

doi: 10.12052/gdutxb.190037

基于软件定义的移动信息物理融合系统

刘珍娟, 张立臣

(广东工业大学 计算机学院, 广东 广州 510006)

摘要: 在汽车和飞机等交通工具上安装传感器等设备, 导致了新一代移动信息物理融合系统MCPS(Mobile Cyber-Physical System)的出现. 针对MCPS主要以硬件资源为核心, 自动化水平不高的问题, 本文提出了一种基于软件定义的MCPS建模方法. 通过结构分析与设计语言AADL(Architectural Analysis and Design Language)对硬件资源虚拟化, 然后使用软件定义的方法对硬件资源进行分析和处理, 可以有效提高MCPS系统的自动化水平. 以移动车辆自主处理分析传感器数据并控制车辆的驾驶为案例, 详细介绍系统的组成和建模过程. 并通过添加安全附件对该模型的安全性进行分析验证, 证实了基于软件定义建模的可行性与有效性.

关键词: 移动信息物理融合系统; 软件定义; 结构分析与设计语言

中图分类号: TP302 文献标志码: A 文章编号: 1007-7162(2019)06-0059-07

A Mobile Cyber-Physical System Based on Software Definition

Liu Zhen-juan, Zhang Li-chen

(School of Computers, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Installing sensors and other devices in vehicles such as cars and aircraft have led to the emergence of a new generation of mobile cyber-physical system (MCPS). However, MCPS is mainly based on hardware resources and has low automation level. A MCPS modeling method is proposed based on software defined approach. Virtualizing the hardware resources by AADL (Architectural Analysis and Design Language), and then analyzing and processing the hardware resources through the software-defined method, the automation level of the MCPS system can be effectively improved. The mobile vehicle which autonomously processes and analyzes sensor data for controlling vehicle driving is taken as an instance and the system composition and model is described. Finally, the security of the system is analyzed by adding a secure annex. This instance demonstrates the feasibility and effectiveness of software-defined modeling method.

Key words: mobile cyber physical systems(MCPS); software-definition; architectural analysis and design language(AADL)

信息物理融合系统(Cyber Physical Systems, CPS)是结合了实时计算系统和物理系统3C(Computation, Communication, Control)系统^[1], 近年来引起了国内外广泛的关注^[2-3]. CPS主要是利用大量的传感器和控制器将网络部分和物理部分整合起来, 在提高物理系统的可控性、高效性和可靠性方面取得了很好的效果. CPS采用了不同类型的传感器, 因此会产生大量的异源异构数据, 这些传感器不仅包括固定在某处收集数据的静态传感器, 也包括随着火车、飞机和

机器人等载体移动的动态传感器^[4], 这也因此引出了移动信息物理融合系统(Mobile Cyber Physical Systems, MCPS), 并且在人们日常生活中随处可见.

MCPS研究的核心问题在于如何有效管理计算系统与物理系统之间复杂的动态交互问题, 旨在提供广泛的应用和服务, 随着MCPS更加智能、节能和舒适, 其复杂性也不断增加^[5]. 然而, 目前大部分MCPS主要是以硬件资源为核心^[6], 这也导致了自动化水平不高. 然后随着工业技术的不断发展, 对自动

收稿日期: 2019-03-15
基金项目: 国家自然科学基金资助项目(61873068)
作者简介: 刘珍娟(1995-), 女, 硕士研究生. 主要研究方向为信息物理融合系统.
通信作者: 张立臣(1962-), 男, 教授. 主要研究方向为信息物理融合系统. E-mail: zhanglichen1962@163.com

化的要求水平越来越高,传统的控制理论不能满足复杂MCPS的控制需求^[7].

本文提出了一种基于软件定义的高自动化水平的MCPS建模方法,是指将整个系统的硬件资源虚拟化之后再由软件来控制硬件系统. 由软件来控制 and 调度硬件资源的方法在很大程度上提高了MCPS的自动化水平,通过在AADL中增添安全附件的方式,分析整个系统各个部分的安全性. 这可以在开发前期有效分析系统出现错误的概率,节约移动车辆系统的开发成本^[8-9].

MCPS是典型的大数据驱动系统^[10-11],因此大数据技术对于MCPS是非常基础但又是非常关键的技术,数据处理平台对处理后的数据分析出有用的结果数据供MCPS中其他平台使用. MCPS的安全性首先体现在必须及时处理大量捕获的数据集,对数据进行冗余处理和错误数据删选可大大提高计算速度. 最后采取大数据的流处理技术保证处理平台的运行效率.

1 相关工作

1.1 软件定义

软件定义的这一术语被提出的时间不长^[12],它的概念却早已被众多研究领域应用. 软件定义是将软件和硬件分开来处理,它将三大计算资源虚拟化,其中包括服务器、存储和网络资源,虚拟化后的资源可以按需分配. 但软件定义并不意味着等同于虚拟化,因为它实现了彻底的软硬件分离. 由硬件来负责存储和计算,由软件来负责管理和控制. 概括起来就是:“硬件资源虚拟化,管理功能可编程”. 目前软件定义的研究主要集中在软件定义网络、软件定义存储和软件定义数据中心等领域^[13]. 实现软件定义的优点包括提升资源利用效率、降低运行成本、拥有更好的可扩展性、支撑智能设备转型.

本文的软件定义是指通过制定策略集,由软件应用程序来管理MCPS,使车辆驾驶高度自动化,在驾驶车辆的过程中控制硬件的行为就可以根据策略集中制定的对应策略来实施操作,也就是由软件来控制硬件系统. 而MCPS是软件与硬件结合的系统,为了在MCPS中实现以编程软件为核心的高自动化水平,将硬件资源看作是可以根据需要进行使用的虚拟化资源. 在车辆硬件资源虚拟化的基础上,编写应用程序,由软件来对MCPS的硬件资源进行管理和控制.

1.2 体系结构分析与设计语言

本文对MCPS进行建模使用了体系结构分析与

设计语言AADL(Architecture Analysis & Design Language),AADL是一种文本和图形语言,它结合了UML、MetaH及多种ADLs的优点,可提供标准的方式设计与分析嵌入式实时系统的软硬件体系结构^[14-15]. AADL可应用于高可靠性、高实时性的系统^[16],对本文的MCPS同样适用.

AADL具有语法简单、功能性强和可扩展的优点. 它提出了两个扩展机制:属性集和扩展. 属性集允许定义自定义属性来扩展标准属性,还可对数据类型进行精确的建模,以便对其进行操作^[17]. 为了加强硬件资源虚拟化和MCPS的连续性和动态性的AADL表示,采用了annex对系统进行了扩展,同时也在建模文件中添加了错误附件,以此来增强系统的安全性.

AADL组件类别大致包括以下4种:

- (1) 应用软件组件:数据、线程、线程组、进程、子程序和子程序组.
- (2) 执行平台组件:处理器、虚拟处理器、存储器、总线、虚拟总线和设备.
- (3) 复合组件:系统,允许软件和执行平台组件被组织成具有良好接口的层次结构.
- (4) 一般组件:抽象.

AADL建模元素及元素间的联系如图1所示.

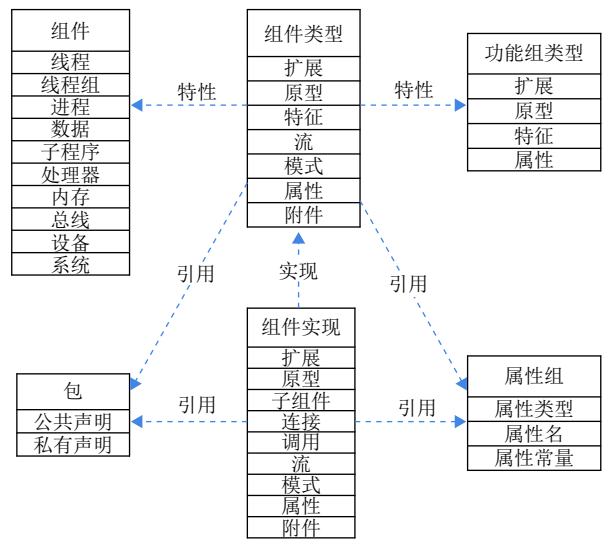


图1 AADL建模元素及元素间的联系图

Fig.1 AADL modeling elements and the connection diagram between elements

2 系统结构分析

本文提出的基于软件定义的MCPS由两部分组成:第1部分是基于软件定义的控制中心,软件定义

是由事先编程好的软件来实现,定义了一系列策略集,这些策略集规定了不同的数据结果对应于不同的决策方案.第2部分是MCPS系统,其中又包括几个小部分,内容如下:

(1) 数据传感器.由几种不同类型的传感器组成的数据传感器,包括速度传感器、位置传感器、距离传感器和加速度传感器等.

(2) 数据收集平台.采集的传感器数据在收集平台进行简单标准化预处理之后传给下一部分.

(3) 数据处理平台.首先要消除冗余数据,降低数据复杂度,有效地提取有用信息.为了保证整个系统的安全,系统的每个部分都需要在非常短的时间内处理待处理的数据.因此采用现有较成熟的大数据流处理技术实时处理采集到的多源异构数据,处理之后的结果数据传递给下一部分.

(4) 数据决策平台,根据上一部分提供的数据进行决策.因为这些数据提供的可供选择的决策很多时候不是唯一的,那就意味着可能会有不同的可供选择的行动方案,此时由决策单元来挑选一种最合适的行动方案或者是对几种方案进行协调.

(5) 控制传感器,通过接收到决策平台发出的控制信号,采取相应的动作,对应的数据传感器数据将会发生相应的变化.比如,当决策平台发现汽车超速时,会向刹车发送控制信号,当汽车采取刹车动作后,速度传感器的数据会降低.

(6) 数据监测中心,是对收集、处理和决策平台的传输数据进行安全认证,当任意一个部分发生数据异常时,数据监测中心发出报警,并向控制传感器发送控制信号,保证整个系统数据传输的安全性.

系统整体设计框架如图2所示,整个系统可以有效实现一个闭环控制,数据传感器通过将数据依次传输到收集、处理平台,然后决策平台向控制传感器发生控制信号,最后控制传感器调节相应的数据传感器数据.

MCPS是基于软件定义实现的,而软件定义的概念应用在MCPS中就是为了实现汽车的高度自动化,甚至是完全自动化.意味着大部分的汽车驾驶行为都可以根据软件应用程序中事先定义好的策略集里面的方案完成.在汽车的行驶过程中,传感器负责收集好汽车安全驾驶所需要的所有相关数据,在收集平台中完成数据预处理步骤.预处理是因为采集的数据的类型不一,为了提升传输速度需要进行标准化预处理.

本文的MCPS可以实现的功能相当于一个驾驶

员所具备的能力.具体说来就是,速度传感器、位置传感器、距离传感器和加速度传感器等传感器采集到的所有数据都是由各种传感器采集来的,可是一个驾驶员需要靠眼睛的观察才能采集到这些数据,眼睛的作用就相当于传感器的功能,比如说根据反光镜来目测后面车辆的靠近速度,根据左右观察来目测附近车辆的相对位置.总的来说,收集平台就相当于人类大脑的知觉中心.而MCPS中的数据处理平台,就相当于驾驶员的大脑的反应中心,对采集到的车辆行驶的相关数据在平台进行处理.对处理后的结果进行判断后再选择一个最合适的处理方案,这个步骤是由驾驶员的大脑的决策中心完成,在MCPS中也设计了一个具有同等功能的决策平台.最后驾驶员会根据处理的结果采取对应的行动,具体就是手控制方向盘的移动和脚控制刹车或者油门,这些由硬件设备完成的功能.

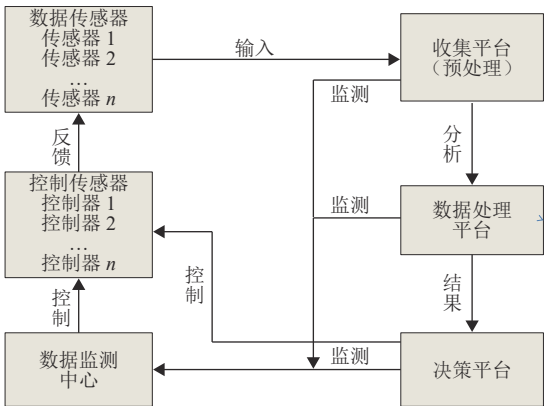


图2 基于软件定义的MCPS框架图
Fig.2 Software-defined MCPS framework diagram

实时性、准确性和安全性对于软件定义的MCPS是非常重要的.通过MCPS系统中的设备分布式部署,保证了采集数据的实时性和准确性,同时实时性也可通过高效的传感器和实时的大数据流处理等技术来帮助完成.而MCPS为了达到安全可靠都是一开始就从各种细节实现上来考虑安全性.在软件定义这一模块中,必须将安全编码需求抽象为设计特征和约束,在正式建模中也该考虑这些约束以防止出现设备故障或者是外部攻击等危险发生.

3 AADL建模实现

AADL可应用于高可靠性.高实时性的系统,它相比Modelica更适合对软件定义的系统进行建模,因此本文采用AADL对MCPS进行建模.本节是通过AADL在OSATE2(支持AADL的开源工具平

台)来实现MCPS的建模,OSATE是作为Eclipse平台上的一套插件,用于AADL建模、编译和分析,支持可调度性、端对端的流延迟、安全性、系统资源与能耗等分析.

本节分别对上节提到的6个部分分别用AADL建模,如图3所示,并添加相应的安全附件,确保整个

系统安全运行. AADL可以对硬件进行虚拟化,通过对硬件添加附件,可以分析整个系统在运行过程中可能出错的概率,及时发现潜在错误,有效节约硬件开发的成本. 本文提出的基于软件定义的AADL建模通过用程序来控制硬件,可以有效提高AADL建模的自动化水平.

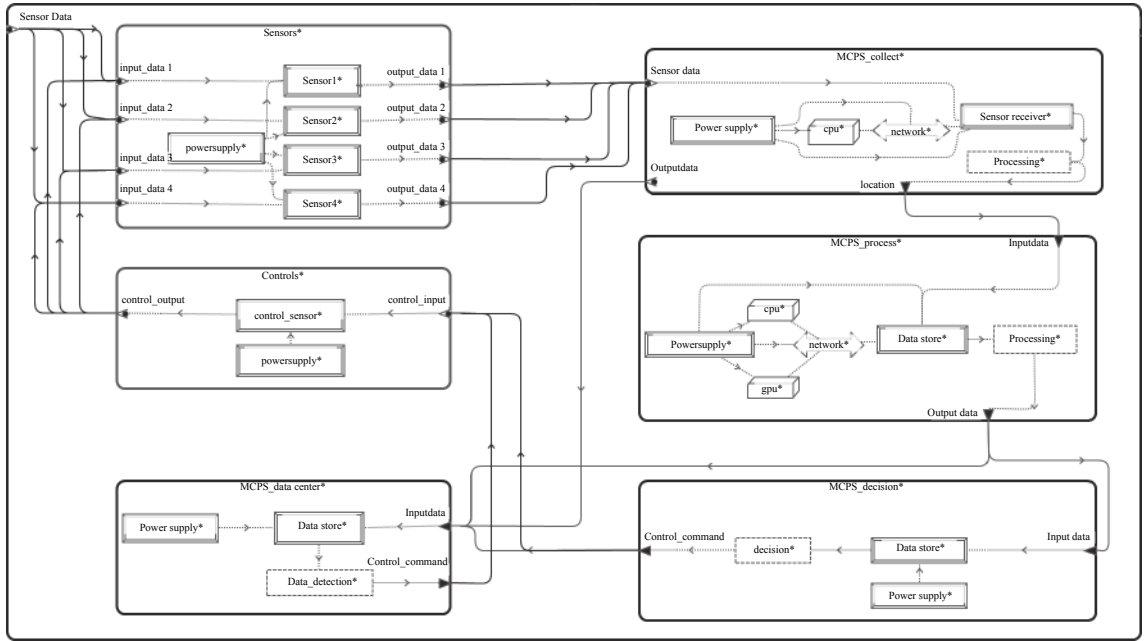


图 3 MCPS的AADL实现
Fig.3 Implementation of MCPS by AADL

3.1 数据传感器

数据传感器主要收集系统驾驶过程中的各项指标,主要包括:数据接收端口、数据发送端口、电源,有些传感器可能还需要网络连接. 这些部件都有可能发生错误,比如传感器可能不能接收数据,不能发送数据以及电源电压过高、过低或者断电. 本文通过添加附件的方式来模拟数据传感器可能出现的错误,附件AADL实现如下所示.

```
annex EMV2 {**
use types ErrorLibrary, CPSErrorLibrary;
use behavior CPSErrorLibrary::CollectingFailed;
error propagations
    data_input: in propagation {Nodata,
IncorrectData};
    data_output: out propagation {Nodata, IncorrectData};
    power_input: in propagation {NoPower}
flows
    Error: error source data_output {Nodata, IncorrectData}
when {SystemFailed};
    nodataflow: error path data_input{Nodata} ->
```

```
sensordata_output{Nodata};
    incorrectdataflow: error path data_input{IncorrectData} -
> data_output{IncorrectData};
end propagations; **}
```

3.2 数据收集平台

数据收集平台主要存储由数据传感器传输过来的数据,主要有数据接收器,在数据收集阶段需要对数据进行一些简单的预处理,因此还需要CPU处理器,而CPU又通过网络数据接收器进行读取操作. 对该部分的安全附件的AADL实现如下所示.

```
annex EMV2 {**
use types ErrorLibrary,CPSErrorLibrary;
use behavior CPSErrorLibrary::CollectingFailed;
error propagations
    bindings: out propagation {ServiceOmission,
ValueError};
    powersource: in propagation {NoPower};
    networkaccess: in propagation {ServiceOmission};
    networkaccess: out propagation {ServiceOmission};
flows
```

```

    cpu_error: error source bindings {ServiceOmission,
ValueError} when {CPUFailure};
    end propagations;
    component error behavior
    propagations
        FailStop-[]-> bindings {ServiceOmission};
    end component; **}

```

3.3 数据处理平台

数据处理平台主要处理由数据收集平台存储的数据,主要由CPU、GPU、数据存储器、数据处理器组成. 该部分可以使用大数据和机器学习等方法,提高数据处理的可靠性和稳定性,为整个系统的稳定运行提供坚实的基础. 对这一部分的安全附件的AADL实现如下所示.

```

annex EMV2 {**
    use types ErrorLibrary, CPSErrorLibrary;
    use behavior CPSErrorLibrary::CollectingFailed;
    error propagations
        Inputdata: in propagation {Nodata, IncorrectData};
        OutputData: out propagation {Nodata, IncorrectData};
    flows
        error: error source OutputData {Nodata,
IncorrectData} when {SystemFailed};
        nodataerror: error path Inputdata {Nodata} ->
OutputData {Nodata};
        incorrectdataerror2: error path Inputdata
{IncorrectData} -> OutputData {IncorrectData};
    end propagations; **}

```

3.4 数据决策平台

数据决策平台主要对数据处理平台的处理结果进行分析和决策,主要由数据存储器 and 数据决策器组成. 数据存储器主要用来存储汽车各项数据指标的安全值范围,这样方便决策器做出正确的判断,其AADL安全附件代码如下所示.

```

annex EMV2 {**
    use types ErrorLibrary, CPSErrorLibrary;
    use behavior CPSErrorLibrary::CollectingFailed;
    error propagations
        Inputdata: in propagation {Nodata, IncorrectData};
        Control_Command: out propagation {Nodata,
IncorrectData};
    flows
        decisionerror: error source Control_Command
{Nodata, IncorrectData} when {SystemFailed};
        nodataerror: error path Inputdata {Nodata} ->
Control_Command {Nodata};

```

```

        incorrecterror: error path Inputdata {IncorrectData}-
> Control_Command {IncorrectData};
    end propagations; **}

```

3.5 控制传感器

控制传感器主要接收由数据决策平台发送过来的控制信号,反过来调节数据传感器的数据,保障整个系统数据的稳定性. 主要包括汽车的各个部分的控制传感器,其附件的AADL实现如下所示.

```

annex EMV2 {**
    use types ErrorLibrary, CPSErrorLibrary;
    use behavior CPSErrorLibrary::CollectingFailed;
    error propagations
        control_in: in propagation {Nodata, IncorrectData};
        control_out: out propagation {Nodata,
IncorrectData};
    flows
        control_error: error source control_output {Nodata,
IncorrectData} when {DatasensorFailed};
        nocontrol_error: error path control_in {Nodata} ->
control_out {Nodata};
        incorrectcontrol_error: error path control_in
{IncorrectData} -> control_out {IncorrectData};
    end propagations;
    properties
        emv2::OccurrenceDistribution=>
[ProbabilityValue=> 8.0e-7; Distribution=> Poisson;] applies
to control_in.Nodata, control_out.Nodata, FailStop;
        emv2::OccurrenceDistribution=>
[ProbabilityValue=> 3.0e-7; Distribution=> Poisson;] applies
to control_in.IncorrectData, control_out.IncorrectData; **}

```

3.6 数据检测中心

数据监测中心主要监测整个系统中各个模块的数据异常情况,并可以直接控制控制传感器及时处理异常数据,是整个系统中非常关键的一步,其AADL实现的安全附件如下所示.

```

annex EMV2 {**
    use types ErrorLibrary, CPSErrorLibrary;
    use behavior CPSErrorLibrary::CollectingFailed;
    error propagations
        Inputdata: in propagation {Nodata, IncorrectData};
        Control_cmd: out propagation {Nodata,
IncorrectData};
    flows
        cmd_error: error source
Control_cmd {Nodata, IncorrectData};
        nodataerror: error path Inputdata {Nodata} ->
Control_Command {Nodata};

```

```
incorrectdataerror: error path Inputdata
{IncorrectData}-> Control_Command{IncorrectData};
end propagations; **};
```

4 数据安全分析

本文以移动车辆自主处理分析传感器数据并控制车辆的驾驶为案例,通过OSATE2的故障分析树对该系统的安全性进行分析验证.

利用AADL对汽车硬件资源进行虚拟化,并初始化各硬件的故障概率,如表1所示. 整个汽车系统包括6个模块:数据传感器模块、数据收集模块、数据处理模块、数据决策模块、控制器模块和数据监测中心模块. 每个模块都有各自的故障概率,如表2所示,其中允许故障概率表示该模块允许发生故障的最大概率,如果超过该值,系统将处于危险状态. 计算故障概率等于该模块下所有的硬件发生故障概率之和. 例如,如图4所示,MCPS_Collect为数据收集模块,系统计算概率等于数据接收器、电源、CPU、数据处理器、网络的故障概率之和,即 $6.9\times10^{-5}=5.0\times10^{-6}+2.0\times10^{-6}+8.0\times10^{-7}+8.0\times10^{-7}+6.0\times10^{-5}$,其他5个模

表 1 汽车硬件故障概率

Table 1 Vehicle hardware failure probability

硬件名称	故障发生概率
数据采集传感器	5.0×10^{-6}
电源	2.0×10^{-6}
CPU	8.0×10^{-7}
GPU	8.0×10^{-7}
网络传输	6.0×10^{-5}
数据接收器	5.0×10^{-6}
数据存储器	5.0×10^{-6}
数据处理器	8.0×10^{-7}

表 2 汽车自动驾驶系统中各模块故障概率

Table 2 Failure probability of each module in automobile automatic driving system

模块类型	允许故障概率	计算故障概率
数据传感器模块	7.0×10^{-5}	2.2×10^{-5}
数据收集模块	7.0×10^{-5}	6.9×10^{-5}
数据处理模块	7.0×10^{-5}	6.9×10^{-5}
数据决策模块	7.0×10^{-5}	7.8×10^{-6}
控制器模块	7.0×10^{-5}	7.0×10^{-6}
数据监测中心模块	7.0×10^{-5}	7.8×10^{-6}

	Error Model Element/Type	Computed Probability	Specified Probability
└─ 'MCPS_Collect'			
└─ 'MCPS_Collect.SensorReceiver'	error state 'CollectFailed'	6.9e-05	7.0e-05
└─ 'MCPS_Collect.PowerSupply'	error state 'HardwareFailed'		5.0e-06
└─ 'MCPS_Collect.cpu'	error state 'HardwareFailed'		2.0e-06
└─ 'MCPS_Collect.processing'	error state 'HardwareFailed'		8.0e-07
└─ 'MCPS_Collect.network'	error state 'HardwareFailed'		8.0e-07
└─ 'MCPS_Process'	error state 'ProcessFailed'	6.9e-05	6.0e-05
└─ 'MCPS_Process.network'	error state 'HardwareFailed'		8.0e-07
└─ 'MCPS_Process.processing'	error state 'HardwareFailed'		8.0e-07
└─ 'MCPS_Process.cpu'	error state 'HardwareFailed'		8.0e-07
└─ 'MCPS_Process.gpu'	error state 'HardwareFailed'		8.0e-07
└─ 'MCPS_Process.PowerSupply'	error state 'HardwareFailed'		2.0e-06
└─ 'MCPS_Process.DataStore'	error state 'HardwareFailed'		5.0e-06
└─ 'MCPS_Decision'	error state 'DecisionFailed'	7.8e-06	7.0e-05
└─ 'MCPS_Decision.decision'	error state 'HardwareFailed'		8.0e-07
└─ 'MCPS_Decision.PowerSupply'	error state 'HardwareFailed'		2.0e-06
└─ 'MCPS_Decision.DataStore'	error state 'HardwareFailed'		5.0e-06
└─ 'MCPS_DataCenter'	error state 'data_auth_failed'	7.8e-06	7.0e-05
└─ 'MCPS_DataCenter.data_detection'	error state 'HardwareFailed'		8.0e-07
└─ 'MCPS_DataCenter.PowerSupply'	error state 'HardwareFailed'		2.0e-06
└─ 'MCPS_DataCenter.DataStore'	error state 'HardwareFailed'		5.0e-06
└─ 'Sensors'	error state 'HardwareFailed'	2.2e-05	7.0e-05
└─ 'Sensors.sensor1'	error state 'HardwareFailed'		5.0e-06
└─ 'Sensors.sensor2'	error state 'HardwareFailed'		5.0e-06
└─ 'Sensors.sensor3'	error state 'HardwareFailed'		5.0e-06
└─ 'Sensors.sensor4'	error state 'HardwareFailed'		5.0e-06
└─ 'Sensors.powersupply'	error state 'HardwareFailed'		2.0e-06
└─ 'Controls'	error state 'HardwareFailed'	7.0e-06	7.0e-05
└─ 'Controls.control_sensor'	error state 'HardwareFailed'		5.0e-06
└─ 'Controls.powersupply'	error state 'HardwareFailed'		2.0e-06

图 4 系统安全分析

Fig.4 System safety analysis

块也以同样方式计算.从表2可以看出,6个模块的计算故障概率都要比允许故障概率小,从而说明本系统设计的移动车辆系统的安全性.假设整个系统只有这6个模块可能出现故障,如表2所示.可以计算出整个系统允许的最大故障概率为 $6 \times 7.0 \times 10^{-5} = 4.2 \times 10^{-4}$,而由OSATE2的故障分析树计算出来的实际故障概率为 $2.2 \times 10^{-5} + 6.9 \times 10^{-5} + 6.9 \times 10^{-5} + 7.8 \times 10^{-6} + 7.0 \times 10^{-6} + 7.8 \times 10^{-6} = 1.8 \times 10^{-4}$.无论是从整体系统还是各个模块的安全系数数据分析,都可以看到本文系统设计的安全性和有效性.

5 总结

本文针对MCPS主要以硬件资源为核心,自动化水平不高这一问题,提出了一种基于软件定义的高自动化水平MCPS建模方法.通过使用AADL建模语言,将硬件资源虚拟化,提高MCPS的自动化水平.以移动车辆自主处理分析传感器数据并控制车辆的驾驶为案例,具体步骤为:首先,传感器接收汽车各项功能的数据,将数据传输给数据收集平台,数据收集平台将数据收集好后传输给数据处理平台,数据处理平台将处理好的数据交给数据决策平台,决策平台控制汽车控制传感器,调节相应的数据传感器数据,整个系统是一个闭环控制,可以实现数据的动态平衡.该系统中还有一个数据监测中心,可以实时监控各个部分的数据异常情况,有效保证了系统的安全有效运行.

通过在AADL中增添安全附件的方式,分析整个系统各个模块故障概率,实时监控系统的安全性,保证系统每个模块都能够正常工作,减少潜在事故发生的概率.

参考文献:

- [1] LIU Y, PENG Y, WANG B, *et al.* Review on cyber-physical systems [J]. *IEEE/CAA Journal of Automatic Sonica*, 2017, 4(1): 27-40.
- [2] XIONG G. Cyber-physical-social system in intelligent transportation [J]. *IEEE/CAA Journal of Automatica Sinica*, 2015, 2(3): 320-333.
- [3] LEE E A. The past, present and future of cyber-physical systems: a focus on models [J]. *Sensors*, 2015, 15(3): 4837-4869.
- [4] ZHANG L. A framework to specify big data driven complex cyber physical control systems[C]// International Conference on Information and Automation (ICIA). [S. l.]: IEEE, 2014: 548-553.
- [5] COLOMBO A W, KARNOUSKOS S, KAYNAK O, *et al.* Industrial cyberphysical systems: a backbone of the fourth industrial revolution [J]. *IEEE Industrial Electronics Magazine*, 2017, 11(1): 6-16.
- [6] HARISON ROBERT, D. VERA, B. AHMAD, *et al.* Engineering methods and tools for cyber-physical automation systems [J]. *Proceedings of the IEEE*, 2016, 104(5): 973-985.
- [7] ZHU Q, BASAR T. A hierarchical security architecture for cyber-physical systems [J]. Office of Scientific & Technical Information Technical Reports, 2011, 1(3): 15-20.
- [8] TANG L A, HAN J, JIANG G. Mining sensor data in cyber-physical systems [J]. *Tsinghua Science and Technology*, 2014, 19(3): 225-234.
- [9] BABICEANU R F, SEKER R. Big data and virtualization for manufacturing cyber-physical systems: a survey of the current status and future outlook [J]. *Computers in Industry*, 2016, 81: 128-137.
- [10] 覃华. 大数据驱动的车联网信息物理融合系统的分析与设计方法[D]. 广州: 广东工业大学, 2016.
- [11] 程奇华, 张立臣. 信息物理融合系统语义模型分析[J]. *广东工业大学学报*, 2016, 33(3): 43-48.
- CHENG Q H, ZHANG L C. A semantic model analysis for cyber-physical systems [J]. *Journal of Guangdong University of Technology*, 2016, 33(3): 43-48.
- [12] 徐慧, 陈宏伟, 宗欣露. 基于可拓论的SDN管理信息语义不匹配问题研究[J]. *广东工业大学学报*, 2017, 34(4): 17-21.
- XU H, CHEN H W, ZONG X L. Semantic-mismatching of management information for software-defined networking based on extension theory [J]. *Journal of Guangdong University of Technology*, 2017, 34(4): 17-21.
- [13] KATHIRAVELU P, VEIGA L. SD-CPS: Taming the challenges of cyber-physical systems with a software-defined approach[C]//International Conference on Software Defined Systems. [S. l.]: IEEE, 2017: 6-13.
- [14] FENG S, ZHANG L. Model transformation for cyber physical systems. advances in computer science and its applications[M]. Berlin, Heidelberg: Springer, 2014. 83-87.
- [15] 李揭阳. 基于AADL的信息物理融合系统的建模与验证[D]. 南京: 南京航空航天大学, 2017.
- [16] 陈海华, 张立臣. 大数据下的CPS自适应路由服务[J]. *计算机工程与应用*, 2018, 54(1): 94-101.
- CHEN H H, ZHANG L C. Cyber-physical system adaptive routing service with large data [J]. *CEA*, 2018, 54(1): 94-101.
- [17] 杨帆, 刘彦, 李仁发, 等. 一种基于数据驱动CPS建模方法研究[J]. *计算机学报*, 2016, 39(5): 961-972.
- YANG F, LIU Y, LI R F, *et al.* A modeling method research based on data in cyber-physical system [J]. *Chinese Journal of Computers*, 2016, 39(5): 961-972.