第 47 卷第 3 期
2015 年 6 月

南 京 航 空 航 天 大 学 学 报
Journal of Nanjing University of Aeronautics & Astronautics

Vol. 47 No. 3
Jun. 2015

DOI:10.16356/j.1005-2615.2015.03.014

基于模型转换的 IMA 系统可调度性验证方法

胡 军[1,2] 程 桢[1] 马金晶[1] 刘 雪[1] 石姣洁[1]

(1.南京航空航天大学计算机科学与技术学院,南京,210016;
2.南京大学计算机软件新技术国家重点实验室,南京,210093)

摘要:综合模块化航空电子系统(Integrated modular avionics ,IMA)中分区运行时间特征满足需求是 IMA 系统安全可靠性运行的一个重要问题。本文针对满足 ARINC653 规范的 IMA 系统的层级调度特性,结合 IMA 系统调度配置信息,提出了一种在模型驱动工程(Model driven engineering ,MDE)框架下,基于实时嵌入式系统建模与分析(Modeling and analysis of real-time and embedded system,MARTE)模型的 ARINC653 分区调度系统建模转换与可调度性验证的方法。借助 MAST 工具及其自定义调度策略功能,分析分区系统调度特性并利用 MARTE 对其进行建模,并利用该工具对 MARTE 模型进行仿真以验证其可调度性,最后给出了一个实例分析。

关键词:综合模块化航电系统;可调度性验证;ARINC653;实时嵌入式系统与分析;模型驱动工程

中图分类号:TP311 **文献标志码**:A **文章编号**:1005-2615(2015)03-0403-09

Schedulable Verification Framework for IMA System Based on Model-Transformation

Hu Jun[1,2], *Cheng Zhen*[1], *Ma Jinjing*[1], *Liu Xue*[1], *Shi Jiaojie*[1]

(1. College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing, 210016, China;
2. State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210093, China)

Abstract: Ensuring the satisfiability of partition time requirement in integrated modular avionics (IMA) is of great importance for the safe and reliable operation of IMA system. A method of modeling, transformation and scheduling validation of ARINC653 partition scheduling system is put forward based on model driven engineering (MDE). The hierarchy scheduling characteristics of IMA system are analyzed combined with the IMA system scheduling configuration information, the model transformation rules are established between the modeling and analysis of real-time and embedded system (MARTE) model elements and the hierarchical scheduling semantic information, and a scheduling validation framework of IMA partition system is designed based on MARTE. Then MAST tool is used to make simulation for the MARTE model to verify the schedulability. Finally, a case analysis is given to illustrate the validity of the method.

Key words: integrated modular avionics; schedulability verification; ARINC653; modeling and analysis of real-time and embedded system (MARTE); model driven engineering

---

基金项目:国家重点基础研究发展计划("九七三"计划)(2014CB744903)资助项目;回国留学人员科研启动基金资助项目;611 航空科研基金资助项目;南京航空航天大学青年科技创新基金(NS2014098)资助项目。

收稿日期:2015-03-02;**修订日期**:2015-04-28

通信作者:胡军,男,副教授,E-mail:hujun@nuaa.edu.cn。

综合模块化航电系统(Integrated modular avionics, IMA)[1-2]是近年来安全关键应用领域中一类重要的复杂嵌入式系统,具备多个实时应用同时在计算平台上以时间/空间多分区形式运行的系统特征。针对IMA类系统的设计与分析已经成为近年来复杂嵌入式系统工程研究领域中的一个重要挑战。

IMA系统中的配置信息通常是由ARINC653规范[3-4]来给出,满足ARINC653标准的IMA系统称之为ARINC653系统。IMA系统配置信息包含了系统架构中所有抽象层面的数据信息,可用来对IMA系统中诸如硬件资源、操作系统接口以及应用程序运行环境等进行参数配置。IMA的调度信息包含在系统配置信息中,针对此类系统进行可调度性验证分析具有重要研究意义。因此,本文结合模型驱动工程的理论,使用专门为嵌入式领域而指定的MARTE建模语言,根据调度配置信息对IMA分区调度系统进行建模,并利用一套用于实时应用建模与调度性分析的开源工具MAST[5]对其进行可调度性验证分析,这项工作具有可行性和必要性。

模型驱动工程(Model driven engineering, MDE)[6-8]是近十年来在系统工程以及软件工程领域中出现的主流方法,其基本思想是以系统模型设计、模型转换与分析/验证为工程的重要核心,提高对复杂工程系统开发与维护的能力和效率。在最新版本的航空软件适航标准DO-178C中也已经正式提出了基于模型的系统开发和形式化方法的要求[9-11]。实时嵌入式系统建模与分析[12-13](Modeling and analysis of real-time and embedded system,MARTE)建模语言支持对复杂嵌入式系统设计中所需的功能建模以及广泛存在的时间约束、资源分配等非功能属性的建模与分析。MARTE是目前工业界已经得到应用的一类专门针对复杂嵌入式系统设计与分析的规范。

ARINC653分区调度系统具有两级调度(分区间调度和分区内调度)的层级性。本文根据ARINC653调度配置信息中当前分区时间的配置,确保分区内的任务能够满足调度时间约束要求,提出了一种在MDE框架下的基于MARTE模型的ARINC653分区调度系统建模转换与可调度性验证的方法,来保证IMA分区任务集的可调度性。

1 ARINC653与MARTE

1.1 ARINC653系统

满足ARINC653规范的IMA架构如图1所示。系统配置信息是ARINC653系统非常重要的组成部分,包括了以上所有层次的相关信息以及参数配置。配置信息通常包括模块级和分区级两大类,分别描述分区间和各分区内的资源配置情况。本文的研究重点在应用软件层,它按资源分配和功能将系统划分为数个模块,模块又划分为数个分区,分区内包含具体的任务进程。通过引入分区概念,将模块分为若干个分区,每个分区分配指定的内存空间和处理器时间片。

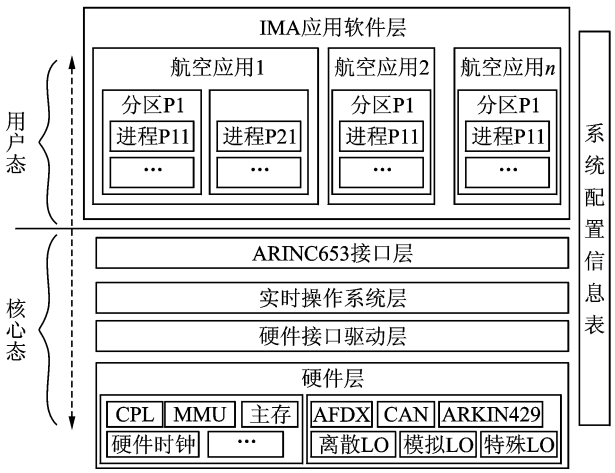

图1 ARINC653的IMA架构

Fig. 1 IMA architecture of ARINC653

在调度系统中,如果一个任务在每次到达之后都能在其截止时间限内执行完成,则判定该任务可调度,否则该任务不可调度。从而对任务可调度性的判定即对任务在截止时间期限内是否执行完成的判定。如果ARINC653分区系统中的所有任务都能在各自的截止时间限内执行完成,则判定分区系统可调度,否则,判定分区系统不可调度。本文即针对分区系统的可调度性判定方法进行研究。

1.2 MARTE

MARTE[12-13]能够对复杂嵌入式实时系统中涉及的软硬件等层面进行功能及非功能多个方面有效建模;其作为UML推荐的实时和嵌入式系统建模的正式规范,目前已经被工业界认可并使用。MARTE中的概念主要分为基础(Foundation)、建模(Modeling)和分析(Analyzing)3个部分,分别封装在基础模型、设计模型和分析模型3个包中,基本体系结构如图2所示。

MARTE所提供的丰富建模元素可以对航空安全关键嵌入式系统的配置信息进行建模,并基于所构建的模型展开进一步的分析与验证。

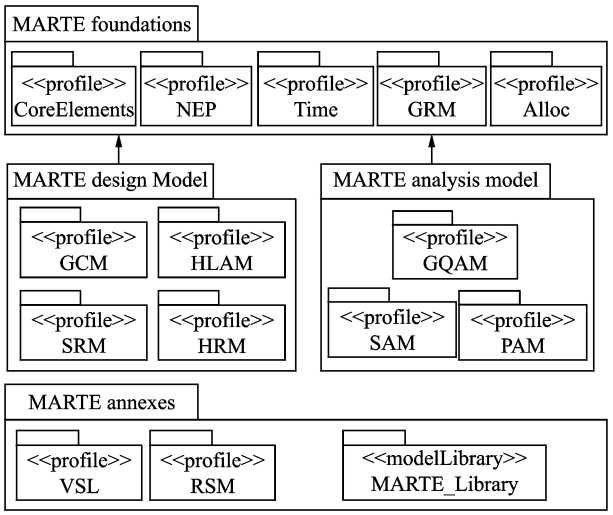

图 2　MARTE 的基本建模元素及架构

Fig. 2　Basic modeling elements and architecture of MARTE

2　IMA 调度配置信息与 MARTE 模型的转换

从 IMA 系统调度配置信息构建 MARTE 调度性分析模型首先需要建立从 ARINC653 规范中所要求的分区调度配置信息到 MARTE 中的各类组件、属性等建模元素之间的语义映射规则,然后设计相应的模型转换方法。目前本文工作的重点是对调度相关的几类核心概念(如:模块、分区、进程)展开模型转换规则及方法研究。

2.1　ARINC653 分区调度模型特征

一个航空电子系统中有若干分区模块,每个分区模块下对应着各自的任务集合。将系统时间按时间片的方式分配给各个分区,各个分区下的任务集合在相应分派的时间片下运行(各分区的时间片都有时间片偏移量(Offset)和时间片大小(Duration)这两个参数,系统周期性地为分区分派时间片),当时间片用完时暂停执行操作,直到下一个时间片的到来。和普通系统调度模型不同的是,分区调度模型有两级调度,如图 3 所示。

(1) 分区间调度

分区间调度是第一级调度,ARINC653 中的分区间调度是确定的,由模块配置信息直接严格确定,每个分区周期性地分配到处理器时间,包括可用资源分配、特殊分区需求、基于时间的活动等,通过这些确定单个分区所需调度的时间窗口,每个分区都是按照各自的时间窗口来调度的。

由于模块里的分区间调度配置信息是固定的,所以分区间调度具有以下几点特性:对应用开发者来说,分区是调度的基本单元;分区之间没有优先级;分区间调度算法是固定不变的,按照一定的周期重复调度,调度周期由主时间片确定,并且调度算法也仅仅是由系统配置信息来决定的,同时在一个主时间片内,所有分区都至少分配到一个分区窗口;ARINC653 操作系统控制所有分区的处理器资源分配。

(2) 分区内调度

分区内调度是第二级调度,它指的是分区内进程间调度。分区内调度是由操作系统根据任务调度策略来对任务进程集进行时间分配。与分区间调度的不同,它是可抢占式的调度策略,比如:EDF(最早截止期优先),LLF(空闲时间优先),RMS(速率单调调度),DMS(截止时间单调调度)等。每个任务的具体分配到的时间片是无法通过配置来指定的,所以无法直接判定每个任务是否都能满足任务的可调度性。

因此,进程级调度具有以下的特征:进程调度是由 ARINC653 操作系统控制的,每个进程都有一个当前优先级;进程调度算法只基于优先级;进程调度支持周期调度和非周期调度;分区中的进程共享该进程所分配到的资源。

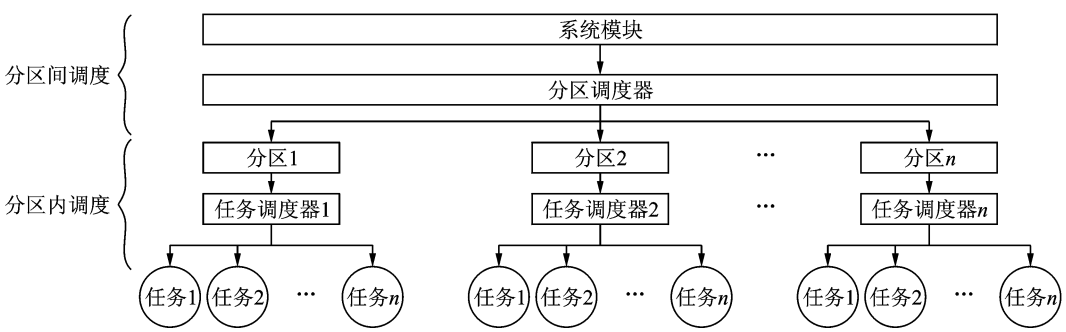

图 3　分区系统调度模型

Fig. 3　Scheduling model of partitioned system

2.2 MARTE 与 ARINC653 的转换规则

本节将重点分析 MARTE 与 ARINC653 分区系统调度配置信息的转换规则，使用 MARTE 中的 hwProcess 组件、swSchedulingResource 组件和 ProcessingResource 组件对 IMA 系统中的模块、进程、分区进行建模，在 3.2 节会介绍利用 MAST 设计自定义调度策略，将自定义策略赋值到上述组件属性中去使得 MARTE 具有了针对 ARINC653 系统所包含的分区调度信息进行建模和分析的能力，具体建模流程如图 4 所示。

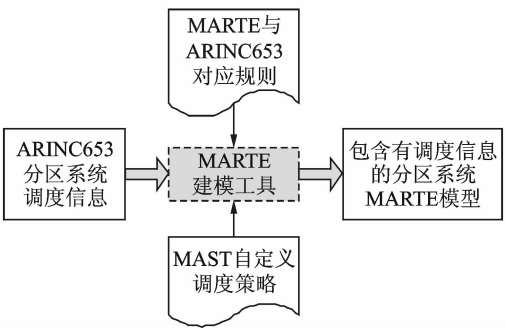

图 4 分区调度建模流程

Fig. 4 Flow of modeling partitioned scheduling

(1) 模块转换规则

ARINC653 调度配置信息中的模块(Module)包含了处理器、通信接口、内存等硬件资源分配的描述，也包括了运行在此模块上的一个或多个航电应用软件的分区信息以及每个分区所分配的系统资源和分区级的调度信息。MARTE 中的 hwProcessor 组件可以用来指明系统运行时的执行环境，包括 CPU 的调度分配、内存分配、通信连接总线等。因此，可以用 MARTE 的 hwProcessor 组件表示 ARINC653 模块概念，用 hwProcessor 所包含的 mainScheduler 属性中的 schedPolicy 来设置分区间的调度策略。由于分区系统调度的复杂性，目前还不存在一个确定的调度策略，因此需要借助 MAST 中的 Transactions 来自定义调度策略，具体自定义调度策略方法见 3.2 节。

(2) 分区转换规则

ARINC653 调度配置信息中的分区(Partition)是 IMA 系统中的一个核心概念，它在时间和空间上是隔离的，每个软件都在自己的分区中运行，不同分区任务的运行不受影响。在模块平台的总时间框架下，分区被调度的周期和运行时间已在调度配置信息分配好；同时考虑空间隔离的需求，不同的分区分配的地址空间也不同。可以用 MARTE 中的 swSchedulingResource 和 ProcessingResource 组件来表示 ARINC653 中的分区概念，swSchedulingResource 和 ProcessingResource 组件共同构建了一个逻辑资源来指明系统运行时资源的分配情况(任务调度，分区资源等)，每个逻辑资源可以用来表明调度信息和内存分配等情况。swSchedulingResource 组件中指明分区内的任务调度信息，swSchedulingResource 中的 schedulers 属性指明分区调度策略相关信息，同时 ProcessingResource 组件指明了分区中的任务集。

(3) 进程转换规则

ARINC653 中的进程(Process)包含了执行代码、执行数据以及堆栈区域等资源，它是系统执行主体。进程被包含在分区中，分区通过指明进程的调度策略、抢占策略、最大响应时间、内存分配情况等信息来控制进程的执行，从而实现相应的应用功能。MARTE 中的 swSchedulingResource 组件是系统最基本的调度执行单元，通过时间周期或外部事件来执行线程，因此可以用 MARTE 中的 swSchedulingResource 组件表示 ARINC653 中的进程概念，相应的任务集所包含的时间约束可通过 swSchedulingResource 组件的相应属性来表示，包括任务的执行周期、执行周期、是否可抢占、截止时间、优先级等。periodElements 属性定义任务的执行周期，timeSliceElements 属性定义任务的执行时间，isPreemptable 属性定义任务是否可抢占，deadlineElements 属性定义任务的截止时间，priorityElements 属性定义任务的固定优先级。

表 1 总结了上述的对应规则，使用这些构件和属性就可以对 ARINC653 系统分区调度信息进行建模，基于建模后包含分区任务集调度信息的 MARTE 模型，就可以采用 MAST 工具进行可调度性分析和判定。

表 1 MARTE 与 ARINC653 分区调度主要概念对应规则

Tab. 1 Semantic mapping from MARTE and ARINC653

MARTE	ARINC653
处理器组件 hwProcessor	模块
mainScheduler	分区间的调度策略
调度资源组件 swSchedulingResource 和处理资源组件 ProcessingResource	分区
schedulers	分区内的调度策略
isPreemptable	分区内的进程任务是否可抢占
调度资源组件 swSchedulingResource	进程
type	进程类型
timeSliceElements	进程执行时间
deadlineElements	进程截止时间
periodElements	进程周期
priorityElements	进程优先级

3　基于 MAST 的分区系统可调度性判定

在完成了第 2 节中给出的从 ARINC653 调度配置信息到 MARTE 调度分析模型的语法转换后,本节中将给出基于模型的系统调度配置信息语义层面的仿真方法验证框架。

3.1　分区系统可调度性验证框架

根据前面介绍的分区调度理论,本文针对含有纯周期任务的分区系统的可调度性判定问题,提出了一种基于第三方工具 MAST 仿真方法实现的可调度性判定方法,方法框架如图 5 所示。MAST 提供自定义的 MAST-1 语言标准和 XML 语言标准[14],MAST 的调度策略具有可扩展性,可以根据需求选用标准自定义调度策略。首先根据 ARINC653 分区调度系统到 MARTE 模型的转换规则对分区调度系统进行建模,同时将自定义调度策略添加到 MARTE 模型中去,然后使用 MAST 工具对其进行分区可调度性分析,得到可调度性判定结果和调度仿真甘特图。

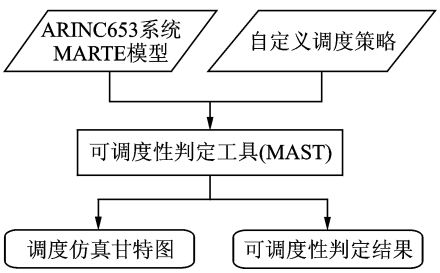

图 5　分区可调度性判定架构

Fig. 5　Framework of partitioned schedulability analysis

3.2　分层调度策略

在使用 MAST 工具所规定的 MAST-1 语言规范结合 IMA 系统分区调度的特征定义一个新的调度策略的过程中,考虑到 MAST-1 仅支持一级调度,而 IMA 系统分区调度分两级调度,本文使用 MAST 自定义调度策略在 MAST-1 上采取预分配时间片的方式来实现两级调度,其中第二级调度——任务调度可以使用 MAST 中的 Scheduling Server 来定义调度策略,第一级调度——分区间调度,需要在第二级调度上采取预先固定分配时间片来实现。

(1) 第一级调度——分区间调度

在文献[15]中提出一个安全关键系统对应的 MAST-1 模型是由一系列 Transactions 组成,每个 Transaction 由一个或多个外部事件(External event)所激活,其内部包含一系列事件处理器(Event handler)和内部事件(Internal event)。所以,可将分区间调度转换成 MAST-1 中的一个特殊 Transaction,这个 Transaction 的入口事件(属于外部事件)的类型为 Periodic,表示主时间片是周期分配的;将分区调度的主时间片轮转周期设置为结束事件(属于内部事件)中的硬全局截止时间(Hard global deadline),各个分区的时间片将设置在入口事件和结束事件之间,并且分别拥有一个硬全局截止时间,表示分区可用的时间片。使用 MAST-1 定义的调度策略的一个实例如图 6 所示。

上述 MAST-1 定义的调度策略程序段描述了一个名为 module 的模块主时间片分配情况,该模

```
Transaction (
    Type=> regular,
    Name=> module,
    External_Events =>
      ( ( Type => Periodic,
          Name => module_input,
          Period => 20.000,
          Max_Jitter => 0.000,
          Phase => 0.000) ),
    Internal_Events =>
      (--分区p1的调度时间属性
      ( Type => Regular,
          Name =>p1_out,
          Timing_Requirements =>
            ( Type => Hard_Global_Deadline,
              Deadline => 5.000,
              Referenced_Event=>module_input)),
       --分区p2的调度时间属性

      ( Type => Regular,
          Name => p2_out,
          Timing_Requirements =>
            ( Type => Hard_Global_Deadline,
              Deadline => 5.000,
              Referenced_Event => p1_out) ) ),

    Event_Handlers =>
      (--从进入模块到分区p1
      (Type => Activity,
          Input_Event => module_input,
          Output_Event => p1_out,
          Activity_Operation => p1,
          Activity_Server => p1_task),
       --从分区p1到分区p2

      (Type => Activity,
          Input_Event => p1_out,
          Output_Event => p2_out,
          Activity_Operation => p2,
          Activity_Server => p2_task) ) );
```

图 6　MAST 调度策略实例

Fig. 6　Example of MAST scheduling strategy

块下有2个分区分别名为p1、p2，入口事件为module_input，周期Period为20，代表主时间片大小为20，模块包含2个内部事件分别名为p1_out，p2_out，p1_out的时间需求Timing_requirements大小为5，表示硬全局截止时间，代表分区p1所分配到的时间片大小为5，同样分区p2所分配到的时间片大小为10。事件处理器Event_Handlers分别指明了分区所对应的任务集，分区p1的任务集为p1_task，分区p2的任务集为p2_task。图7左半部分给出了分区间调度时间片的部分情况，由于ARINC653分区时间片分配的固定性，所以在Transaction中表示出的是流水线式时间片轮转。

(2) 第二级调度——分区内调度

根据MAST-1模型特性，将分区内调度转换成一个Transaction，入口事件(Internal_events)为Unbounded类型，由第一级调度来触发，入口事件直接指向Multicast类型，Multicast类型代表执行多个任务，将Multicast指向的一个或多个活动(Activity)，代表该分区执行一个或多个任务，同时每个任务都有入口事件和结束事件，其中结束事件包含了硬全局截止时间，表示对应任务的最晚截止时间。

图7右半部分给出了上述转换方法的一个简单实例，分区名为p1，包含两个任务t1_task、t2_task，t1_out指明了任务t1_task的硬全局截止时间为3，t2_out指明了任务t2_task的硬全局截止时间为2，具体的MAST-1调度策略程序段与分区间调度类似，这里不再重复介绍。

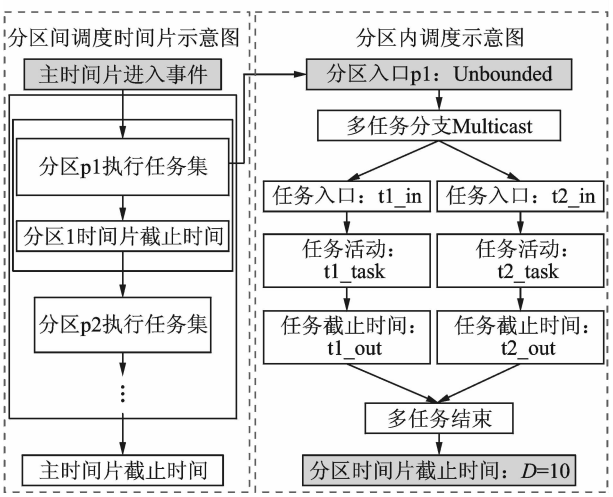

图7 MAST-1分区调度时间片示意图

Fig. 7 MAST-1 partion scheduling slot

3.3 分区可调度性的判定

在调度系统中，如果一个任务在每次到达之后都能在其截止时间限内执行完成，则判定该任务可调度，否则该任务不可调度。从而对任务可调度性的判定即对任务在截止时间期限内是否执行完成的判定。在ARINC653分区系统中，如果所有分区包含的任务集合都能在各自的截止时间限内执行完成，则判定分区系统可调度，否则分区系统不可调度。

对于普通的调度模型，任务集中的所有的任务都是周期任务，并且每个任务的执行仅依赖于其相应的周期。因此，除第一次任务到达的时间点之外的所有任务到达的时间点都是确定的，可通过设定系统时钟变量的方法来模拟调度过程。假设任务集s有n个周期性任务，设定n个任务都在0时刻到达，这n个任务下一次同时到达的时间为此n个任务周期的最小公倍数的时间。n个周期任务的完成在整体上呈现出周期性，并且周期的值为所有任务周期值的最小公倍数，文献[16]给出了相关证明。

针对分区系统以及上述设定的调度模型，系统为分区的周期分配相应的时间片，每个分区下的任务集中的任务都是周期性的任务，所有任务的触发到达均呈现出周期性特点。在分区调度层面，所有任务的到达执行与所属分区的时间片分配情况在整体上是周期性的，值为任务周期值与分区分到的时间片的最小公倍数。

依据上述最小公倍数的原理可知，任务的执行过程以一个值为周期往复执行，因此在判断任务集的可调度性采用仿真方法模拟任务调度的实现过程中，将系统时钟设置成一个有周期值的时钟区域，在此周期内分析任务集的调度状况，从而等价地描述在系统时钟中的剩余时间区域的可调度情况。因为在任务执行过程中有确定性和周期性的存在，所以选取系统时钟的时钟区域是确定的有穷的，由此使用仿真方法来判断分区调度模型的可调度性也是可行的。

4 实例分析

本节将给出一个MARTE对分区系统调度信息建模以及系统可调度性判定的例子，介绍如何借助于MAST自定义的调度策略，使用MARTE对ARINC653系统的分区调度模型进行建模，再利用MAST进行调度仿真，根据仿真返回的结果可知ARINC653分区系统是否可调度。

表 2 描述了一个 ARINC653 系统分区间和分区内的调度信息,并描述了一个总时间框架(10 ms)下分区和分区内任务集的调度情况。系统含有 Pr1 和 Pr2,两个分区,调度策略分别是 DMS 和 RMS,分配的时间片大小分别为 6 和 4,每个分区内都包含有任务(T),周期(Tc),执行时间(Ti)和截止时间(Td)等任务集参数。

表 2　ARINC653 系统调度信息

Tab. 2　Example of ARINC653 schedule congfiguration

分区	时间片	时钟偏移	分配周期	调度策略	分区下任务集参数			
					T	Tc	Ti	Td
Pr1	6	0	10	DMS	T1	10	3	10
					T2	5	1	5
					T	T	Ti	Td
Pr2	4	0	10	RMS	T3	20	2	20
					T4	10	2	10

系统包含 4 个周期任务且周期等于截止时间,调度信息满足上述介绍的分区调度模型。这 4 个周期任务的周期分别是 10,5,20 和 10 ms,最小公倍数是 20 ms,分区获得时间片的周期是 10 ms,所以所有任务周期与分区获得时间片的周期的最小公倍数是 20 ms,即仿真时长是 20 ms。

根据前面介绍的内容,包含表 2 分区调度信息所对应的 MARTE 模型如图 8 所示,将该模型作为 MAST 工具的输入,同时,编辑好自定义的调度策略(如:类似如 3.2 节中的实例形式),调度结果如图 9,10 所示。

可调度性判定结果分为两部分,第一部分是任务在仿真时长内的调度甘特图如图 9 所示,第二部分是判定结果如图 10 所示。调度甘特图中的一小格代表一个时间片,在本例中时间片大小是1 ms,每个分区获得时间的周期是 10 ms。

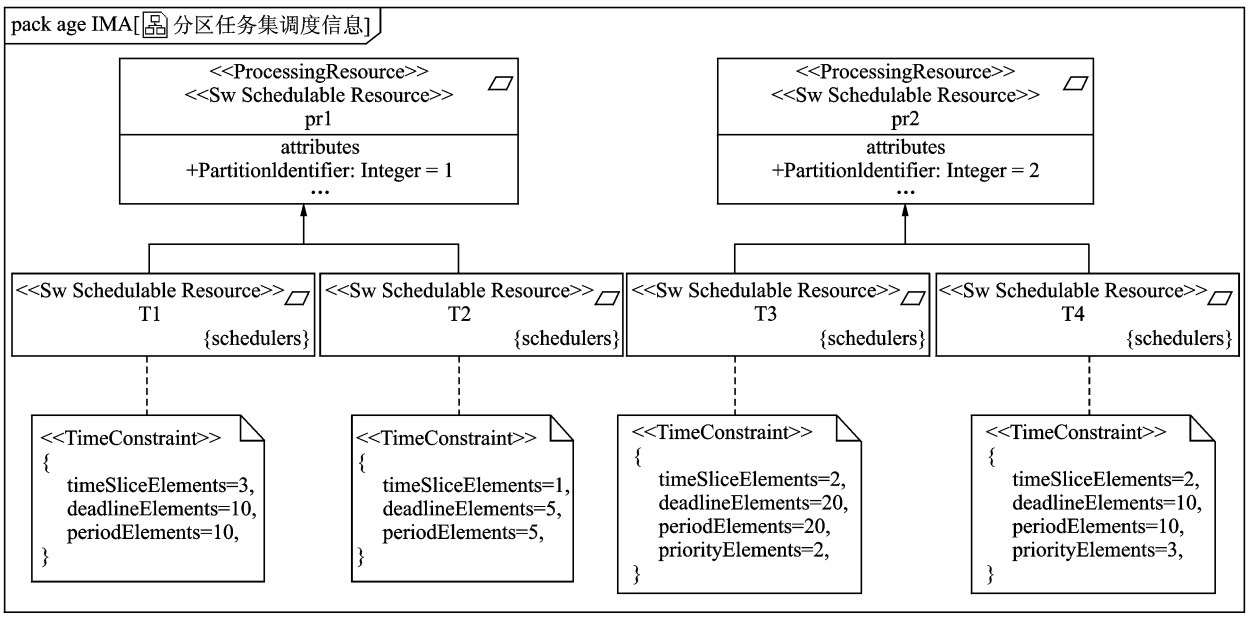

图 8　分区任务调度的 MARTE 模型

Fig. 8　Partion process scheduling of MARTE

分区 partition1 包含任务 T1 和 T2,它被首先分配到6 ms的系统时钟,此时分区 partition2 中任务是空闲不运行的。由于分区 partition1 的调度策略是 DMS,任务的截止时间与优先级成反比,而任务 T1 的截止时间大于 T2,所以 T2 的优先级大于 T1。T2 优先获得系统时钟,当 T2 完成它的任务后(执行时间 1 ms),T1 再获得系统时钟。当 T1 完成它的任务(执行时间 3 ms),分区 partiton1 共执行了 4 ms 的时间,在之后的一个时间片上,由于 partiton1 里没有任务可运行,partiton2 还没有被分配到系统时钟,系统处于空闲状态。在第 6 ms时刻,周期为 5 ms 的任务 T1 再次被分配到 1 ms的时间片。当 T1 执行完这个时间片后,分区

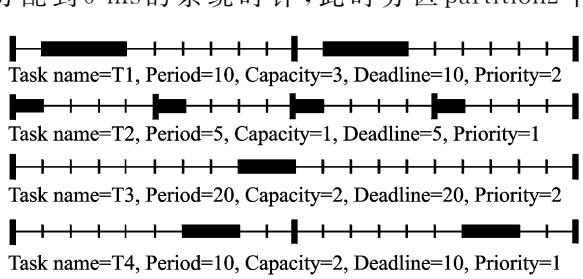

图 9　分区任务调度仿真甘特图

Fig. 9　Gantt chart of partion process scheduling

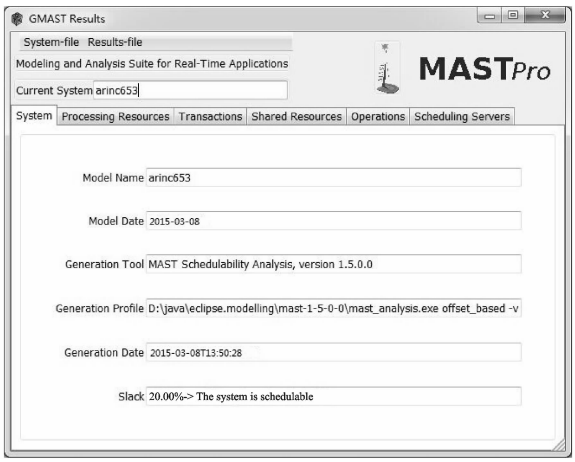

图10 分区任务可调度性分析结果

Fig. 10 Result of partion process schedulability

partiton1 的执行时间达到了 6 ms,系统将把下一个时间片分配给分区 partiton2。

分区 partition2 包含任务 T3 和 T4,由于分区 partition2 的调度策略是 RMS,任务的优先级跟任务的周期成反比。T3 的周期大于 T4 的周期,所以 T4 的优先级大于 T3 的优先级。T4 将会先被分配到系统时钟,执行完任务后(执行时间 2 ms),T3 再获得系统时钟,也执行完任务(执行时间 2 ms)。分区 partition2 共执行了 4 ms 时间,此时系统执行时间达到 10 ms,下个时间片,系统再次将系统时钟分配给分区 partiton1。不过在这个周期内,由于 T3 的周期是 20 ms,它将不会被激活和分配系统时间片。综上所述,系统将以 20 ms 为周期,周期性的进行时间片的分配。MAST 运行的可调度性判定结果表示系统是可调度的,同时给出了系统调度的空闲时间在总的调度周期所占比为 20%,即在整个调度过程中,有 4 ms 的时间系统处于空闲状态。

5 结束语

目前,安全关键系统已经广泛应用于许多安全领域,如航空航天、汽车、医药等领域,同时,这也导致系统越来越复杂;本文研究的以 ARINC653 规范为代表的综合航电架构是一个典型安全关键系统。此类系统中的安全性分析与验证已经成为重要的研究领域;与本文研究内容相关的文献包括以下几类,如:文献[17,18]介绍了几种方法用来检测安全关键系统的潜质错误。不过这些方法依赖于不同的符号和语言,在整个开发阶段无法统一,因此才导致基于模型驱动的安全系统的分析方法的提出[19]。使用模型驱动的方法分析安全系统已经被应用在多个项目中[20,21]。关于安全关键系统的 MARTE 模型的可调度性分析方面,本文所使用的 MAST 就是一款实现调度分析工具,它可以根据 EDF 或 EMS 等经典调度策略公式,基于处理器利用率进行可调性判定。MAST 可以提供自定义的 MAST-1 语言标准和 XML 语言标准[21],根据需求选用标准自定义调度策略,结合模型转换规则对安全关键系统进行建模并利用 MAST 工具对模型进行系统可调度性验证分析。针对分区系统的可调度判定的研究,文献[22]给出了如何在任务时间需求函数的基础上去计算系统所消耗的时间,并得出了系统可调度性的判定定理。文献[23]分析了分区参数对任务调度实时性的影响,通过计算任务的响应时间上界,推导出分区调度下的任务可调度条件。

与以上相关工作相比,本文提出了一种采用模型转换的方法对 ARINC653 系统调度配置信息进行验证分析。首先,将 ARINC653 系统的调度配置信息转换成 MARTE 模型;其次,制定相应的自定义调度策略;最后基于前面的 MARTE 模型和自定义调度策略,调用 MAST 工具,进行可调度性判定。通过可调度性判定,检查 ARINC653 系统调度配置信息是否正确,找出错误进行重新配置再判定,提高系统的安全性和可靠性。

未来的进一步工作主要包括以下两个方面:(1)目前可调度性判定采用 MAST 工具中的 MAST-1 建模规范对转换后的 MARTE 模型进行描述,而该工具的发起者正在制定 MAST-2 建模规范,新规范能更好地支持分区系统的调度性分析,提高易读性、可用性,本文研究下一步工作将会研究采用 MAST-2 来自定义调度策略;(2)将本文提出的方法应用在某型号航电系统的研发过程中,以获取对该研究方法的反馈和改进,并进一步设计直接针对 MARTE 模型语义信息的形式化验证方法及辅助工具,同时对目前中间转换进行精简,提高整个系统的在实际应用中的可用性和工作效率。

参考文献:

[1] Parr G R, Edwards R. Integrated modular avionics [J]. Air & Space Europe, 1999, 1(2): 72-75.

[2] Watkins C B, Walter R. Transitioning from federated avionics architectures to integrated modular avionics [C] // Digital Avionics Systems Conference, 2007. Corinth: IEEE, 2007: 1-2.

[3] Committee A E E. Avionics application software

standard interface[M]. Aunapolis: Aeronautical Radio, 1997.

[4] Airrines Electronic Engineering Committee. Arionics application software standard interface part 1—required services[J]. ARINC Document ARINC Specification, 2006(653):1-2.

[5] Pasaje J L M, Harbour M G, Drake J M. Mast real-time view: A graphic UML tool for modeling object-oriented real-time systems[C]// Real-Time Systems Symposium. [S. l.]: IEEE, 2001: 245-256.

[6] Völter M, Stahl T, Bettin J, et al. Model-driven software development: Technology, engineering, management[M]. NewYork: John Wiley & Sons, 2013.

[7] Hutchinson J, Rouncefield M, Whittle J. Model-driven engineering practices in industry[C]// Software Engineering (ICSE). [S. l.]: IEEE, 2011: 633-642.

[8] Rutle A, Rossini A, Lamo Y, et al. A formal approach to the specification and transformation of constraints in MDE[J]. The Journal of Logic and Algebraic Programming, 2012, 81(4): 422-457.

[9] Moy Y, Ledinot E, Delseny H, et al. Testing or formal verification: DO-178C alternatives and industrial Experience[J]. Software, 2013, 30(3): 50-57.

[10] Rierson L. Developing safety-critical software: A practical guide for aviation software and DO-178C compliance[M]. Boca Raton: CRC Press, 2013.

[11] Rushby J. New challenges in certification for aircraft software[C]// Proceedings of the ninth ACM international conference on Embedded software. NewYork: ACM, 2011: 211-218.

[12] Graf S, Gérard S, Haugen O, et al. Modeling and analysis of real-time and embedded systems[C]// Satellite Events at the MoDELS 2005 Conference. Berlin: Springer, 2006: 58-66.

[13] Omg M. Modeling and analysis of real-time and embedded systems specification version 1.1 formal-11-06-02[Z]. Needham: Object Management Group, 2011.

[14] Vrba P. Mast: manufacturing agent simulation tool [C]// Emerging Technologies and Factory Automation. [S. l.]: IEEE, 2003: 282-287.

[15] González Harbour M, Gutiérrez García J J, Palencia Gutiérrez J C, et al. Mast: Modeling and analysis suite for real time applications[C]// Real-Time Systems, 13th Euromicro Conference on. [S. l.]: IEEE, 2001: 125-134.

[16] Leung J, Merrill M L. A note on preemptive scheduling of periodic real-time tasks[J]. Information Processing Letters, 1980, 11(3): 115-118.

[17] Atchison B, Lindsay P. Safety validation of embedded control software using Z animation [C]// Proc of the 5th IEEE International Symposium on High Assurance Systems Engineering. Albuquerque, USA: IEEE Press, 2000: 228-237.

[18] Conmy P, Nicholson M, McDermid J. Safety assurance contracts for integrated modular avionics[C]// Proc of the 8th Australian Workshop on Safety Critical Systems and Software. Canberra, Australian: [s. n.], 2003: 6978.

[19] Kashi R N, Amarnathan M. Perspectives on the use of model based development approach for safety critical avionics software development [C]// Proc of International Conference on Aerospace Science and Technology. Bangaloire, India: [s. n.], 2008.

[20] Chilenski J. Aerospace vehicle systems institute systems and software integration verification overview [C]// Proc of AADL Safety and Security Modeling Meeting. [S. l.]: IEEE Press, 2007.

[21] Delange J, Hugues J, Pautet L, et al. Code generation strategies from AADL architectural descriptions targeting the high integrity domain [C]// Proc of the 4th European Congress ERTS Embedded Real-time Software. Toulouse, France: [s. n.], 2008.

[22] 何 峰, 宋丽茹, 熊华钢. 航空电子双层任务分区调度设计[J]. 北京航空航天大学学报, 2008, 34(11): 1364-1368.
He Feng, Song Liru, Xiong Huagang. Two level task partition scheduling design in integrated modular avionics[J]. Journal of Beijing University of Aeronautics and Astronautics, 2008, 34(11): 1364-1368.

[23] 周天然, 熊华钢. 航空电子系统混合实时任务的双层调度[J]. 航空学报, 2011, 32(6): 1067-1074.
Zhou Tianran, Xiong Huagang. Two-level hierarchical scheduling for hybrid real-time tasks in avionic systems[J]. Acta Aeronautica et Astronautica Sinica, 2011, 32(6): 1067-1074.