软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(6):1123−1139 [doi: 10.3724/SP.J.1001.2011.04026]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

策略驱动的可靠嵌入式系统建模及分析方法*

范贵生[1,2], 虞慧群[1+], 陈丽琼[3], 刘冬梅[1]

[1](华东理工大学 计算机科学与工程系,上海 200237)

[2](武汉大学 软件工程国家重点实验室,湖北 武汉 430072)

[3](上海应用技术学院 计算机科学与信息工程系,上海 200235)

Strategy Driven Modeling and Analysis of Reliable Embedded Systems

FAN Gui-Sheng[1,2], YU Hui-Qun[1+], CHEN Li-Qiong[3], LIU Dong-Mei[1]

[1](Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

[2](State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)

[3](Department of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai 200235, China)

+ Corresponding author: E-mail: yhq@ecust.edu.cn, http://cs.ecust.edu.cn/~yhq

Fan GS, Yu HQ, Chen LQ, Liu DM. Strategy driven modeling and analysis of reliable embedded systems. *Journal of Software*, 2011,22(6):1123−1139. http://www.jos.org.cn/1000-9825/4026.htm

Abstract: This paper proposes a strategy driven approach to modeling and analyzing reliable embedded systems according to their characteristics. Petri nets are used as the formal description language for embedded systems, which formally specify embedded system's elements such as equipment, computing, physical interaction, components, and communication processes. This research studies reliability assurance strategies for embedded systems by analyzing various fault types and their characteristics. An aspect-oriented method is used to extract reliability related concerns. A complete embedded system model is obtained by constructing reliability aspect models and then applying the weaving mechanism to dynamically combining components and aspects. The effectiveness of reliability assurance strategies is analyzed based on the theories of Petri nets. A case study demonstrates that the approach can simplify design and modeling processes of embedded systems and contribute to improving its quality.

Key words: embedded systems; reliability; design; model; aspect-orientation

摘 要: 针对嵌入式系统的特点,提出一种策略驱动的可靠嵌入式系统建模与分析方法.基于 Petri 网建立嵌入式系统的形式化描述语言,并对设备、计算与物理交互、组件及通信过程等要素进行建模.分析嵌入式系统的主要故障类型和特征,探索嵌入式系统的可靠性保障策略.采用面向方面思想提取可靠性保障策略相关关注点.通过构造关注点模型,并利用编织机制,将关注点模型动态地集成为一个完整的嵌入式系统可靠模型.利用 Petri 网相关理论分析嵌入式系统可靠性保障策略的有效性.具体实例表明,该方法能够简化嵌入式系统的设计与分析过程,有效地提高嵌入式系统的设计质量.

* 基金项目: 国家自然科学基金(60903020, 60773094); 国家科技支撑计划(2009BAH46B01); 上海市曙光计划(07SG32); 上海市科委重点实验室基金(09DZ2272600); 中央高校基本科研业务费专项基金(WH0913009)

收稿时间: 2010-07-10; 修改时间: 2011-03-29; 定稿时间: 2011-04-11

关键词: 嵌入式系统;可靠性;设计;模型;面向方面
中图法分类号: TP311 文献标识码: A

随着计算机、网络通信和控制技术的不断发展,嵌入式系统应用在广度和深度方面不断取得新的突破,使得系统运行具有更快的响应速度、更高的操作精度及更广的环境适应性.计算机与物理设备紧密耦合的复杂嵌入式系统也称为信息物理融合系统(cyber-physical system,简称 CPS)[1],在交通、能源、汽车、航天航空、医疗保健等领域日益得到广泛应用.鉴于嵌入式系统应用的巨大社会效益和经济价值,有关嵌入式系统设计与开发方法的研究已经成为当今国际工业界和学术界共同关注的新热点[2].

嵌入式系统往往工作环境恶劣、受电噪声干扰较大,而且随着软件越来越复杂,系统运行不稳定的现象愈来愈严重,可靠性已经成为衡量潜入式系统优劣的重要因素.然而,嵌入式系统所处的环境灵活多变和网络本身的不确定性,使其在系统的可靠性方面面临许多新的挑战:首先,复杂嵌入式系统包含离散和连续成分,既要实现功能性需求又要满足非功能性需求,呈现出确定性与随机性共在的行为特征[3].嵌入式系统的这些特征要求相应的模型语言需具有丰富的表达能力;其次,嵌入式系统的结构类型多种多样、行为特征纷繁复杂,系统必须具备健壮性和容错性.同时,需要有新的技术建立嵌入式系统软件结构、成分抽象以及组合关系;最后,可靠性保障对于嵌入式系统至关重要,需要研究嵌入式系统软件性质验证方法,用于对不同层次的嵌入式系统模型成分及集成系统进行分析和验证,从而保障嵌入式系统的有效运行.

可靠嵌入式系统的设计需要一套严格的理论和方法支撑.关注点分离(separation of concerns)是软件工程的一条基本原则.面向方面编程(aspect-oriented programming,简称 AOP)[4]是在面向对象方法基础上发展起来的一种新型程序设计方法,其核心是将分散的关注点从对象结构中提炼出来,作为一类元素.可靠性问题是嵌入式系统重要的非功能需求,可以利用 AOP 技术把相关问题作为系统的一个独立方面,方便嵌入式系统的设计与开发.此外,可靠性需求与嵌入式系统功能模块的分离,使得两部分都能够较好地被重用.然而,面向方面的需求分析常常会出现歧义、模糊和二义性的现象.因此,可借助形式化方法对 AOP 规约的需求进行分析,以增加系统模型的语义约束.Petri 网作为一种直观的图形建模工具和一种具有丰富数学基础的形式化模型[5],可以广泛应用于描述和研究并发、异步和分布式特征的系统,并提供一种可操作语义.使用形式化的表示方法可以解释、模拟并刻画可靠嵌入式系统设计与分析过程,规范嵌入式系统的属性,并可以从理论上验证嵌入式系统可靠性策略的有效性,从而增强对目标问题和实体运行机制的管理.

基于上述背景,本文以关注点分离和建模为基本思想,以嵌入式系统的业务流程为核心,提出可靠嵌入式系统的建模与分析方法.针对嵌入式系统的领域特点,重点分析嵌入式系统的设备、计算与物理的交互、组件及通信过程的形式化描述.根据嵌入式系统的可靠性需求,提出相应可靠性保障策略,包括故障输出、设备容错、任务容错和通信容错 4 个横切关注点,利用面向方面规范关注点的行为描述它们内在联系.通过 Petri 网对这些关注点及其横切关系进行建模,编织机制将这些模块动态地集成为一个完整的模型.最后,利用 Petri 网理论分析分析可靠性保障策略的有效性.石化行业物料平衡的应用及实验结果表明,该方法能简化嵌入式系统的建模过程,有效地提高系统的设计质量和可靠性,对开发具有高可靠的嵌入式系统具有重要的理论意义和实用价值.

本文第 1 节提出可靠嵌入式系统设计框架.第 2 节构造嵌入式系统的核心网模型.第 3 节给出可靠嵌入式系统的设计技术.第 4 节通过实例和仿真实验说明方法的有效性.最后是结论和下一步工作.

1 可靠嵌入式系统设计框架

1.1 设计框架的执行流程

可靠嵌入式系统设计框架的具体执行流程如图 1 所示,该方法采用面向方面编程思想对嵌入式系统的需求进行解析,研究嵌入式系统的形式化模型和可靠性保障策略,主要包含 7 个步骤:

步骤 1. 以石化行业的物料平衡应用为驱动进行需求抽象,提取系统的多维关注点集合及其关系.

步骤 2. 基于 Petri 网建立嵌入式系统需求的形式化描述语言,探索相应执行语义以描述系统的离散和连续、动态和静态特性等.

步骤 3. 针对嵌入式系统故障类型和特征,研究可靠性保障策略及其编织规则,不仅要完成关注点的功能,还应刻画关注点间关系.

步骤 4. 对关注点集、关注点之间关系和可靠保障策略进行建模,以形成一个模型库.

步骤 5. 基于核心关注点及其关系构建嵌入式系统的核心网模型,并根据编制规则动态织入横切关注点,以形成系统的可靠模型.

步骤 6. 利用相关理论分析模型的一致性、执行正确性、方面的兼容性.

步骤 7. 采用可靠性保障策略和模型对系统的运行进行动态调整和监控.

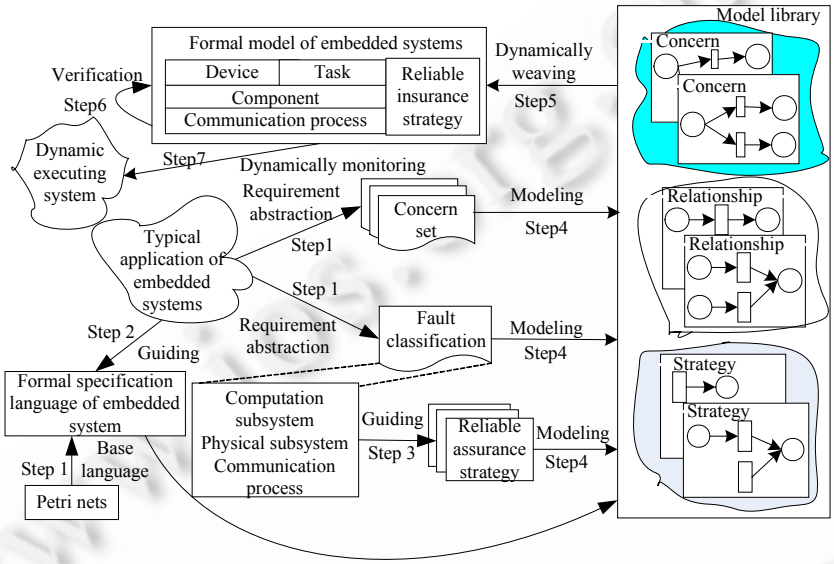


Fig.1 A framework of reliable embedded system design

图 1 可靠嵌入式系统设计框架

1.2 可靠嵌入式系统需求

复杂嵌入式系统结构如图 2 所示,主要包含计算、物理和混合 3 个子系统.计算子系统主要用于支持标准的实时监控和数据处理,物理子系统是嵌入式系统底层的动态物理处理过程,混合子系统则在计算和物理子系统之间构建一座桥梁.每个子系统又包含相应的组件(计算单元)、连接件.其中:组件可以完成系统的特定功能,如物理检测、资源转换等;而连接件用于完成组件间的通信需求,如数据传送、共享等.每个组件、连接件又可划分为一系列相互关联的任务集.为了确保敏感数据的访问,某些任务会有特定的执行条件.

发布/订阅系统具有松耦合、匿名、多对多通信和可扩展等特点,已成为支持新一代网络计算的重要基础中间件平台[6].在发布/订阅系统中,信息的生产者和消费者之间所交互的信息被称为事件.本文假设嵌入式系统主要采用拓扑结构为总线型的发布/订阅系统.下面给出可靠嵌入式系统的需求模型定义.

定义 1. 可靠嵌入式系统的需求模型是一个六元组:$\Xi=\{De,C,HC,Bri,DC,RD\}$:

(1) $De=(DeS,DeA,DeN)$是有限的设备集,其中,DeS,DeA,DeN 分别是传感器、执行器和网络节点集合;

(2) $C=(subC,R,RN,TE,RL,RM)$是有限的组件集,其中:

① $subC=TK\cup SC$ 是有限任务或子组件集,R 是有限资源集,标记 $tk_{i,j},sc_{i,j},r_{i,j}$ 分别表示组件 C_i 的第 j 个任务、子组件和资源;

② $RN:R\rightarrow N^*$是指组件包含资源的数量;

③ TE 是任务的属性函数,$\forall tk_{i,j}\in TK, TE(tk_{i,j})=(I_{i,j},O_{i,j},v_{i,j},ec_{i,j},rt_{i,j})$分别描述任务输入、输出、执行速率(可选,是连续任务的属性)、执行条件(默认为真)和可靠性;

④ $RL:subC\times subC\to\{>,\|,+,\nabla\},>,\|,+,\nabla$分别表示顺序、并行、选择和互斥关系;

⑤ RM 是任务的通信函数,其功能是为任务指明需要发布的事件或者订阅事件的条件;

(3) $HC:C\to\{C,Py,H\}$是组件的属性函数,其中,C,Py,H 分别表示计算、物理和混合;

(4) Bri 是系统的路由表,描述了嵌入式系统的事件传输路径;

(5) $DC:C\to De^*$为物理子系统的组件指定底层设备,若存在计算模型则需要额外提供;

(6) $RD:DeS\cup DeA\to(0,1)$是传感器和执行器的可靠性函数.

嵌入式系统中,各个任务或设备的可靠性是指在规定的时间内,该任务或设备不引起嵌入式系统失效的概率.这里,失效是指嵌入式系统不能完成所需要的功能,具体可靠性分配/分解方法见文献[7,8].记组件 C_i 中发布事件、订阅事件的任务集分别为 $Pm(C_i)$和 $Sm(C_i)$.为了将嵌入式系统的建模和分析过程阐述清楚,本文假设嵌入式系统具有以下特性:每个子组件可以独立看成一个组件,具有相应的任务和资源,需要完成特定的功能;通信链路可靠性均为 r;每个任务不能在未完成时自行挂起;任务一旦将事件发送出去便可以继续运行;任务间的切换开销可忽略;一个组件在同一时刻不可用的同一类型设备(传感器、执行器)不超过∂个(可以根据实际情况进行适当调整);设备从发生故障到恢复正常需要一定时间,在此阶段备份设备不会发生故障.

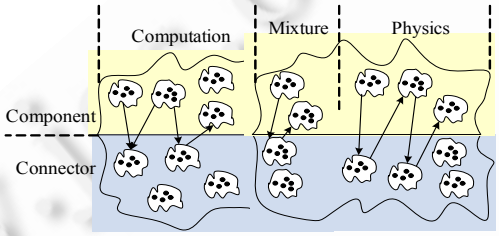


Fig.2 Structure of embedded system

图 2 嵌入式系统结构图

2 嵌入式系统的核心网模型

2.1 组合网

由于嵌入式系统的功能是由多个独立运行的组件构成,根据 AOP 思想可知:嵌入式系统的执行流程为核心关注点;而横切关注点是指系统的非功能需求,本文主要指组件的属性和一些控制措施.如无特殊说明,本文后面所涉及的关注点均指横切关注点.相应地,可以将模型分为组合网、通知网、引入网和方面网.其中:组合网用于描述嵌入式系统的执行流程;通知网和引入网对应横切关注点的切入点;而将组合网和横切关注点所对应的通知网、引入网按照一定的编织规则进行编织,则形成一个方面网.

定义 2. 八元组$\Sigma=(N,h,IO,D,A_T,A_F,\lambda,M_0)$称作基网(base net,简称 BN),其中:

(1) $N=(P,T,F)$是一个基本 Petri 网,其中,P,T,F 分别表示库所、变迁、弧和权的有限集,且两两不相交;

(2) $h:P\cap T\to\{D,C\}$是一个混合函数,将库所和变迁划分为离散节点(集合 P^D 和 T^D)和连续节点(集合 P^C 和 T^C),其中,$P=P^D\cup P^C, T=T^D\cup T^C, P^D\cap P^C=T^D\cap T^C=\varnothing$;

(3) $IO\subset P$ 是一类特殊的库所,称为Σ的接口,并用虚线圆圈表示;

(4) D 为Σ的非空个体集,规定 f_D,f_S 分别为 D 上给定的公式集(布尔表达式)、符号和集,个体集决定了Σ标注中的类型、运算和函数;

(5) $A_T:T\to f_D$,对于 $t\in T, A_T(t)$中的自由变量必须是以 t 为一端有向弧上的自由变量;

(6) $A_F:F\to f_S\times R_0^+$,若$(p,t)\in F$ 或$(t,p)\in F$,则 $A_F(p,t)$或 $A_F(t,p)$为 n 元符号,缺省为空;

(7) $\lambda:T\to N^*\times R_0^+\times(0,1)$是变迁的属性函数,$\forall t_i\in T,\lambda(t_i)=(\alpha_i,\beta_i,\eta_i)$.其中,$\alpha_i$描述 t_i优先级,默认值为 0,本文假设α_i值越低变迁的优先级越高.对于离散变迁,β_i描述 t_i的延迟时间,默认为 0;对于连续变迁,β_i可反映其最大触发速率,即最大触发速率 $v_{\max}=1/\beta_i.\eta_i$描述了 t_i触发的概率;

(8) $M_0:P\to f_S\times R_0^+$是Σ的初始标识.

本文分别采用双圈、单圈、实心条、空心条表示连续库所、离散库所、连续变迁和离散变迁.个体集 D 主要用来描述系统中的资源,如实际资源和数据包.本文把嵌入式系统中的消息数据包统一抽象成个体ε.如无特殊说明,模型中出现的个体均为ε.任意 $x\in(P\cup T)$,集合${}^\bullet x=\{y|y\in(P\cup T)\wedge(y,x)\in F\}$和 $x^\bullet=\{y|y\in(P\cup T)\wedge(x,y)\in F\}$分别对应于 x 的输入和输出.

定义 3. 六元组$\Omega=\{\Sigma,\Gamma,TI,PI,TA,PA\}$称作组合网(composition net,简称 *CN*),其中:

(1) Σ是一个 *BN* 模型,描述了Ω的基本结构;

(2) $\Gamma=\{\Gamma_i|i\in N^*\}$是页的有限集,每一个页是除$\Omega$以外的 *BN* 模型或 *CN* 模型;

(3) $TI\subset T$ 是替代节点的集合,其中每个页面对应一个替代节点;

(4) $PI\subset P$ 是端口节点的集合,端口节点描述替代节点的输入和输出;

(5) $TA:TI\to\Gamma$是一个页分配函数,即为每个替代节点分配具体页面;

(6) *PA* 是端口映像函数,其功能是把替代节点的端口节点映射到对应页面的接口.

要求 *CN* 模型Ω满足如下条件:

(1) 为确保 P^D 的非负性,要求:$\forall p\in P^D,\forall t\in T^C$,若$(p,t)\in F$,则$(t,p)\in F\wedge A_F(p,t)=A_F(t,p)$;

(2) $v_i(\tau)$是变迁 t_i 在τ时刻的瞬时触发速率;

(3) 每个库所 p 中仅包含一种个体.为了区分各个网中的变迁和库所,规定 $N_i.x$ 为网 N_i 中的元素 x.设组合网Ω的关联矩阵为 $IM(\Omega)$.

称映射 $CutN:\{N_i.x_j,N_m.x_n,\ldots,N_f.x_k\}$为系统的一个切入点,其中:*CutN* 分别为切入点的名称;$N_i.x_j,N_m.x_n,\ldots,N_f.x_k$为切入点相应的连接点.三元组 $con=\{CutN,AN,IN\}$称为系统一个关注点,其中,*CutN*,*AN*,*IN* 分别为关注点的切入点集、通知网集和引入网集.通知网和引入网都是一个组合网,分别描述切入点的执行逻辑和需要引入的执行流程.设Ω为编织前的网模型,集合 *con* 为需要织入的关注点集,则编织后方面网Ω'的构造步骤如下:

(1) 根据 *con* 中所有切入点的定义,在切入点处依次把对应引入网添加到Ω中,形成一个新的组合网;

(2) 按照编织规则,增加相应的元素,同时合并相同的元素.

在某时刻τ,各库所中个体的分布状况称为 *CN* 模型的标识,记做 $M^\tau.\forall p\in P,M^\tau(p)=\{(d_i,q_i),\ldots,(d_k,q_k)\}$.其中,$\{d_i,d_j,\ldots,d_k\}$为标识 M^τ下库所 p 中存放的个体集,而$\{q_i,q_j,\ldots,q_k\}$为相应的数量.如无特殊说明,则 $M(p)$等于库所 p 中个体的数量值.记 $t\langle d_1,d_2,\ldots,d_n\rangle$为变迁 t 的替换,替换主要是对 t 的输入/输出弧以及 $A_T(t)$中出现的所有自由变量绑定相应的个体.记 $A_T(t)\langle d_1,d_2,\ldots,d_n\rangle$和 $A_F(p,t)\langle d_1,d_2,\ldots,d_n\rangle$分别描述将个体 $d_1,d_2,\ldots,d_n$ 替换到公式 $A_T(t)$和谓词 $A_F(p,t)$所得到的值.若替换 $t\langle d_1,d_2,\ldots,d_n\rangle$使得 $A_T(t)\langle d_1,d_2,\ldots,d_n\rangle=A_F(t)\langle d_1,d_2,\ldots,d_n\rangle$=true,则称 $t\langle d_1,d_2,\ldots,d_n\rangle$是标识 M^τ下变迁 t 的可行替换.记变迁 t 在标识 M^τ下的所有可行替换集合为 $VP(M^\tau,t)$.

定义 4. 设Ω为 *CN* 模型,M^τ为Ω在τ时刻的标识,$\forall t\in T$,若 $VP(M^\tau,t)\neq\varnothing$,且$\exists t\langle d_1,d_2,\ldots,d_n\rangle\in VP(M^\tau,t)$,使得:

(1) $t\in T^D:(\forall p\in{}^\bullet t,M^\tau(p)\geqslant A_F(p,t)\langle d_1,d_2,\ldots,d_n\rangle)$;

(2) $t\in T^C$:

① $\forall p\in{}^\bullet t\cap P^D,M^\tau(p)\geqslant A_F(p,t)\langle d_1,d_2,\ldots,d_n\rangle$;

② $\forall p\in{}^\bullet t\cap P^C,|M^\tau(p)|\neq\varnothing\vee(\exists t_j\in T^C\cap{}^\bullet p,v_j(\tau)>0\wedge p\notin{}^\bullet t_j)$,

则称变迁 t 在标识 M^τ下有发生权.将标识 M^τ下所有有发生权变迁的集合记为 $ET(M^\tau)$.标识 M^τ下,t_i的触发是有效的当且仅当 t_i 在标识 M^τ下有发生权,且变迁集 $ET(M^\tau)$中不存在比 t_i 优先级高的变迁.将 M^τ下所有可以有效触发的变迁集合记为 $FT(M^\tau)$.在标识 M^τ下,通过有效触发变迁 t_i 的一个可行替换 $t_i\langle d_1,d_2,\ldots,d_n\rangle$到达新标识 M'的过程记为 $M^\tau[t_i\langle d_1,d_2,\ldots,d_n\rangle M'$.将 M^τ下并发的变迁集合称为 M^τ的最大并发集,记为 $MT(M^\tau)$.

集合 $H(M^\tau)=\{t\langle d_1,d_2,\ldots,d_n\rangle|t\in MT(M),t\langle d_1,d_2,\ldots,d_n\rangle\in VP(M,t)\}$称为 M 的一个最大触发集.

定义 5. 设Ω为 *CN* 模型,M^τ为Ω在τ时刻的标识,在时刻$\tau+\omega(\omega>0)$,通过有效触发 $H(M^\tau)$到达新的标识 $M^{\tau+\omega}$,记作 $M^\tau[H(M^\tau)>M^{\tau+\omega}.M^{\tau+\omega}$分别按如下规则计算:$\forall t_i\langle d_1,d_2,\ldots,d_n\rangle\in H(M^\tau)$:

(1) 若 $t_i\in T^D$:$\forall p_j\in {}^\bullet t_i\cup t_i^\bullet$:$M^{\tau+\omega}(p_j)=M^\tau(p_j)-A_F(p_j,t_i)\langle d_1,d_2,\ldots,d_n\rangle+A_F(t_i,p_j)\langle d_1,d_2,\ldots,d_n\rangle$;

(2) 若 $t_i\in T^C$:$M^{\tau+\omega}(p_j)=M^\tau(p_j)-\int_0^\omega v_i(\tau)\mathrm{d}\tau$,向量 $S(M^\tau)=[s_i]$,其中,s_i 对应变迁 t_i.

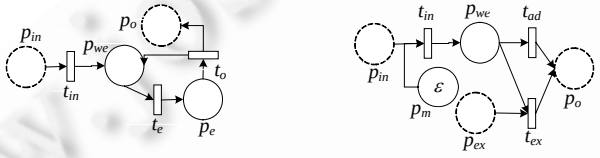
$$s_i=\begin{cases}1, & t_i\in FT(M^\tau)\cap T^D\\ v_i(\tau), & t_i\in FT(M^\tau)\cap T^C\\ 0, & \text{else}\end{cases},$$

则定义 5 可以转换为 $M'=M^\tau+IM(\Omega)\cdot S(M^\tau)$.因此,可以利用关联矩阵分析 *CN* 模型.如果存在触发序列 $H_1,H_2,...,H_k$ 和标识序列 $M_1,M_2,\ldots,M_k$,使得 $M[H_1>M_1[H_2>\ldots M_{k-1}[H_k>M_k$,则称 M_k 是从 M 可达的.记 $R(M)$为 M 的所有可达标识集合,规定 $M\in R(M)$.

2.2 嵌入式系统的核心网模型

(1) 设备的建模

传感器 de_i 的 *CN* 模型 deN_i 如图 3(a)所示,具体执行流程:若得到调用($M(p_{in})\neq\varnothing$)则触发变迁 t_{in} 对设备初始化,并进入运行位置;若运行成功 t_e 则输出采集数据到 p_e;变迁 t_o 的作用是将采集的数据输出,同时传感器重新进入运行位置,变迁 t_o 的延迟时间$\beta(t_o)$等于传感器的采样时间间隔.执行器 de_i 的 *CN* 模型 deN_i 如图 3(b)所示,其中,变迁 t_{ad} 表示调整操作,若有执行指令输入(p_{ex})则调用执行操作 t_{ex}.设置变迁 t_{ex} 的优先级比 t_{ad} 要高.



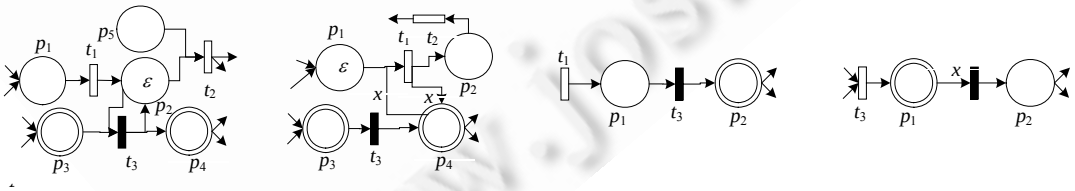
(a) (b)

Fig.3 *CN* model of lower level device

图 3 底层设备的 *CN* 模型

(2) 计算与物理的交互模型

物理和计算的交互作用可分为计算对物理的影响、物理对计算的影响、物理和计算系统的相互转换 4 种,具体的 *CN* 模型如图 4 所示.其中:图 4(a)是计算对物理的影响,假设 $M_0(p_2)=\varepsilon$,$|M_0(p_3)|\neq\varnothing$,且离散变迁的优先级比连续的高.系统通过 p_1 和 p_5 控制连续变迁 t_3 的执行与否,变迁 t_1 的触发会输出个体到库所 p_2;图 4(b)是物理对计算的影响;而图 4(c)和图 4(d)分别是将系统从离散转换成连续和从连续转换成离散.



(a) (b) (c) (d)

Fig.4 *CN* model of interaction

图 4 交互的 *CN* 模型

(3) 组件的建模

C_i 的组合网模型为 CN_i,具体建模过程为:

① 任务的建模:将任务 $tk_{i,j}$ 抽象为等待执行($p_{s,i,j}$),执行($t_{i,j}$)和结果输出($p_{e,i,j}$),变迁 $t_{i,j}$ 的 A_T 函数是根据任务的执行条件设置;

② 子组件 $sc_{i,j}$ 的建模:引入替代节点 $TSC_{i,j}$ 描述 $sc_{i,j}$ 页面,根据其输入/输出接口引入相应端口节点;

③ 若组件 C_i 中包含 n 个设备,则根据设备的建模过程引入设备的替代节点和相应的端口节点;

④ 资源的建模:为组件中每个资源 r 都建立一个库所 p_r 来存放,同时,将 r 抽象成个体 d_r.设 r 的数目为 n,则设置 $M_0(p_r)=(d_r,n)$来表示有 n 个 r 可用.根据任务的属性函数,可以确定任务与资源的对应关系;

⑤ 任务基本关系进行建模:$tk_{i,j}>tk_{i,k},tk_{i,j}\|tk_{i,k},tk_{i,j}\nabla tk_{i,k},tk_{i,j}+tk_{i,k}$ 的模型分别如图 5(a)~图 5(d)所示(这里仅考虑任务的执行流程).其中,设 $RL(tk_{i,g},tk_{i,j})=RL(tk_{i,g},tk_{i,k})=RL(tk_{i,j},tk_{i,f})=RL(tk_{i,k},tk_{i,f})=>$.图 5(c)中,设任务互斥访问资源 r.同样,可以对子组件间、任务与子组件间基本关系进行建模;

⑥ 组件的初始化和结束:引入 t_{in} 和 p_s(输入接口)来描述整个组件的开始操作和开始位置,依据组件的特征对整个系统进行初始化:

$${}^{\bullet}p_s=\varnothing,p_s{}^{\bullet}=t_{in},{}^{\bullet}t_{in}=p_s,t_{in}{}^{\bullet}=\{p_{s,i,j}|\forall sc_{i,k}\in subC(C_i),RL(sc_{i,k},tk_{i,j})\neq>\}\cup\{p_s^{i,j}|\forall sc_{i,k}\in subC(C_i),RL(sc_{i,k},sc_{i,j})\neq>\}.$$

同时,引入 t_e 和 p_e(输出接口)来描述整个组件的结束操作和结束位置:

$$t_e{}^{\bullet}=p_o,{}^{\bullet}p_o=t_e,{}^{\bullet}t_e=\{p_{e,i,j}|\forall sc_{i,k}\in subC(C_i),RL(tk_{i,j},sc_{i,k})\neq>\}\cup\{p_o^{i,j}|\forall sc_{i,k}\in subC(C_i),RL(sc_{i,j},sc_{i,k})\neq>\},p_o{}^{\bullet}=\varnothing.$$

⑦ 集成模型:根据任务、子组件间的基本关系组合对应的模型,同时设置变迁优先级均为 3(最低等级).

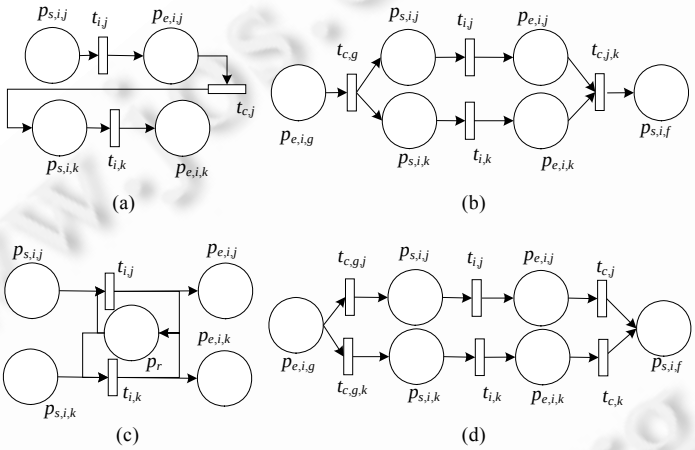


Fig.5 *CN* model of basic relationship among tasks

图 5 任务基本关系的 *CN* 模型

(4) 通信过程建模

本文把事件抽象成个体 d_i.组件间发布和订阅事件的 *CN* 模型分别如图 6 所示,假设任务 $tk_{i,j},\ldots,tk_{v,b}$ 是生产者,任务 $tk_{t,k}$ 是一个消费者.引入库所 p_p 用于存放总线令牌,p_c 用于保存发布的所有事件.对于生产者 $tk_{i,j}$,引入库所 $p_{p,i,j}$ 用于存放发布的事件并等待发布.对于消费者 $tk_{t,k}$,引入变迁 $t_{sb,t,k}$ 将符合条件的事件打包并存放在库所 $p_{we,t,k}$ 中等待网络连接,变迁 $t_{sb,t,k}$ 的 A_T 函数等于事件的订阅条件;若链接成功 $t_{e,t,k}$,则进入路由传输.对于每个路由 ro_i,引入库所 $p_{r,i}$,并将对应路由转发操作抽象成一个变迁 $t_{t,i}$.按照路由表,排在第 1 个路径上的变迁优先级均为 0,排在第 2 个路径上的变迁均为 1,以此类推.

(5) 核心网模型的集成

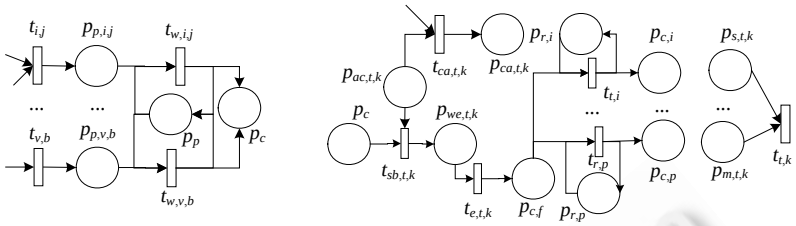
根据嵌入式系统的结构特点,需求模型Ξ的核心网模型Ω构造步骤是:

① 按照底层设备、组件的构造方法,结合任务、设备属性函数,分别构造系统所调用设备、组件的模型;

② 根据组件的通信需求,组合各个组件的模型;

③ 引入变迁 t_s 和库所 p_s 分别描述整个系统的开始操作和开始位置,依据组件的特征对整个系统进行初始

化,使得:${}^{\bullet}p_s=\varnothing,p_s{}^{\bullet}=t_s,{}^{\bullet}t_s=p_s,t_s{}^{\bullet}=\{\ p_s^i\mid\forall C_k\in C,C_i\notin subC_k\wedge Sm(C_i)=\varnothing\}$;引入变迁 t_e 和库所 p_e 分别描述整个系统的结束操作和结束位置,使得:$t_e{}^{\bullet}=p_e,p_e{}^{\bullet}=t_e,{}^{\bullet}t_e=\{\ p_o^i\mid\forall C_k\in C,C_i\notin subC_k\wedge Pm(C_i)=\varnothing\},p_e{}^{\bullet}=\varnothing$.



(a) (b)

Fig.6 *CN* model of communication process

图 6 通信过程的 *CN* 模型

2.3 模型性质

设组件 C_i、子组件 $sc_{i,j}$、设备 de_i 对应的 *CN* 模型分别表示为 $CN_i,scN_{i,j},deN_i$,模型 $scN_{i,j}$ 中的变迁、库所集合分别标记为 $T_{i,j}$ 和 $P_{i,j}$.M^τ是组件 C_i 在τ时刻的标识,则:(1) 函数 $TM(M^\tau,sc_{i,j})$为子组件模型 $scN_{i,j}$ 中所有库所在标识 M 下包含的个体值,称为 M^τ在子组件 $sc_{i,j}$ 上的映射;(2) $\forall M\in R(M^\tau)$,记$\delta(M^\tau,M)$为 M^τ到 M 的触发序列集合;(3) $\forall\delta^k\in\delta(M^\tau,M)$,触发序列 $\delta^k_{sc\to j}=\{t_f\mid t_f\in\delta^k\cap T_{i,j}\}$称为 δ^k 在子组件 $sc_{i,j}$ 上投影序列.若 $M(p_e)=\varepsilon$,则称 M 为模型的一个正常终止标识,即完成所需要的功能.记集合 $M^F(\Omega)$为模型Ω的正常终止标识集合.

定理 1. 设 M^τ是组件 C_i 在τ时刻的标识,$TM(M^\tau,sc_{i,j})$为标识 M^τ在子组件 $sc_{i,j}$ 上的投影,则$\forall M'\in R(M^\tau)$,$\forall\delta^k\in\delta(M^\tau,M')$有 $TM(M^\tau,sc_{i,j})[\ \delta^k_{sc\to j}>TM(M',sc_{i,j})$.

证明:因为 $TM(M^\tau,sc_{i,j})$是标识 M^τ在子组件 $sc_{i,j}$ 上映像,根据函数 TM 的定义可知,$\forall p_s\in P_{i,j}$有

$$M^\tau(p_s)=TM(M^\tau,sc_{i,j})(p_s).$$

所以,$TM(M^\tau,sc_{i,j})$是 $sc_{i,j}$ 在τ时刻的一个标识,任取标识 M^τ下的一个最大触发集 $H_1(M^\tau)$.

因为 $sc_{i,j}$ 是组件 C_i 的一个子组件,根据组件 C_i 的建模过程可知,$\forall t\in T_{i,j},({}^{\bullet}t\cup t^{\bullet})\cap(P_i-P_{i,j})=\varnothing$,所以变迁 t_i 的发生权仅与 $P_{i,j}$ 中库所个体的变化有关,即$\exists TM(M^\tau,sc_{i,j})$的一个最大触发集 $H_2(TM(M^\tau,sc_{i,j}))$,使得

$$H_2(TM(M^\tau,sc_{i,j}))=H_1(M^\tau)\cap T_{i,j}.$$

设 $TM(M^\tau,sc_{i,j})[H_2(TM(M^\tau,sc_{i,j}))>M_1,M^\tau[H_1(M^\tau)>M_2$.

因为 $H_2(TM(M^\tau,sc_{i,j}))=H_1(M^\tau)\cap T_{i,j}$,所以 $M_1=TM(M_2,sc_{i,j})$.依此类推可得:$\forall M'\in R(M^\tau),\forall\delta^k\in\delta(M^\tau,M')$有

$$TM(M^\tau,sc_{i,j})[\ \delta^k_{sc\to j}>TM(M',sc_{i,j}).$$

综上所述,设 M^τ是组件 C_i 在τ时刻的标识,$TM(M^\tau,sc_{i,j})$为标识 M^τ在子组件 $sc_{i,j}$ 上的映射,则$\forall M'\in R(M^\tau)$,$\forall\delta^k\in\delta(M^\tau,M')$有 $TM(M^\tau,sc_{i,j})[\ \delta^k_{sc\to j}>TM(M',sc_{i,j})$.证毕. □

定理 1 说明,可以根据实际需要将嵌入式系统的 *CN* 模型逐层分解到各个子组件上,进而减少分析的复杂度.因此,采用 *CN* 模型对嵌入式系统进行建模和分析是可行的.

执行的正确性映像到模型是指模型运行过程所到达的标识均符合用户的需求.由于核心关注点是嵌入式系统的执行流程,关系到整个系统的运行,因此执行的正确性只需要在核心网上分析即可,以减少分析复杂度.

定理 2. 在核心网Ω中,设集合 $M^F(\Omega)$为模型Ω的正常终止标识集合,$\forall M\in R(M_0)$,若$\exists C_i\in C$ 使得 $M(CN_i{}^{\bullet}p_s)\neq\varnothing$,则有$\forall M'\in M^F(\Omega)\cap R(M),\forall\delta^k\in\delta(M,M'),CN_i{}^{\bullet}t_e\in\delta^k$.

证明:$\forall M'\in M^F(\Omega)\cap R(M)$,根据 C_i 的结构可以分两种情形:C_i 不包含子组件,C_i 包含子组件.

情形 1.C_i 不包含子组件,即 C_i 中仅包含任务或设备.

因为 $M(CN_i{}^{\bullet}p_s)\neq\varnothing\wedge{}^{\bullet}CN_i{}^{\bullet}t_{in}=CN_i{}^{\bullet}p_s$,所以 $CN_i{}^{\bullet}t_{in}\in FT(M)$.

因为 $CN_i \bullet p_s^{\bullet}=CN_i \bullet t_{in}$,所以 M 的所有最大触发集均包含 $CN_i \bullet t_{in}$,任选一个最大触发集 H_1,设 $M[H_1>M_1$.

从模型 CN_i 的构造规则可知,在标识 M_1 下,CN_i 得到初始化,开始执行各个任务.由于在核心网中仅考虑任务的执行流程,因此 CN_i 会一直执行下去,直到所有任务运行结束.此时,$CN_i \bullet t_e$ 具有发生权.

因为 $M' \in M^F(\Omega)$,所以 $FT(M')=\varnothing$,即:$\forall \delta^k \in \delta(M,M'), CN_i \bullet t_e \in \delta^k$.

情形 2.C_i 包含子组件.假设定理 2 对 C_i 中包含的子组件均成立,下面证明在组件 C_i 上命题同样成立.

同理,$CN_i \bullet t_{in} \in FT(M)$且 M 的所有最大触发集均包含 $CN_i \bullet t_{in}$,任选一个最大触发集 H_1,设 $M[H_1>M_1$.

从 CN_i 的构造规则可知,在标识 M_1 下,CN_i 得到初始化.因为定理 2 对 C_i 所包含子组件均成立,即 $\forall sc_{i,j} \in subC_i$, $\exists M_2 \in R(M_1)$ 使得 $M_2(CN_i \bullet p_o^{i,j}) \neq \varnothing$,因此,$CN_i$ 会一直执行直到所有任务和子组件运行结束,即 $CN_i \bullet t_e$ 具有发生权.

同理可得,$\forall \delta^k \in \delta(M,M'), CN_i \bullet t_e \in \delta^k$.证毕. □

从定理 2 可知,核心网模型可以实时感知系统的动态变化.此外,通过设置变迁执行条件来动态控制系统的运行.设嵌入式系统的核心网模型为 Ω,$\forall M_k \in M^F(\Omega)$,任务集合 $ETK(M_k)=\{tk_{i,j} | t_{i,j} \in \delta(M,M_k)\}$ 称为系统一个完整任务集.通过触发一个完整任务集,系统可以到达一个结束标识,即完成功能需求.记所有完整任务集合为 $AET(\Omega)$.

3 可靠嵌入式系统的设计

3.1 嵌入式系统可靠性保障策略

网络环境的异构性、分布式自治等特性决定了嵌入式系统在执行过程中可能会面临基础设施失效、计算系统故障、网络失效等问题.本文基于嵌入式系统特征对系统主要故障进行归纳:

(1) 物理系统故障:传感器检测数据误差较大、传感器或执行器发生故障变成不可用的.

(2) 计算系统故障:组件中某个任务运行失败.

(3) 通信故障:路由节点失效或网络链接中断.

下面根据系统可能发生的故障情况,提出嵌入式系统的可靠性保障策略(假设系统的可靠性要求为 DP),主要分为设备策略、任务策略和通信策略.

(1) 设备策略:设备策略指当设备发生故障时调用备份设备代替执行:

① 若传感器存在计算模型,则对计算模型进行建模预测采集数据,并与实际采集的数据进行对比,一旦误差大于门限 δ(可以根据实际情况调整),则认为该传感器发生故障;同样,执行器发生故障表现为不可用;

② 在每个组件中增加 ∂ 个备份设备.一旦检测到设备生故障,就启用一个可用备用设备.对于传感器,还将当前错误数据校正为理想的数据.若传感器恢复正常,则中断备份传感器重新启用原先传感器;若执行器正常运行,则等待下一次调用.

(2) 任务策略:采用故障屏蔽机制中的主动复制技术对系统中关键任务进行空间冗余计算:

① 由于嵌入式系统每个任务的可靠性都比较高,因此本文主要采用 2-元或 3-元备份;

② 当系统执行任务时,将请求发送给所有任务副本;

③ 系统在接收到请求应答时,随机选择一个执行结果.如果所有副本都运行失败,则说明该任务运行故障.

(3) 通信策略:采用冗余技术实现通信过程的容错.路由节点失效时,则回退到上一步重新选择;若是链接失败则进行重新发送,每个数据包都最多只能发送 y 次(使得 $y=Round(\log_{1-r}(DP^{1/X}))$,其中,$X$ 是系统中任务、订阅请求总数).若一个数据包发送失败,则重新竞争路由;若 y 次都失败,则说明该通信任务运行故障.

多元备份虽然可以提高系统的可靠性,但 CN 模型的状态空间随着备份的增加呈非线性增加.假设通信策略中每个数据包可发送 y 次,下面给出任务 $tk_{i,j}$ 副本数 $BackN(tk_{i,j})$ 的计算步骤(设 $BackN(tk_{i,j})$ 的初始值为 0, $AE=AET(\Omega)$):

(1) 若 $AE\neq\varnothing$则$\forall ETK_k\in AE,\forall tk_{i,j}\in ETK_k$,执行步骤(2)和步骤(3);

(2) 计算阈值 $z,N_p=|ETK_k\cap Sm(\Omega)|$,则 $z=\left(\frac{DP}{(1-(1-r)^y)^{N_p}}\right)^{\frac{1}{|ETK_k|}}$;

(3) 计算副本数 $BackN(tk_{i,j})$: $BackN(tk_{i,j})=\max\{Round(\log_{1-rt_{i,j}}z),BackN(tk_{i,j})\}$,$AE=AE-ETK_k$.

其中,$Round(x)$是对 x 向上取整的函数.一直执行到所有任务的副本数都计算完毕.

根据任务的属性及其副本数,可计算完整任务集的可靠性:$\forall ETK_k\in AET(\Omega),N_p=|ETK_k\cap Sm(\Omega)|$,则 ETK_k 对应的执行流程可靠性等于 $FR(ETK_k)=\prod_{t_{i,j}\in ETK_k}\left(1-(1-rt_{i,j})^{BackN(tk_{i,j})}\right)\times\prod_{t_{i,j}\in ETK_k\cap Sm(\Omega)}(1-(1-r)^y)$.

3.2 嵌入式系统可靠性保障策略的切入

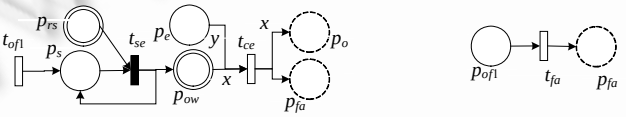
下面采用 CN 对可靠性保障策略中故障输出关注点 con_{of}、设备容错关注点 con_{de}、任务容错关注点 con_{tk} 和通信容错关注点 con_{cf}进行建模.在核心网上织入$\{con_{of},con_{de},con_{tk},con_{cf}\}$得到的模型称为可靠模型$\Omega_s$.从 AOP 的原理可知,通知网包含在引入网中,因此下面对横切关注点进行建模时仅给出引入网模型.

(1) 故障输出关注点 con_{of}

故障输出切入点:

t_{of1}:$\{deN_i\bullet t_{in}|de_i\in DeS\}$,$p_{o1f}$:$\{deN_i\bullet p_{we}|de_i\in DeS\cup DeA\}\cup\{CN_i\bullet p_{s,i,j}|tk_{i,j}\in TK\}\cup\{CN_i\bullet p_{we,t,k}|tk_{t,k}\in Sm(C_i)\}$.

引入网如图 7 所示.切入点 t_{of1} 用于输出传感器检测资料误差较大的故障,根据计算模型引入 p_{rs}, t_{se} 和 p_{ow} 描述资源的理想变化过程;变迁 t_{ce} 对预测数据和传感器采集的数据进行对比.因此,变迁 t_{ce} 的 A_T 函数为$|x-y|>\delta$.切入点 p_{of1} 用于输出设备不可用、任务运行失败或发送事件时链接失败的故障信息.con_{of} 的编织规则为:针对每个任务、设备和订阅请求,分别引入故障输出的切入点及其引入网;同时,为了及时输出故障,新增加变迁优先级均为 0.



(a) (b)

Fig.7 Modeling concerns of fault outputting

图 7 故障输出关注点的建模

(2) 设备容错关注点 con_{de}

设备容错切入点定义为

p_{de1}:$\{deN_i\bullet p_o|de_i\in DeS\}$,$p_{de2}$:$\{deN_i\bullet p_{we}|de_i\in DeS\}$,$p_{de3}$:$\{deN_i\bullet p_{ex}|de_i\in DeA\}$,$p_{de4}$:$\{deN_i\bullet p_m|de_i\in DeA\}$.

对应引入网如图 8(a)、图 8(b)所示.p_{de1},p_{de2} 分别用于传感器的备份执行和重启传感器过程.p_{de3},p_{de4} 分别用于执行器的备份执行和重新启用过程.其中,变迁 t_{bad},t_{bex} 分别表示备份执行器的调整和执行操作.con_{de} 的编织规则:在每个 C_i 中引入库所 p_{bsp},p_{bap},分别表示组件中未被调用备份传感器和执行器个数.根据设备的特性,动态编织设备故障切入点,设在 M 下 de_j 发生故障($M(deN_j\bullet p_{fa})=\varepsilon$),则依次做如下操作:① 若 de_j 是传感器,则织入 p_{de1} 和 p_{de2};否则,织入 p_{de3} 和 p_{de4};② 在设备所属 C_i 的 CN 模型 CN_i 中增加故障处理模块,如图 8(e)所示.

(3) 任务容错关注点 con_{tk}

任务容错切入点定义为 t_{tk}:$\{CN_i\bullet t_{i,j}|tk_{i,j}\in subC_i\}$,$p_{tk}\{CN_i\bullet t_{i,j}|tk_{i,j}\in subC_i\}$,对应引入网如图 9(a)、图 9(b)所示.切入点 t_{tk} 描述任务的执行操作,其中,对于每个副本 $tk_{i,j}^k$,引入 $p_{s,i,j,k}$,$p_{f,i,j,k}$,$p_{o,i,j,k}$ 分别描述备份等待执行位置、执行失败位置和执行输出位置.库所 p_{we} 用于存放执行结果,若存在某个备份执行成功,则调用 t_e 来随机选择一个结果输出.con_{tk} 的编织规则:根据 C_i 的实际需求在 $tk_{i,j}$ 上依次切入 t_{tk} 和 p_{tk},引入的变迁优先级均为 0.

(4) 通信容错关注点 con_{cf}

通信容错切入点定义为 p_{cf}:$\{CN_i \bullet p_{we,t,k} | tk_{t,k} \in Sm(C_i)\}$,对应引入网如图 9(c)所示.若链接失败 t_r 且还可以重试(p_p 含有个体)则重试;若重试 y 次都失败,则输出失败信息.编织规则为:每次织入时,需要根据特定的请求对引入的变迁和库所进行具体命名,且变迁 t_r 与对应的链接成功操作优先级相等.

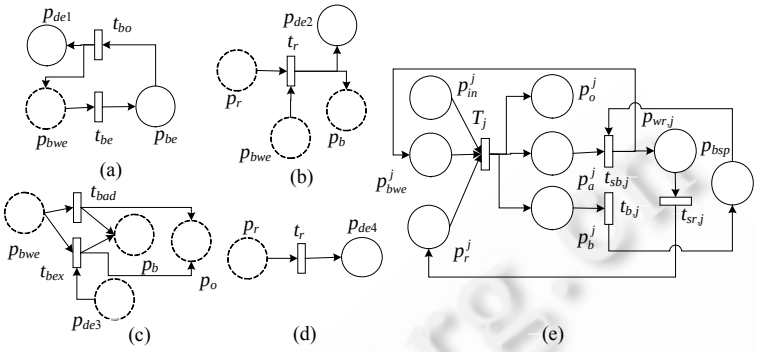


Fig.8 Modeling concerns of the device's fault

图 8 设备故障关注点的建模

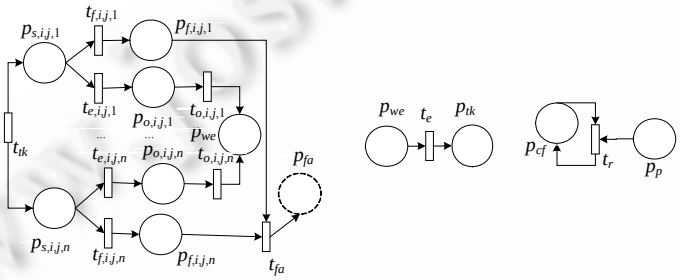


Fig.9 Modeling concerns of the task's fault

图 9 任务容错关注点的建模

3.3 可靠性保障策略的有效性

由于可靠模型Ω_s的状态空间会有一些特殊的标识,设 M 为Ω_s的一个可达标识:若 $FT(M)=\varnothing$,则称 M 为Ω_s的终止标识,记 $TS(\Omega_s)=\{M|M\in R(M_0)\wedge FT(M)=\varnothing\}$;若 $M\in TS(\Omega_s)\wedge M(p_e)=\varepsilon$,则称 M 为Ω_s的正常终止标识,否则称为失败终止标识.记 $TS^E(\Omega_s)=\{M|M\in TS(\Omega_s)\wedge(M(p_e)=\varepsilon)\}$,$TS^F(\Omega_s)=TS(\Omega_s)-TS^E(\Omega_s)$.

本文将可靠性保障策略抽象成多个横切关注点,在核心网模型运行过程中动态地织入.同一个关注点的不同切入点,在织入过程中可能涉及相同位置,相互之间会互相干扰.因此,有必要分析各个横切点之间的兼容性.

定理 3. 设嵌入式系统的核心网模型为Ω,con_{de},con_{tk}分别是设备容错关注点和任务容错关注点,$cut1$,$cut2$分别为两个切入点:

(1) $\forall cut1,cut2\in con_{de},\forall de_i\in De$,设在 deN_i 上前后织入 $cut1$,$cut2$ 得到 $deN^{1,2}$ 和前后织入 $cut2$,$cut1$ 得到 $deN^{2,1}$,若 $M_1(deN_i \bullet p_{in})=\varepsilon$,则有 $R(M_1,deN^{1,2})=R(M_1,deN^{2,1})$;

(2) $\forall tk_{i,j}\in TK$,设在 $tk_{i,j}$ 的模型前后织入 t_{tk},p_{tk} 得到 $tk^{1,2}$ 和前后织入 p_{tk},t_{tk} 得到 $tk^{2,1}$,若 $M_1(C_i \bullet p_{s,i,j})=\varepsilon$,则有 $R(M_1,tk^{1,2})=R(M_1,tk^{2,1})$.

证明:因为切入点织入的时序不同不会影响到组合网的结构,所以模型 $deN^{1,2}$,$deN^{2,1}$ 中包含的变迁、库所、弧及其属性相同,下面分别分析 $deN^{1,2}$,$deN^{2,1}$ 的可达标识.

因为 $M_1(deN_i\bullet p_{in})=\varepsilon$,所以 $M_1(deN^{1,2}\bullet p_{in})=\varepsilon$.

因为 ${}^\bullet deN^{1,2}\bullet t_{in}=deN^1\bullet p_{in}\wedge deN^1\bullet p_{in}{}^\bullet=deN^{1,2}\bullet t_{in}$,所以 $deN^{1,2}\bullet t_{in}\in MT(M_1)$,即 M_1 下的最大触发集都包含 $deN^{1,2}\bullet t_{in}$,任选一个最大触发集 H_1,设 $M_1[H_1>M_2$.

根据传感器可能发生的情形:采集数据误差大于门限、变成不可用、执行成功.

- 当数据误差大于门限时,由于 t_{ce} 的优先级高于 t_o,且 $A_T(t_{ce})$=true,故系统会触发 t_e,t_{se},t_{ce},并输出故障 p_{fa};
- 当传感器变成不可用时,系统会触发变迁 t_{fa},并输出故障 p_{fa};
- 当传感器执行成功时,虽然 t_{ce} 的优先级高于 t_o,但 $A_T(t_{ce})$=false,所以系统会触发变迁 t_e,t_{se},t_o,并输出结果 p_o.

同理可以分析 $deN^{2,1}$ 的执行语义等价于 $deN^{1,2}$,所以 $R(M_1,deN^{1,2})=R(M_1,deN^{2,1})$.

同理,根据组合网的运行语义可得子命题(2)成立.证毕. □

根据定理 3 可知,本文提出的编织规则可以确保各个横切点之间的兼容性,使得切入后模型保持一致性和不变性.因此,将可靠性保障策略切入到嵌入式系统的核心网具有可行性.下面分析可靠性保障策略的有效性.

定理 4. 设需求模型Ξ对应的可靠模型为Ω_s,其中,$\exists tk_{i,j}\in TK,\Omega_s$ 在 $tk_{i,j}$ 上织入任务容错关注点,设 M_0 是Ω_s 的初始标识,则:

(1) $\forall M\in TS^E(\Omega_s),\forall\delta^k\in\delta(M_0,M),\forall C_i\in C,CN_i\bullet t_{fa}\notin\delta^k$;

(2) $\forall M\in R(M_0),\forall\delta^k\in\delta(M_0,M),\exists de_i\in De,deN_i\bullet t_{fa}\in\delta^k$,则$\exists M'\in TS^E(\Omega_s)$使得 $M'\in R(M)$;

(3) $\forall M\in R(M_0),\forall\delta^k\in\delta(M_0,M)$,若$\exists t_{f,i,j,g},t_{f,i,j,h}\in T_i$,使得 $t_{f,i,j,g}\in\delta^k,t_{f,i,j,h}\notin\delta^k$,则$\exists M'\in TS^E(\Omega_s)$,使得 $M'\in R(M)$.

证明:(1) 反证法.设 M_0 到 M 之间可能经过的标识集合 $rm=\{M_1,M_2,\ldots,M_n\}$,且设$\forall M\in TS^E(\Sigma),\forall\delta^k\in\delta(M_0,M)$,$\exists C_i\in C,CN_i\bullet t_{fa}\in\delta^k$.

因为 $CN_i\bullet t_{fa}\in\delta^k$,所以$\exists M_i\in rm$,使得 $M_g(CN_i\bullet p_{fa})=\varepsilon$,且$\exists C_j\in C,C_i\in subC_j$.

又因为 $M_g(CN_i\bullet p_{fa})=\varepsilon$,所以 $M_g(CN_i\bullet p_o)=\varnothing$.根据定理 2 可知,$\forall M'\in R(M_g),M'(CN_j\bullet p_o)=\varnothing$.以此类推,一直往上直到最顶层可得 $M'(p_e)=\varnothing$,所以$\forall M'\in R(M_g),M'\notin TS^E(\Omega_s)$.

因为 $M\in R(M_g)$,所以 $M\notin TS^E(\Omega_s)$,与前提矛盾.

所以假设不成立,即$\forall M\in TS^E(\Omega_s),\forall\delta^k\in\delta(M_0,M),\forall C_i\in C,CN_i\bullet t_{fa}\notin\delta^k$.

同理可证子命题(2)和子命题(3)成立.证毕. □

定理 4 说明,若Ω_s 处于某个正常结束位置,则系统可以完成所有组件功能;当某个设备发生故障时,系统仍然可以继续运行并正常结束;当某个任务副本运行失败时,只要其余一个副本运行成功,系统就可以继续运行.

定理 5. 设需求模型Ξ对应的可靠模型为Ω_s,系统的可靠性要求为 DP,根据任务策略计算出任务 $tk_{i,j}$ 的副本数为 $BackN(tk_{i,j})$,则$\forall ETK_k\in AET(\Omega),FR(ETK_k)\geqslant DP$.

证明:$\forall ETK_k\in AET(\Omega)$,设 $N_p=|ETK_k\cap Sm(\Omega)|$, $z=\left(\dfrac{DP}{(1-(1-r)^y)^{N_p}}\right)^{\frac{1}{|ETK_k|}}$,

因为$\forall tk_{i,j}\in TK$,采取 $BackN(tk_{i,j})$个副本后,整个任务的可靠性为 $1-(1-rt_{i,j})^{BackN(tk_{i,j})}$,

因为 $BackN(tk_{i,j})=\max\{Round(\log_{1-rt_{i,j}}z),BackN(tk_{i,j})\}$,所以 $BackN(tk_{i,j})\geqslant Round(\log_{1-rt_{i,j}}z)$,即

$$1-(1-rt_{i,j})^{BackN(tk_{i,j})}\geqslant z.$$

又因为 $FR(ETK_k)=\prod_{t_{i,j}\in ETK_k}\left(1-(1-rt_{i,j})^{BackN(tk_{i,j})}\right)\times\prod_{t_{i,j}\in ETK_k\cap Sm(\Omega)}(1-(1-r)^y)$,

所以 $FR(ETK_k)\geqslant z^{|ETK_k|}\times\prod_{t_{i,j}\in ETK_k\cap Sm(\Omega)}(1-(1-r))^y=\dfrac{DP}{(1-(1-r)^y)^{N_p}}\times\prod_{t_{i,j}\in ETK_k\cap Sm(\Omega)}(1-(1-r)^y)$.

因为 $N_p=|ETK_k\cap Sm(\Omega)|$,所以 $\dfrac{DP}{(1-(1-r)^y)^{N_p}}\times\prod_{t_{i,j}\in ETK_k\cap Sm(\Omega)}(1-(1-r)^y)=DP$, 即 $FR(ETK_k)\geqslant DP$.

综上所述,$\forall ETK_k\in AET(\Omega),FR(ETK_k)\geqslant DP$.证毕. □

定理 5 说明,任务策略可以在保证系统满足高可靠性要求.同理可得,嵌入式系统采用任务策略后再使用通信策略可以确保系统的高可靠性.因此,本文中阈值的设置方法可以控制任务的副本数、数据包的发送次数.

4 实例仿真

本文以正在开发的石化行业物料平衡系统为例,来说明嵌入式系统的建模与分析过程.物料平衡流程包含下列组件:组件 C_1 用于执行原油蒸馏工艺,并产生煤油、灯油、蜡油和渣油;组件 C_2 用于执行蜡油裂化工业以产生催化汽油;组件 C_3 用于实现渣油焦化工业,并产生焦化石油和石油焦;C_4 跟踪各个油罐的存储量;平衡组件 C_5 准确跟踪原油、各种半成品、成品的库存和生产工艺;生产控制组件(C_6)包含生产成本计算(C_7)和生产量分配(C_8).其中:C_1,C_2,C_3 是 C_4 的子组件,且 C_6,C_7,C_8 是计算子系统的组件;其余为物理子系统的组件.

- C_1:$tk_{1,1}>tk_{1,2}>(tk_{1,3}||tk_{1,4}||tk_{1,5}||(tk_{1,6}+tk_{1,7})||tk_{1,8})$;
- C_2:$tk_{2,1}>tk_{2,2}>tk_{2,3}$;
- C_3:$tk_{3,1}>tk_{3,2}>(tk_{3,3}||tk_{3,4})$;
- C_4:$tk_{4,1}>(tk_{4,2}||tk_{4,3}||tk_{4,4}||tk_{4,5}||tk_{4,6}||tk_{4,7}||tk_{4,8}||tk_{4,9}||tk_{4,10}||tk_{4,11})>t_{4,12}$;
- C_5:$(C_4||C_1>(C_2||C_3))>tk_{5,1}>tk_{5,2}>(tk_{5,3}+tk_{5,4})>(tk_{5,5}||tk_{5,6})>tk_{5,7}$;
- C_6:$tk_{6,1}>C_7>C_8>tk_{6,2}$;
- C_7:$tk_{7,1}>tk_{7,2}>tk_{7,3}$;
- C_8:$tk_{8,1}>tk_{8,2}>tk_{8,3}>(tk_{8,2}||tk_{4,3}||tk_{4,4}||tk_{4,5})$.

通信过程主要有:变迁 $t_{1,2},t_{2,2},t_{3,2},t_{4,12}$ 发布的事件经由 $t_{5,1}$ 订阅;$t_{5,7}$ 发布的事件由 $t_{7,1}$ 消耗;$t_{7,3}$ 发布事件则由 $t_{8,2}$ 消耗.同时假设:每个生产过程都会产生气体和造成一定损耗;每种油对应一个油罐,每个油罐都有一个传感器用于采集存储量;每次生产过程都引入传感器采集各种产品的产出与消耗;每个任务、传感器和链接的可靠性均为 99.5%,传感器修复时间为 10m;流量变量直接由仪表测量,数据由现场 DCS 经接口送往实时数据库,平均 18s 采集一次(传感器),而计算模型由化验室分析(根据上一次数据校正结果),直接输入到需求中进行管理, 4h 更新一次;假设每个传感器采集的数据与预测值的误差应该不超过预测值的 50%;每个事件的传递都有一个专门的路由.根据本文所提出的方法,首先构建物料平衡的核心网模型,然后织入可靠性保障策略的不同关注点以形成可靠模型.其中,蜡油油罐的传感器和组件 C_6 的可靠模型分别如图 10 所示.根据核心网模型的执行语义可知,核心网可以刻画物料平衡的执行流程,并满足执行的各种约束.若没有采取可靠性保障策略,系统的可靠性为 77.0549%;根据可靠模型,可计算出在系统执行 1 小时的可靠性为 99.9994%.采用可靠性保障策略能有效地提高系统的可靠性.

设备容错关注点提出对于组件设置备份传感器来代替故障传感器执行,而备份传感器会增加生产成本和管理.因此,有必要基于历史数据分析物料平衡中组件需设置多少个备份传感器比较合适.下面以组件 C_3 焦化过程中半个小时采集渣油 $x1$、焦化石油 $x2$ 和石油焦 $x3$ 的数据为基础(如图 11(a)所示),研究 C_3 中所需的备份传感器,其中,$x1p$,$x2p$,$x2p$ 标注预测值,$x1c$,$x2c$,$x3c$ 标注采集值,单位为 10^{-3}/KG.S^{-1}.从实际 0.5 小时采集的数据可得,在 0.5 小时内,只有两个采集的数据出现故障,分别为原料渣油第 16 个采集点(4.8m)和焦化石油的第 60 个采集点(18m).两次故障时间间隔超过 10m(13.2m),因此当第 2 个故障发生时,渣油传感器已经重新可用,所以只要引入一个备份传感器即可.

为了评估方法的有效性,本文设计仿真实验并评估方法的效果.首先,随机生成 960 个任务构成嵌入式系统的任务资源.每个任务至少包含任务名称、任务功能等基本信息,同时产生 320 个传感器和 120 个链接作为嵌入式系统的物理资源.如无特殊说明,下面仿真中的物料平衡均包含 48 个任务,16 个传感器,6 个通信过程.

实验 1 的目的是说明任务容错关注点的有效性,假设系统的可靠性要求为 99.9%.具体实验步骤如下:

(1) 取 480 个任务,160 个传感器,60 个链接属性分为 10 组,每组对应为一个物料平衡,依次对每组做步骤 2、步骤 3;

(2) 根据核心网的集成步骤,构造物料平衡的核心网模型,计算出对应的完整任务集;

(3) 根据任务执行副本数的计算公式,分别计算每个任务的副本数,并计算整个物料平衡的可靠性.

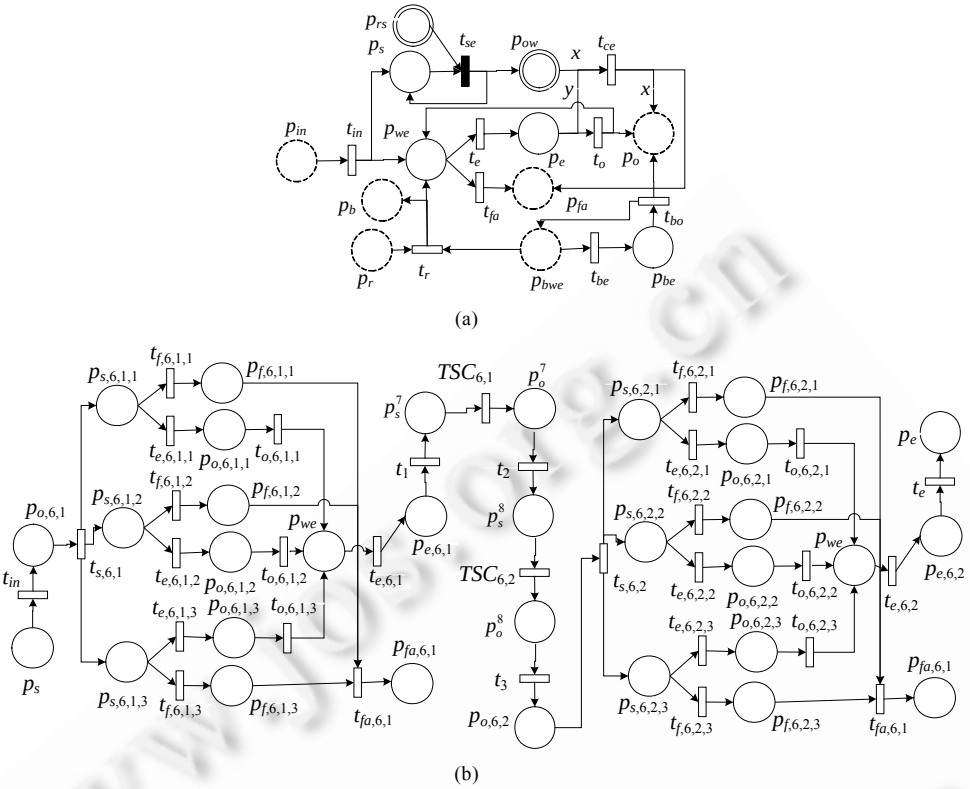


Fig.10 Resource balancing system model

图 10 物料平衡系统模型

实验 1 的实验结果如图 11(b)所示,其中,*R* 表示用户的可靠性要求,2-*B* 是每个任务采取二元备份情况下物料平衡的可靠性.分析该图可知:

(1) 当所有任务都执行 2 元备份时,10 组物料平衡的可靠性都得到不同程度的提高,但仍然不能满足高可靠性要求;

(2) 采用可靠性保障策略可以确保系统的可靠性得到满足,10 组物料平衡平均可靠性为 99.9752%;

(3) 分析仿真结果发现,需要执行 3 元备份的任务数仅占总数的 40%以下.因此,采用该方法不仅可以满足用户可靠性要求,且可减少任务副本,进而间接减少模型的状态空间.

实验 2 的目的是说明采用可靠性保障策略的有效性,具体实验步骤如下:

(1) 取 480 个任务,160 个传感器,60 个链接属性分为 10 组,每组对应一个物料平衡,依次对每组做步骤(2);

(2) 计算物料平衡在下列情况下的可靠性:没有采用任何策略 *R*1,单独采用任务容错 *R*2、通信容错 *R*3、设备容错 *R*4 和采用可靠性保障策略 *R*5.

实验 2 的实验结果如图 11(c)所示,分析该图可知:

(1) 虽然任务的可靠性均较高(任务的平均可靠性为 99.486%),但是系统的可靠性仍然较低.10 组物料平衡可靠性分布在 70%~80%,远远不能满足高可靠性要求;

(2) 单独采用任务容错、设备容错、通信容错都或多或少地提高了系统可靠性,但总体还未能满足要求.

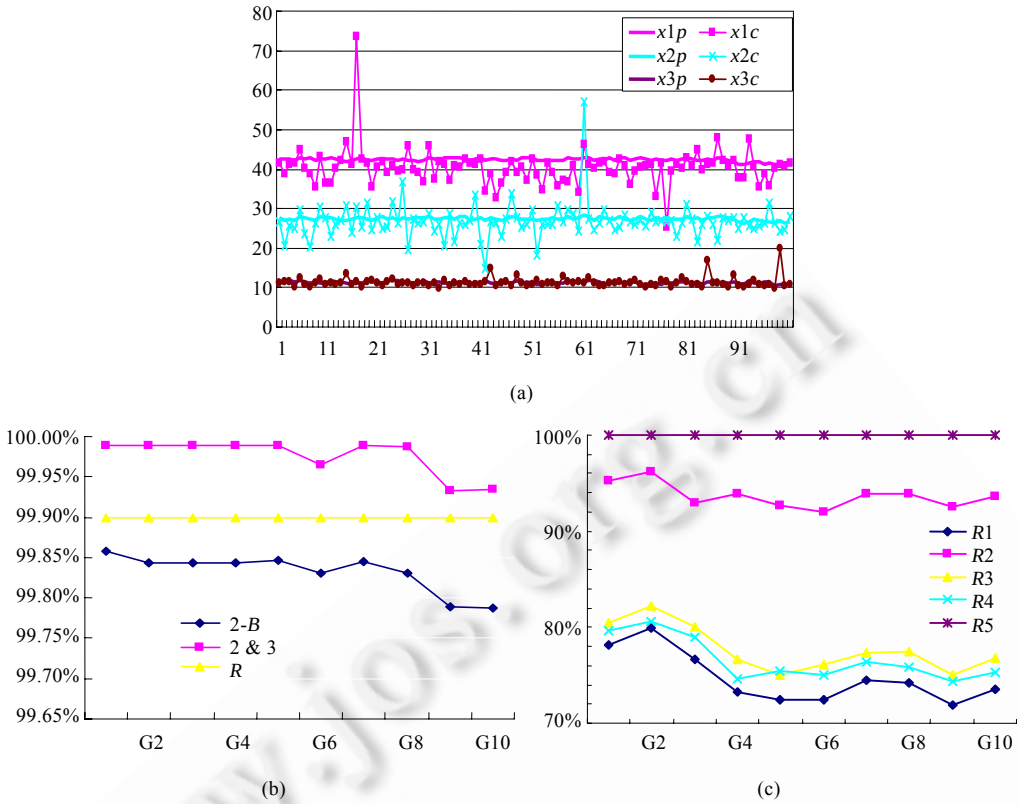


Fig.11 Simulation results

图 11 仿真结果

5 相关工作

形式化方法是保证嵌入式系统正确性及提高嵌入式系统可靠性的重要手段.文献[9]使用 Petri 网对嵌入式系统进行建模与验证,并采用模型检验分析系统的性质.文献[10]采用 VDM++对分布式实时嵌入式软件进行规约,并运用 VDM 验证工具来验证系统的性质.文献[11]给出了关注点之间横切关系的形式化描述和横切关注点之间时序冲突的形式定义,并在此基础上给出了时序冲突分析的步骤.这些工作采用形式化方法对不同的领域进行分析,但没有涉及到嵌入式系统的动态和静态特性、物理和计算控制,对系统非功能属性及离散与连续的特性也没有考虑.

在对系统的可靠性进行分析方面也有一些研究工作,文献[12,13]提出使用 Markov 链对系统可靠性进行分析的数学模型,并通过系统中构件与构件间交互的情况来对可靠性进行分析.这些方法通常是假设系统中各组件间的关系是已知的,而没有考虑组件间的交互情况.本文使用 Petri 网建模可以准确地刻画系统中各个组件的关系,为可靠性分析提供基本的支持.在基于模型的故障诊断方法来提高系统的可靠性也取得一定的进展,针对错误诊断中难以协调各个组件的问题,文献[14]提出使用协调器对各个组件的消息进行分析,从而发现和定位网络中的错误组件.Ardissono 等人提出一种基于错误诊断的框架[15],该框架可识别引起异常的原因并采取相应的处理措施.这些方法在对故障进行处理时可以得到一定的效果,但是它们只能在运行时才能发现错误,对相关的特性也很难进行推导.本文提出面向方面扩展,可以更好地刻画嵌入式系统的可靠特性,灵活处理嵌入式系统的执行状态,并规范可靠性的方面及横切关注点,对系统的切入点作灵活的处理.此外,本文从理论上证明可靠性保障技术的有效性,确保系统的可靠性满足需求,从而方便对嵌入式系统进行设计与开发.

与本文相近的工作还有文献[16,17],它们为本文的工作提供了一定的依据.文献[16]利用形式化的面向方面的方法对安全软件架构进行分析,并验证方法的正确性,但没有考虑方面的分离及方面模型之间的依赖性.文献[17]使用时态逻辑来描述和分析 CPS 系统软件结构的时间约束,并采用模型检验的方法来验证系统的性质.该方法没有考虑系统的容错特性,而容错对于复杂嵌入式系统的正确执行具有重要的作用.Petri 网被用于嵌入式系统的自动抽象和验证[18],提出对 CPS 系统的各个组成部分进行抽象的方法.该方法没有考虑系统的连续与离散信号,对系统的计算与物理的交互、组件及通信过程也没有涉及.文献[19]提出一个CPS系统的形式化框架,研究面向用户控制的 CPS 系统逻辑描述和动态性质分析.该框架考虑了物理设备的交互、计算及通信,但该方法不易于扩展.本文提出的方法可以根据实际需要灵活扩展,并规范系统的方面及横切关注点.

6 结束语

在嵌入式系统设计开发过程中,尤其是早期对系统可靠性进行分析具有重要意义.根据分析的结果及时对设计中不合理的地方进行调整,避免由于设计不合理导致软件最终实现不符合需求的情况,从而减少修改的代价,提高开发的效率.与现有工作相比,应用本文所提出方法可以达到如下效果:(1) 正确刻画了嵌入式系统动态和静态特性、物理和计算控制,采用 Petri 网对嵌入式系统的设备、计算与物理的交互、组件等进行建模,所构造的模型可以确保嵌入式系统正确运行;(2) 组合过程的灵活性,在组合过程中可以根据实际需求对组件、设备进行增减,并可以动态织入可靠性保障的相关技术;(3) 给出嵌入式系统的可靠性保障策略,采用面向方面思想动态织入相关关注点,并基于模型分析可靠性保障策略的有效性,确保系统满足实际的需求;(4) 面向方面编程思想的优越性,在分析嵌入式系统功能够属性时,只需采用核心网.仅在涉及到可靠性分析时才用到可靠性模型,从而减少分析的复杂度.因此,该方法能简化可靠嵌入式系统设计与分析的过程,有效提高系统的设计质量.

面向方面软件开发具有易理解、更加模块化以及更易于复用等特性.目前,我们正着手完善 Petri 网及其面向方面扩展的形式化语法和语义,并进一步研究嵌入式系统的行为特征.同时,开发相应的支撑工具,形成一套更具有实际应用价值的分析方法.

致谢 感谢匿名评阅人对本文工作提出的宝贵意见和建议.

References:

[1] Lee EA. Cyber physical systems: Design challenges. Technical Report, UCB/EECS-2008-8, Berkeley: EECS Department, University of California, 2008.

[2] Adler R, Schaefer I, Trapp M, Poetzsch-Heffter A. Component-Based modeling and verification of dynamic adaptation in safety-critical embedded systems. ACM Trans. on Embedded Computing Systems, 2010,10(2):20.1−20.39. [doi: 10.1145/1880050.1880056]

[3] Zhao F, Bailey-Kellogg C, Fromherz MPJ. Physics-Based encapsulation in embedded software for distributed sensing and control applications. Proc. of the IEEE, 2003,91(1):40−63. [doi: 10.1109/JPROC.2002.805819]

[4] Kiczales G, Lamping J, Mendhekar A, Maeda C, Lopes C, Loingtier JM, Irwin J. Aspect-Oriented programming. In: Aksit M, Matsuoka S, eds. Proc. of the European Conf. on Object-Oriented Programming. LNCS 12412, Heidelberg: Springer-Verlag, 1997. 220−242. [doi: 10.1007/BFb0053381]

[5] Girault C, Valk R. Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications. Berlin: Springer-Verlag, 2003.

[6] Ma JG, Huang T, Wang JL, Xu G, Ye D. Underlying techniques for large-scale distributed computing oriented publish/subscribe system. Journal of Software, 2006,17(1):134−147 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/134.htm [doi: 10.1360/jos170134]

[7] Shooman ML. Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design. New York: John Wiley & Sons, Inc., 2002.

[8] Musa JD. Software Reliability Engineering. New York: Osborne/McGraw-Hill, 1998.

[9] Cortés LA, Eles P, Peng ZB. Modeling and formal verification of embedded systems based on a Petri net representation. Journal of Systems Architecture, 2003,49(12-15):571−598. [doi: 10.1016/S1383-7621(03)00096-1]

[10] Verhoef M, Larsen PG, Hooman J. Modeling and validating distributed embedded real-time systems with VDM++. In: Misra J, Nipkow T, Sekerinski E, eds. Proc. of the Formal Methods. LNCS 4085, Heidelberg: Springer-Verlag, 2006. 147−162. [doi: 10.1007/11813040_11]

[11] Zhang LL, Ying S, Ni YC, Zhao K, Wen J. An analysis approach for software architectural concerns. Chinese Journal of Computers, 2009,32(9):1782−1791 (in Chinese with English abstract).

[12] Trung PT, Thang HQ. Building the reliability prediction model of component-based software architectures. Int'l Journal of Information Technology, 2009,5(1):18−25.

[13] Wang WL, Pan D, Chen MH. Architecture based software reliability modeling. Journal of Systems and Software, 2006,79(1): 132−146. [doi: 10.1016/j.jss.2005.09.004]

[14] Li YM, Ye LN, Dague P, Melliti T. A decentralized model-based diagnosis for BPEL services. In: Proc. of the 21st IEEE Int'l Conf. on Tools with Artificial Intelligence. Arras: IEEE Computer Society, 2009. 609−616. [doi: 10.1109/ICTAI.2009.77]

[15] Ardissono L, Furnari R, Goy A, Petrone G, Segnan M. Fault tolerant Web service orchestration by means of diagnosis. In: Proc. of the 3rd European Workshop on Software Architecture. LNCS 4344, Nantes: Springer-Verlag, 2006. 2−16. [doi: 10.1007/11966104_2]

[16] Yu HQ, Liu DM, He XD, Yang L, Gao S. Secure software architectures design by aspect orientation. In: Proc. of the 10th IEEE Int'l Conf. on Engineering of Complex Computer Systems. Washington: IEEE Computer Society, 2005. 47−55. [doi: 10.1109/ICECCS.2005.75]

[17] Zhang J, Goldsby HJ, Cheng BHC. Modular verification of dynamically adaptive systems. In: Proc. of the 8th ACM Int'l Conf. on Aspect-Oriented Software Development. New York: ACM Press, 2009. 161−172. [doi: 10.1145/1509239.1509262]

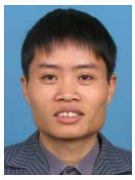
[18] Thacker RA, Jones KR, Myers CJ, Zheng H. Automatic abstraction for verification of cyber-physical systems. In: Proc. of the Int'l Conf. on Cyber-Physical Systems. New York: ACM Press, 2010. 12−21. [doi: 10.1145/1795194.1795197]

[19] Bujorianu MC, Barringer H. An integrated specification logic for cyber-physical systems. In: Proc. of the 14th IEEE Int'l Conf. on Engineering of Complex Computer Systems. Washington: IEEE Computer Society, 2009. 291−300. [doi: 10.1109/ICECCS.2009.36]

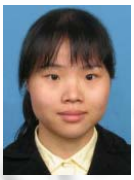
附中文参考文献:

[6] 马建刚,黄涛,汪锦岭,徐罡,叶丹.面向大规模分布式计算发布订阅系统核心技术.软件学报,2006,17(1):134−147. http://www.jos.org.cn/1000-9825/17/134.htm [doi: 10.1360/jos170134]

[11] 张琳琳,应时,倪友,赵楷,文静.一种软件体系结构关注点分析方法.计算机学报,2009,32(9):1782−1791.



范贵生(1980－),男,福建莆田人,博士,助理研究员,CCF 会员,主要研究领域为软件工程,面向服务计算,形式化方法.



陈丽琼(1982－),女,博士,讲师,主要研究领域为分布式计算,嵌入式系统,形式化方法.

虞慧群(1967－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程,可信计算与安全,形式化方法.

刘冬梅(1970－),女,博士,高级工程师,主要研究领域为软件工程,面向服务计算.