. Piscataway, NJ: IEEE Press; 2008. p. E.2-1-7.
21. RTCA (Firme). *Integrated modular avionics (IMA) development guidance and certification considerations*. Washington D.C.: RTCA Inc.; 2005.
22. Delange J, Gilles O, Hugues J, Pautet L. Model-based engineering for the development of ARINC653 architectures. *SAE Int J Aerospace* 2010;**3**(1):79–86.
23. Windsor J, Hjortnaes K. Time and space partitioning in spacecraft avionics. In: *The third IEEE international conference on space mission challenges for information technology*; 2009 July 19-23; Pasadena, USA. Piscataway, NJ: IEEE Press; 2009. p. 13–20.
24. Littlefield-Lawwill J, Kinnan L. System considerations for robust time and space partitioning in integrated modular avionics. In: *Digital avionics systems conference*; 2008 Oct 26–30; St. Paul, USA. Piscataway, NJ: IEEE Press; 2008. p. B.1–B.11.
25. Prisaznuk PJ. ARINC 653 role in integrated modular avionics. In: *Digital avionics systems conference*; 2008 Oct. 26–30; Crowne Plaza St., USA. Piscataway, NJ: IEEE Press; 2008. p. 1.E.5-1-10.
26. Watkins CB, Walter R. Transitioning from federated avionics architectures to integrated modular avionics. In: *Digital avionics systems conference*; 2007 Oct. 21–25; Dallas, USA. Piscataway, NJ: IEEE Press; 2007. p. 2.A.1-1-10.
27. ARINC specification 653: Part 2. Avionics application software standard interface, extended services [Internet]. [2014-11-02]. Available from <https://www.arinc.com/cf/store/index>.
28. Sha L, Abdelzaher T, Årzén KE, Cervin A, Baker T, Burns A, et al. Real time scheduling theory: a historical perspective. *Real-Time Syst* 2004;**28**(2–3):101–55.
29. Zhou TR, Xiong HG. Design of energy-efficient hierarchical scheduling for integrated modular avionics systems. *Chin J Aeronaut* 2012;**25**(1):109–14.
30. Stankovic JA, Spuri M, Ramamritham K, Buttazzo GC. *Deadline scheduling for real-time systems*. Boston, MA: Springer, US; 1998.
31. Leung JYT. A new algorithm for scheduling periodic, real-time tasks. *Algorithmica* 1989;**4**(1–4):209–19.
32. Lee YH, Kim D, Younis M, Zhou J. Scheduling tool and algorithm for integrated modular avionics systems. In: *Digital avionics systems conference*; 2000 Oct. 07–13; Philadelphia, USA. Piscataway, NJ: IEEE Press; 2000. p. 1C2/1-2/8.
33. Gui SL, Luo L, Tang SS, Meng Y. Optimal static partition configuration in ARINC653 system. *J Electron Sci Technol* 2011;**9**(4):373–8.
34. Sagaspe L, Bieber P. Constraint-based design and allocation of shared avionics resources. In: *Digital avionics systems conference*; 2007 Oct. 21–25; Dallas, USA. Piscataway, NJ: IEEE Press; 2007. p. 2. A.5-1-10.
35. Fleming CH, Leveson NG. Improving hazard analysis and certification of integrated modular avionics. *J Aerospace Inform Syst* 2014;**11**(6):397–411.
36. Al Sheikh A, Brun O, Hladik PE. Partition scheduling on an IMA platform with strict periodicity and communication delays. In: *18th international conference on real-time and network systems*; 2010 Nov 4–5; Toulouse, France. NewYork: ACM Inc.; 2010. p. 179–88.

37. Ratzer AV, Wells L, Lassen HM, Laursen M, Qvortrup JF, Stissing MS, et al. *Applications and theory of petri nets 2003*. Berlin: Springer; 2003. p. 450–62.

Li Zelin was born in Shanxi Province in 1991. He received the B.S. degree from Beijing Institute of Technology in 2013, and is currently pursuing an M.S. degree in reliability and systems engineering from Beihang University. His research interests include software reliability and safety, and embedded real-time operating systems modeling.

Wang Shihai received his Ph.D. in computer science from the University of Manchester, UK in 2010. He joined the School of Reliability and Systems Engineering, Science and Technology on Reliability and Environmental Engineering Laboratory, Beihang University, as a lecturer in 2011. Currently, his research interests include software testing, software fault prediction and pattern recognition and applications in software reliability.

Zhao Tingdi is a professor at the School of Reliability and Systems Engineering, Beihang University. He received his Ph.D. degree in aircraft design from Beihang University. His research activities are focused on system reliability, system safety and systems engineering.

Liu Bin is a research professor at the School of Reliability and Systems Engineering, Beihang University. He received his Ph.D. in aircraft design from Beihang University, and has been engaged in teaching, research and management services in the fields of software engineering and software reliability engineering. His main research focus includes software reliability and software testability.