软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(6):1252−1266 [doi: 10.3724/SP.J.1001.2011.04014]

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

AADL 模型可靠性分析评估工具*

董云卫, 王广仁+, 张 凡, 高 磊

(西北工业大学 计算机学院,陕西 西安 710072)

Reliability Analysis and Assessment Tool for AADL Model

DONG Yun-Wei, WANG Guang-Ren+, ZHANG Fan, GAO Lei

(College of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

+ Corresponding author: E-mail: wanggry@gmail.com

Dong YW, Wang GR, Zhang F, Gao L. Reliability analysis and assessment tool for AADL model. *Journal of Software*, 2011,22(6):1252−1266. http://www.jos.org.cn/1000-9825/4014.htm

Abstract: This paper focuses on a reliability model of embedded system architecture using AADL (architecture analysis and design language). It performs transformation from AADL reliability model to GSPN (general stochastic Petri net) model and assesses AADL architecture reliability computation model by the means of GSPN theory. To support the reliability analysis and assessment automatically, this paper designs and implements an AADL reliability assessment model tool, ARAM (AADL reliability assessment model tool), with formal methods. It is integrated into OSATE (the open source AADL tool environment) and uses PIPE2 (platform independent Petri net editor 2) to carry out the reliability assessment of GSPN model. Meanwhile, this paper also presents a case study on the reliability analysis and assessment of avionics flight control system to demonstrate the performance of ARAM.
Key words: AADL; reliability model; GSPN; reliability analysis and assessment; formal method

摘 要: 主要针对 AADL(architecture analysis and design language)嵌入式系统体系结构进行可靠性建模,实现 AADL 可靠性模型到广义随机 Petri 网(general stochastic Petri net,简称 GSPN)可靠性计算模型的转换,并基于 GSPN 可靠性计算模型对嵌入式系统进行可靠性评估.为了支持可靠性分析评估过程的自动化,根据模型转换的形式化方法,设计并实现了 AADL 可靠性评估工具(AADL reliability assessment model tool,简称 ARAM),该工具集成在 AADL 体系结构设计工具 OSATE(the open source AADL tool environment)中,并内置 Petri 网计算工具 PIPE2(platform independent Petri net editor 2),实现基于 GSPN 模型的可靠性分析评估.同时,结合航空飞行控制系统的可靠性分析评估介绍了 ARAM 工具的应用情况.
关键词: AADL;可靠性模型;GSPN;可靠性分析与评估;形式化方法
中图法分类号: TP302 **文献标识码**: A

随着嵌入式系统的发展,嵌入式系统结构越来越复杂,规模越来越大,对系统的开发成本、开发周期及非功能属性(可调度性、可靠性、安全性)的要求也越来越高,早期的嵌入式系统开发方法已经不能满足当前需求.

* 基金项目: 国家自然科学基金(60736017); 国家高技术研究发展计划(863)(2009AA01Z147); 西北工业大学基础研究基金(JC200917)

收稿时间: 2010-07-10; 修改时间: 2011-03-29; 定稿时间: 2011-04-06

为此,业界引入了模型驱动结构方法(model driven architecture,简称 MDA)[1],系统开发被提升到更高的级别——模型级,针对特定计算平台的编码工作由机器自动完成,模型成为开发过程中的核心,从而可以在模型级别就对系统的非功能属性进行分析.如果系统的非功能属性不能满足需求,可以从模型级别对系统架构进行修改,这样就可以极大地缩短系统的开发周期,节约开发成本.针对这样的应用需求和技术发展趋势,美国自动化工程师协会(Society of Automotive Engineers,简称 SAE)发布了航空标准 AS5506——架构分析与设计语言 AADL (architecture analysis and design language)[2,3].AADL 描述复杂的实时嵌入式系统架构,它不关心构件的具体实现,通过构件以及构件间的交互、软件构件与硬件构件的绑定,对实时高可靠嵌入式系统进行描述与分析.

形式化方法在模型设计和模型属性分析方面具有较好的描述和分析能力,它采用数学逻辑方法,能够帮助系统设计和分析人员发现非形式化方法不容易发现的系统描述的不一致、不明确以及不完整性,是提高软件系统的安全性与可靠性的重要手段.AADL 是一种半形式化的建模语言,具有确定语义定义并且有严格语法的语言表达的规范.由于 AADL 只是一种建模语言,只是对构件进行了相应的属性描述,例如线程执行时间描述、构件的安全等级描述、构件错误模型附录库等等,如果想对 AADL 系统架构模型进行相应的非功能属性进行分析,还需要对 AADL 系统架构模型中的构件属性进行形式化描述,然后再借用形式化理论方法和相关工具来验证系统模型的非功能属性是否满足要求.目前,利用 AADL 错误模型(error model)虽然可以对 AADL 系统架构可靠性建模,但是它只能表示为一个静态的系统模型.为了对 AADL 系统架构模型进行可靠性分析和验证,需要采用形式化的方法对 AADL 系统架构可靠性模型中系统构件间的交互行为的动态性进行描述.本文采用 GSPN 对 AADL 模型中构件及其错误状态可能的失效进行刻画,建立 AADL 可靠性计算模型,这样就可以基于 GSPN 可靠性计算模型来验证 AADL 可靠性模型是否满足设计要求.

基于 AADL 系统架构的可靠性分析评估方法工作量大、任务繁琐,需要有较好的工具支持才能实现对系统架构模型的可靠性分析评估,有利于在系统设计的早期对系统的可靠性需求进行分析和验证,提高系统的质量.本文提出的系统可靠性设计和实现方法方便系统架构可靠性分析自动化,方便用户在系统设计的早期对嵌入式系统的可靠性进行分析评估.因此,本文设计与实现了 AADL 可靠性模型分析评估工具——ARAM(AADL reliability assessment model tool).ARAM 工具实现了从 AADL 可靠性模型到 GSPN 模型的转换,然后基于 GSPN 可靠性模型进行分析评估.用户使用 OSATE 建立 AADL 可靠性模型,利用 ARAM 工具对建立的 AADL 可靠性模型进行分析评估,这样就实现了从建立模型到分析评估评估模型的一体化.

本文第 1 节介绍 AADL 可靠性模型分析的相关研究.第 2 节介绍 AADL 可靠性模型向 GSPN 模型转化的形式化方法.第 3 节介绍 ARAM 工具总体框架设计.第 4 节介绍 ARAM 工具的实现.第 5 节结合航空飞行控制系统实例,介绍 ARAM 工具的应用.第 6 节是本文总结.

1 AADL 可靠性分析相关研究

当前,系统可靠性评估方法主要分为两种:一种是基于代码的系统可靠性评估方法,另一种是基于构件的系统可靠性评估方法.基于代码的系统可靠性评估方法实现简单,但存在人为因素的影响,评估准确度不高.对具有高可靠性要求且结构越来越复杂的嵌入式系统而言,这种方法只能对软件代码实现初步的可靠性评估,在系统统设计阶段不宜使用.基于构件的系统可靠性评估方法相比于基于代码的系统可靠性评估方法而言,更适合基于模型的系统开发方法.现在有很多组织基于 AADL 模型进行设计与分析研究,并开发了一些相关的设计分析工具.其中,卡耐基梅隆大学软件工程研究所(software engineering institute,简称 SEI)基于 Eclipse 平台开发了开源 AADL 设计工具 OSATE[4].OSATE 提供从前到后对 AADL 模型进行处理的工具集,可以以 XML 文本文件或图形的方式表示 AADL 模型文件,并对 AADL 模型文件进行语法和语义的检查.OSATE 可以将一个系统构件进行实例化,通过实例化,不需要完整的声明模型就可以对 AADL 模型进行处理,而且 OSATE 是开源的,为其他的基于 AADL 模型的分析提供了环境.许多研究者以 OSATE 环境为基础研究开发针对 AADL 模型属性分析的工具,如 AADL 模型仿真测试工具 AADS[5],它实现了从 AADL 到 SystemC 的转换,基于 SystemC 代码对 AADL 模型进行仿真和性能分析.Isograph 软件[6]是由英国 Isograph 公司开发的可靠性、可用性、安全性、维修性和

保障性工程设计分析软件,采用故障树分析法、马尔可夫分析和可靠性框图分析等方法对系统可靠性进行分析.因 AADL 更适用于复杂嵌入式实时系统体系结构建模,ARAM 工具在可靠性分析与评估中可以根据模型驱动结构方法建模特点对嵌入式系统模型可靠性进行分析.

AADL 可靠性模型由 AADL 架构模型和 AADL 错误模型两部分构成:AADL 架构模型从软件和硬件两方面来描述一个嵌入式系统,描述了系统的各个构件、构件之间的连接关系以及软件与硬件的绑定关系;AADL 错误模型描述了构件可靠性信息,通过错误模型子附录和 AADL 架构模型联系[7−9].AADL 架构模型和错误模型描述都是通过类型(type)和实现(implementation)来实现的.在架构模型中,类型定义了构件的功能接口,实现定义了构件的内部结构;而在错误模型中,类型定义了错误状态、错误事件和错误传播,实现定义了由错误事件和错误传播引起的错误状态之间的变迁.AADL 架构模型和错误模型通过错误模型子附录联系起来,错误模型子附录需要在架构模型中声明.错误是通过构件之间的连接或者软硬件之间的绑定进行传播的.另外,在错误模型子附录中定义了错误的过滤与屏蔽规则(Guard_In,Guard_Out)和连接错误状态到模式机制(Guard_Event, Guard_Transition).

然而,AADL 可靠性模型是一种静态的模型,现有分析可靠性模型的方法主要包含故障树、GSPN、Markov 链等,不能直接用来分析 AADL 模型的可靠性,并且现有的 AADL 设计工具也没有对 AADL 可靠性模型进行分析与评估的功能.虽然文献[10−12]中提出了 AADL 可靠性模型向 GSPN 可靠性计算模型的转换理论,通过这些理论分析可以实现基本错误附录库、基本依赖关系元素(Out-In Propagation)以及高级依赖关系元素(Guard_In, Guard_Out,Guard_Event,Guard_Transition)向 GSPN 可靠性计算模型的转换,但是这些研究在实现 AADL 元素向 GSPN 模型转换时主要采用了实例的方式对这些元素进行描述,其中,高级依赖关系元素中采用实例的方法缺点明显,没有实现高级依赖元素和 GSPN 模型元素之间的一一对应.高级规则转换对应为 GSPN 模型元素中的弧,本文在这里不作详细介绍.而且在实现依赖关系元素向 GSPN 模型元素转换时,没有把依赖关系元素和 GSPN 模型元素实现清晰的对应,给设计基于模型的 AADL 可靠性模型分析评估自动化工具带来了许多挑战.因此,本论文在深入研究这些转换理论的基础上,对原有转换规则进行分解,实现了 AADL 模型元素到 GSPN 可靠性模型元素之间的一一对应关系,并通过形式化的描述方法对转换规则进行定义,为下一步高级依赖元素转换规则打下基础,同时方便了 ARAM 工具的设计与程序实现.

2 AADL 可靠性模型向 GPSN 模型转换形式化方法

本文采用 GSPN 模型作为系统架构模型的可靠性计算模型,并建立 AADL 可靠性模型与 GSPN 可靠性模型模型之间的对应关系,如图 1 所示.根据系统设计可靠性需求,设计了 AADL 可靠性模型分析框架,实现了 AADL 可靠性模型向 GSPN 模型的自动转换.这样就可以实现 GSPN 可靠性计算模型评估,根据评估的结果判断原 AADL 系统可靠性模型是否满足可靠性需求,如果不满足,需要重新设计 AADL 模型框架.

GSPN 是随机 Petri 网的一种扩充,能够准确描述系统的状态及其状态之间的变迁,可以对系统的可达性和可靠性等进行分析,因此在复杂系统的建模与性能分析中取得了显著的成功[13].

定义 1. GSPN 模型是一个六元组 $GSPN=(S,T,F,W,M_0,\lambda)$,其中:

a) S 是位置集合,$S=\{p_1,p_2,p_3,\ldots,p_n\}$;

b) T 是变迁集合,$T=T_t\cup T_i, T_t\cap T_i=\varnothing$,时间变迁集 $T_t=\{t_1,t_2,\ldots,t_k\}$,瞬时变迁集 $T_i=\{t_{k+1},t_{k+2},\ldots,t_n\}$,与时间变迁集相关联的平均变迁实施速率集合$\lambda=\{\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_k\}$;

c) F 是弧的集合,$F=A_{st}\cup A_{ts}\cup A_i, A_{st}\subseteq S\times T$,描述从位置到变迁的弧集合;$A_{ts}=T\times S$,描述从变迁到位置的弧集合;$A_i\subseteq S\times T$,描述从位置到变迁的禁止弧集合;

d) W 描述瞬时变迁上的权重,$W:T_i\rightarrow N+$,其中,$N+$是非零自然数;

e) $M_0:S\rightarrow N$ 是初始标志,N 表示自然数;

f) λ表示时间变迁实施速率集.

基于 GSPN 描述的嵌入式系统可靠性计算模型是对系统 AADL 模型中各种交互行为的动态性建模,利用

GSPN 描述 AADL 模型的可靠性,关键是建立 AADL 可靠性模型向 GSPN 可靠性计算模型的转化关系.这种模型转换分为 3 阶段完成:1) 对错误模型进行形式化定义,使用自动机模型对其进行描述,实现错误模型基本元素到 GSPN 模型的转换;2) 在基本模型转换的基础上,实现 Out-Propagation 和 In-Propagation 向 GSPN 模型元素进行转换,并对 Out-Propagation 和 In-Propagation 进行匹配;3) 给所有的向外错误传播 Out Propagation 添加对应的 GSPN 模型标志的吸收,最终实现 AADL 构件之间依赖关系向 GSPN 模型的转换.下面对这 3 个阶段的工作分别阐述.

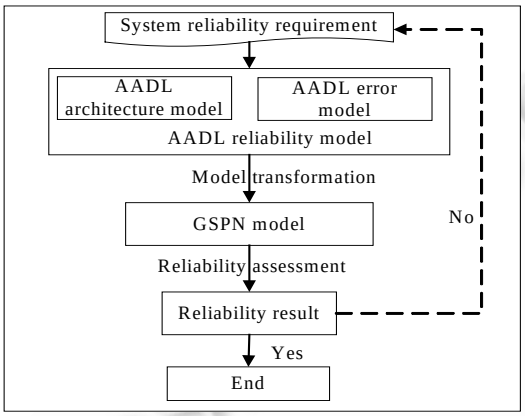


Fig.1 Analysis framework of AADL reliability model

图 1 AADL 可靠性模型分析框架

2.1 错误模型基本元素转换

AADL 错误模型基本元素和错误状态之间的变迁可以用一个有限自动机模型描述,这个自动机模型和 GSPN 模型之间存在一种映射关系.

定义 2. 用有限自动机描述的 AADL 错误模型表示为 $EM=(ES,EE,TR,IES,F)$,其中:

a) ES 是所有错误状态的集合,$ES=\{es_1,es_2,es_3,\ldots,es_m\}$;

b) EE 是所有错误事件的集合,$EE=EE_p\cup EE_f$,服从泊松分布的错误事件变迁集 $EE_p=\{ee_1,ee_2,ee_3,\ldots,ee_k\}$,服从固定概率分布的错误事件集 $EE_p=\{ee_{k+1},ee_{k+2},ee_{k+3},\ldots,ee_n\}$;

c) TR 是所有错误状态间变迁的集合,转移函数 $TR(es_i,ee_j)=es_k$;

d) IES 是初始错误状态集,因 AADL 每一个错误模型类型只有一个初始错误状态,所以这个集合中只含有一个元素,$IES=\{es|es$ 是初始错误状态$\}$;

e) F 是所有接受状态的集合,所有的错误状态都是可以接受的.

根据 GSPN 的定义 1,实现 AADL 错误模型基本元素到 GSPN 模型的对应函数 $GSPN\_EM$ 可表示为:

定义 3. AADL 基本错误模型元素向 GSPN 模型元素转换:

a) $GSPN\_EM(es)=p,GSPN\_EM(ES)=S$,实现 AADL 错误状态到 GSPN 中位置之间的转换;

b) $GSPN\_EM(ee)=t,GSPN\_EM(EE_p)=T_t,GSPN\_EM(EE_f)=T_i$,即 AADL 错误模型中服从柏松分布的错误事件转换到 GSPN 中的时间变迁,AADL 错误模型中服从固定概率分布的错误事件转换到 GSPN 中的瞬时变迁;

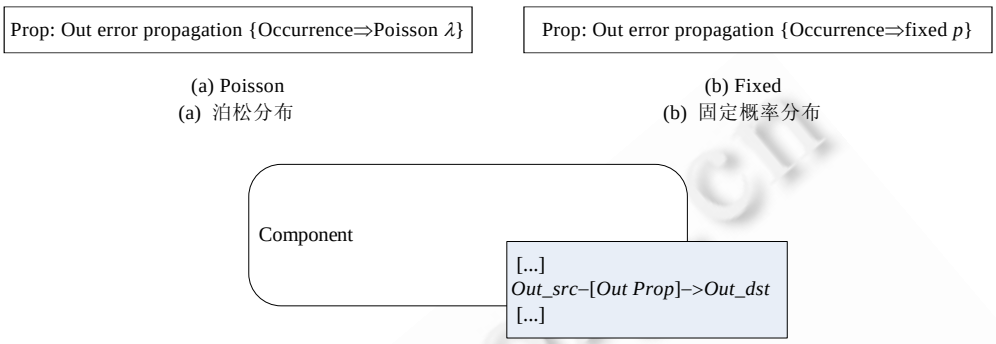
c) $GSPN\_EM(TR)=A_{st}\cup A_{ts}$,将 AADL 中错误状态之间的变迁转换到对应的 GSPN 中位置到变迁的弧和变迁到位置的弧;

d) $GSPN\_EM(es)=p,p$ 是包含一个标志的位置,将初始错误状态 es 元素转换到只包含一个标志的位置.

2.2 错误模型元素基本依赖规则Out Propagation

通过上述转换,实现了 AADL 错误模型基本元素到 GSPN 模型元素的转换,接着需要实现 AADL 基本错误模型依赖元素向 GSPN 模型元素的转换.

Out Propagation 定义了错误源构件通过向外错误传播影响接受构件错误发生过程,Out Propagation 的对外传播有两种:一种是服从泊松分布函数的,另一种是固定概率的,分别对应时间变迁和瞬时变迁,如图 2 所示.



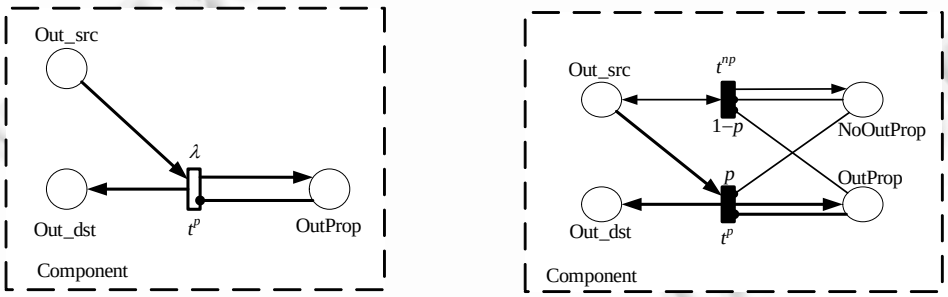
(a) Poisson
(a) 泊松分布

(b) Fixed
(b) 固定概率分布

(c) Component with Out Propagation
(c) 向外传播错误的构件

Fig.2 Example of Out-Propagation
图 2 Out-Propagation 例子

图 3 为针对图 2 的 GPSN 模型转换结果,在图 3(a)中,向外错误传播服从变量为λ的泊松分布;在图 3(b)中,向外错误传播服从固定概率 p.显然,不向外错误传播的固定概率为 $1-p$.



(a) GSPN model of Poisson
(a) Poisson 分布 GSPN 转换

(b) GSPN model of fixed
(b) 固定概率分布 GSPN 转换

Fig.3 GSPN model of Out-Propagation
图 3 Out-Propagation 的 GSPN 模型

从图 3 可以看出,Out Propagation 向 GSPN 模型的转换结果是一个 GSPN 基本元素位置、变迁和弧的集合,利用形式化方法对其进行定义:

定义 4. Out Propagation 向 GSPN 模型元素转换的形式化规则:

a) $GSPN\_EM(Out\text{-}Propagation)=Out\_S^{pt}\cup Out\_T^{tp}\cup t^p\_A_{ts}\cup t^{po}\_A_{st}$,当 Out Propagation 服从泊松分布;

b) $GSPN\_EM(Out\text{-}Propagation)=Out\_S^{pi}\cup Out\_T^{ip}\cup t^p\_A_{ts}\cup t^{np}\_A_{st}\cup t^{pio}\_A_{st}\cup t^{npio}\_A_{st}$,当 Out Propagation 服从固定概率分布;

其中:

- $Out\_S^{pt}=\{OutProp\}$;

- Out_S^{pi}={*OutProp*,*NoOutProp*};
- Out_T^{tp}={t^p};
- Out_T^{ip}={t^p,t^{np}};
- $t^p\_A_{ts}$=Out_T^{tp}×Out_S^{pt};
- $t^{np}\_A_{ts}$={t^{np}}×{*NoOutProp*}.

禁止弧:$t^{po}\_A_{st}$=Out_S^{pt}×Out_T^{tp},$t^{pio}\_A_{st}$=Out_S^{pi}×Out_T^{tp},$t^{npio}\_A_{st}$=Out_S^{pi}×{t^{np}}.

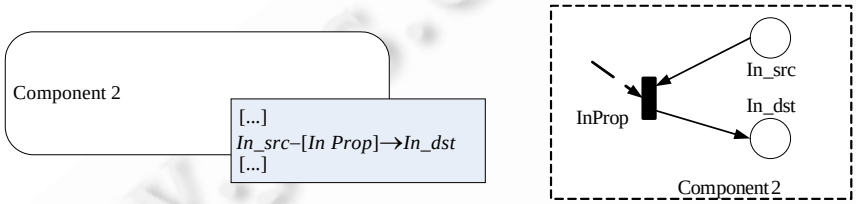
定义 5. 错误状态变迁规则 *Out_src*–[*OutProp*]→*Out_dst* 描述错误源构件通过向外传播一个错误 Prop,从错误状态 Out_src 转移到 Out_dst,完成 Out Propagation 的转换后, *Out_src*–[*Out Prop*]→*Out_dst* 向 GSPN 模型的转换规则 *GSPN_EM*(*Out_src*–[*Out Prop*]→*Out_dst*)=A_{st}∪A_{ts}:

a) A_{st}={*Out_src*}×{t^p},A_{ts}={t^p}×{*Out_dst*},当 Out Propagation 服从泊松分布;

b) A_{st}={*Out_src*}×{t^p}∪{*Out_src*}×{t^{np}},A_{ts}={t^p}×{*Out_dst*}∪{t^{np}}×{*Out_src*},当 Out Propagation 服从固定概率分布.

2.3 错误模型元素基本依赖规则In Propagation

In Propagation 定义了向内错误传播是其他构件的错误影响当前构件而使当前产生错误的过程.图 4 是向内错误传播的一个示例.



(a) Component with In-Propagation
(a) 接受错误传播构件

(b) GSPN model of In-Propagation
(b) In-Propagation 的 GSPN 模型

Fig.4 Example of In-Propagation
图 4 In-Propagation 模型例子

当从外部传播进来错误时,将激发当前构件从初始错误状态转移到目的错误状态.向内错误传播是与向外错误传播相对应的,故向内错误传播不存在发生率问题,如图 4(b)所示.定义 InProp 为瞬时迁移,其发生概率为 1.

定义 6. In Propagation 向 GSPN 模型元素转换的形式化规则:*GSPN_EM*(*In-Propagation*)={*InProp*}.

错误状态变迁 *In_src*–[*In Prop*]→*In_dst* 描述接受构件接受到一个错误传播 Prop,从错误状态 In_src 转移到错误状态 In_dst,*In_src*–[*In Prop*]→*In_dst* 向 GSPN 模型的转换 *GSPN_EM*(*In_src*–[*In Prop*]→*In_dst*)=A_{st}∪A_{ts}.其中,A_{st}={*In_src*}×{*InProp*},A_{ts}={*InProp*}×{*In_dst*}.

2.4 Out-In Propagation匹配

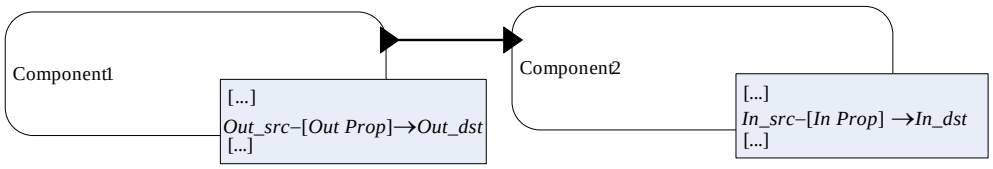
前面定义了 AADL 中 Out Propagation 和 In Propagation 的 GSPN 转换规则,但 In Propagation 是依赖于 Out Propagation 存在的,单独进行 In Propagation 或者 Out Propagation 的转换是毫无意义的.

图 5 是对 Out Propagation 和 In Propagation 的 GSPN 模型转换的图形化描述,如图 5(b)和图 5(c)所示,左半部分是一个向外传播错误 Out Propagation 构件,右半部分是一个接受错误 In Propagation 构件,中间是一个连接 OutProp 和 InProp 的双向弧,这条双向弧是 Out-In Propagation 匹配的结果.

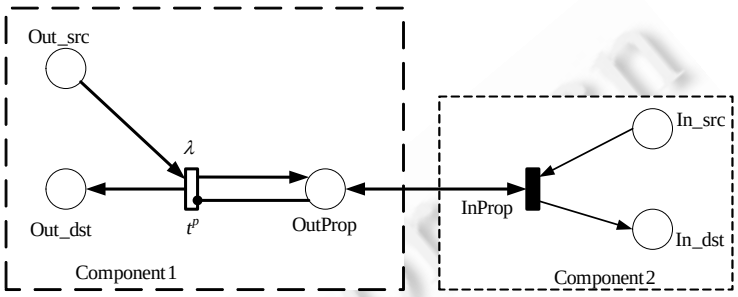
定义 7. Out-In Propagation 向 GSPN 模型元素转换的形式化规则:

$$GSPN\_EM(Out\text{-}In\text{-}Propagation)=Out\_In\_A_{st}\cup Out\_In\_A_{ts},$$

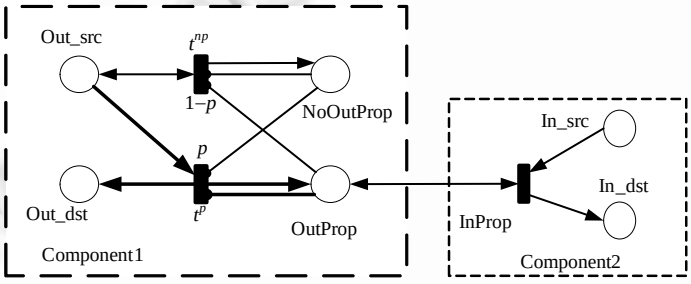
其中,$Out\_In\_A_{st}$={*OutProp*}×{*InProp*},$Out\_In\_A_{ts}$={*InProp*}×{*OutProp*}.



(a) Components with Out-In Propagation
(a) Out-In Propagation 匹配构件



(b) GSPN model of Out-In Propagation with Poisson
(b) 泊松分布 Out-In Propagation 匹配的 GSPN 模型



(c) GSPN model of Out-In Propagation with fixed
(c) 固定概率分布 Out-In Propagation 匹配的 GSPN 模型

Fig.5 Example of Out-In Propagation
图 5 Out-In Propagation 匹配模型例子

2.5 错误传播位置清空规则

发送构件通过 Out Propagation 发出一个错误,如果这个错误不能被接受构件接受,那么这个错误标志应该消失,不应该继续存留在 Out Propagation 的位置中,位置 OutProp 中的标志表示为构件的发送错误,因此这个标志应该被清除.

如果下面两个条件中有一个发生,位置 OutProp 中的标志不应该被吸收,否则标志应该被吸收:

a) 促使位置 OutProp 中产生标志的所有错误源都一直处于错误发送状态,例如图 5(b)中的发送错误构件一直处于 Out_src 状态,那么错误表示不应该被吸收;

b) 如果接受构件处于可以接受错误的状态,那么错误应该被接受,也不应该被吸收.例如,在图 5(b)中,接受构件处于 In_src 状态,错误标志应该被接受构件接受,不应该被吸收.

通过上面这两条规则,可以得到位置 OutProp 中的标志的清空条件是因素 *a* 和因素 *b* 都不发生,因此,其布尔条件表达式是 $[\wedge_{i=1}^{n}(\neg t_i^{cause})]\wedge[\wedge_{j=1}^{m}(\neg t_j^{effect})]$.其中,$t_i^{cause}$ 表示促使位置 OutProp 中产生标志的错误源,t_j^{effect} 表示接受 OutProp 中错误标志的接受源.

如图 5(b)所示,$t_1^{cause} = Out\_src, t_1^{effect} = In\_src$,OutProp 中标志的清空布尔表达式是

$$\neg t_1^{cause} \wedge \neg t_1^{effect} = \neg Out\_src \wedge \neg In\_src.$$

因此,清空位置 OutProp 中的标志的形式化规则:$GSPN\_EM(Emptying\text{-}OutProp)=Ab\_T^e \cup T^{eo}\_A_{st} \cup T^e\_A_{st}$.

其中,$Ab\_T^e=\{t^e\}, T^e\_A_{st}=\{OutProp\}\times Ab\_T^e$.

禁止弧:$T^{eo}\_A_{st}=\{Out\_src\}\times Ab\_T^e \cup \{In\_src\}\times Ab\_T^e$.

利用上面对吸收公式的分析,对图 5(b)的位置 OutProp 添加吸收后,形成图 6 所示的 GSPN 模型.

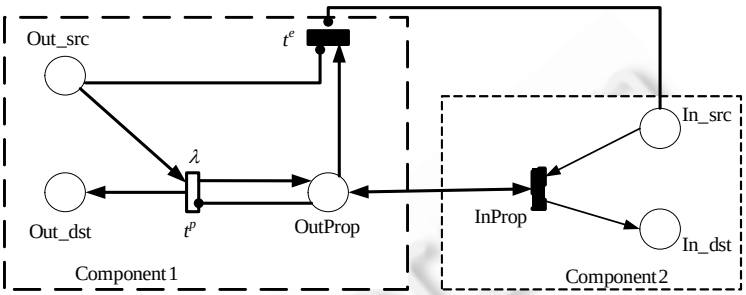


Fig.6 GSPN model of emptying the OutProp place

图 6 清除位置 OutProp 中标志的 GSPN 模型

3 ARAM 工具总体架构设计

ARAM 工具主要功能是实现 AADL 可靠性模型向 GSPN 模型的自动转换,然后调用开源工具 PIPE2[14]对转换生成的 GSPN 可靠性模型进行评估与分析.AADL 可靠性模型主要包括 AADL 架构模型和 AADL 错误模型,这两个模型都可以使用开源工具 OSATE 进行建立,OSATE 可以对 AADL 可靠性模型进行基本的语法和语义检查.

如图 7 所示,OSATE 和 PIPE2 是基于 Eclipse 平台开发的.OSATE 为 AADL 可靠性模型的建立提供设计环境,PIPE2 为 GSPN 模型的分析与评估提供了环境.ARAM 工具是基于 OSATE 和 PIPE2 开发的,实现了 AADL 可靠性模型的自动转换,最终调用 PIPE2 对 GSPN 模型进行评估.应用层中,ARAM 工具的输入是 AAXL 文件,AAXL 是 OSATE 提供给外界开发工具的文件接口,是 XML 格式的文件;ARAM 工具的输出是可靠性评估与分析结果.如中间层所示,ARAM 工具可以分成 3 个主要功能:

a) AADL 可靠性模型设计功能.在 OSATE 和错误模型附录库插件开发环境基础上设计 AADL 架构模型和 AADL 错误模型,利用 OSATE 对 AADL 架构模型和 AADL 错误模型进行解析,生成 AAXL 模型;然后,在 AAXL 模型中选择需要评估的系统构件,将其实例化.OSATE 将系统实例化,以 AAXL 文件的形式进行保存.通过以上分析,得到 ARAM 工具模型转换功能模块所需要的 AADL 架构模型、AADL 错误模型以及 AADL 实例模型的 AAXL 文件;

b) AADL 可靠性模型向 GSPN 模型转换功能.利用 XML 解析器 SAX 对得到的 AAXL 模型文件进行解析、利用 OSATE 中 Ecore 模型生成的 AADL 类元素代码.其中,AADL Ecore 模型结构非常复杂,将 AAXL 文件中元素的类型解析到对应的 AADL 构件类、AADL 错误模型类和 AADL 实例模型类,通过 AADL 可靠性模型向 GSPN 模型转换规则,调用 PIPE2 中对应的 GSPN 模型元素 Place 类、Transition 类等.其中,PIPE2 代码量大,最终将 AADL 可靠性模型类转换到对应的 GSPN 模型类;

c) GSPN 可靠性评估功能.调用开源工具 PIPE2 内的评估函数对生成的 GSPN 模型计算出可靠性结果,调整错误事件或者错误传播中的泊松分布的参数值,得出可靠性评估曲线,从而进一步分析系统的可靠性变化趋势,根据系统的可靠性需求,选择对应的各个子构件的性能参数.结合 ARAM 工具评估计算的特点,针对原有 PIPE2 的状态空间搜索时进行计算的特点,实现一次状态空间搜索多次计算,

改变了计算时间随评估次数线性增长的缺点,极大地提高了可靠性评估效率.

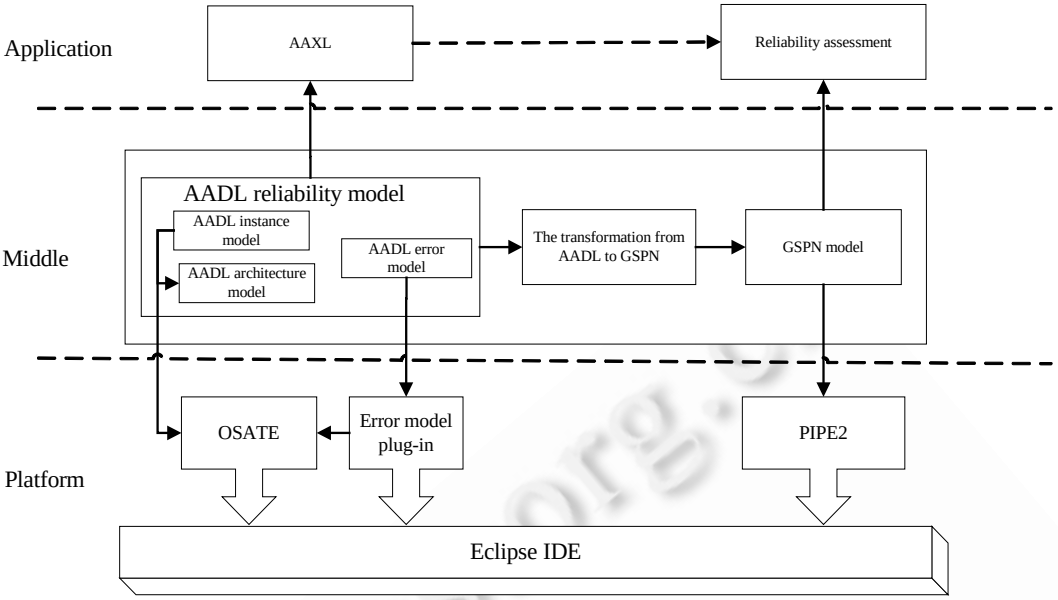


Fig.7 Framework of ARAM

图 7 ARAM 工具总体架构

4 ARAM 工具实现

ARAM 工具是基于 OSATE 和 PIPE2 工具开发的,与 OSATE 和 PIPE2 之间结合得比较紧密,对 OSATE 基本元素类(ErrorState 类、ErrorEvent 类、ErrorPropagation 类等)和 PIPE2 中的基本元素类(Place 类、Transition 类、Arc 类等)使用较多.ARAM 工具中有一些功能类(AadlErrorAnnextoGSPN 类)继承和实现了 OSATE 中的抽象类,通过对 OSATE 类和 PIPE2 类的大量继承,提高了 ARAM 工具代码的可靠性,最终实现从 AADL 可靠性模型到 GSPN 模型的转换.图 8 为 ARAM 工具类图.

(a) 类 AramAbstractAaxlAction 继承和实现了 Eclipse 插件平台中的 IWorkbenchWindowActionDelegate, IObjectActionDelegate 类,这个类的主要功能是提取 AADL 可靠性工程中的 AADL 架构模型和 AADL 错误模型中的 AAXL 文件,然后将 AAXL 文件进行解析,将 AAXL 中的元素转换到对应的 AADL 基本元素类(ErrorState 类、ErrorEvent 类、ErrorPropagation 类、Memory 类、Process 类等);

(b) 类 AramAction 继承了 AramAbstactAaxlAction 类,实现了其中的关键方法 doAaxlAction(),起到承前启后的作用,调用后面 AadltoGSPN,是实现模型转换的一个关键步骤;

(c) 类 AadlErrorAnnextoGSPN 继承和实现了 AADL 错误模型插件中的类 edu.cmu.sei.aadl.errorannex.-util.ErrorannexSwitch,从图 8 可以看出,这个类比较重要,主要功能是实现错误模型附录库中的错误模型类型和错误模型实现中的基本元素到 GSPN 模型基本元素的转换;

(d) 类 AadltoGSPN 继承和实现了 OSATE 插件中的 edu.cmu.sei.aadl.model.util.AadlProcessingSwitch 类,这个类的主要功能是:根据每个构件自带的错误模型,利用 AADL 架构模型构件之间的依赖关系,把每个构件的错误模型和其他构件的错误模型联系起来.通过这个类的 caseAnnexLibrary 方法,调用 AadlErrorAnnextoGSPN 中的错误模型到 GSPN 模型的转换,实现错误模型附录库到 GSPN 模型的转换;通过这个类的 caseAnnexSubclause 方法,实现构件和错误模型实现之间的关联;通过 caseAnnexLibrary 和 caseAnnexSubclause 这两个方法,把 AadltoGSPN 类和 AadlErrorAnnextoGSPN 类联系起来.

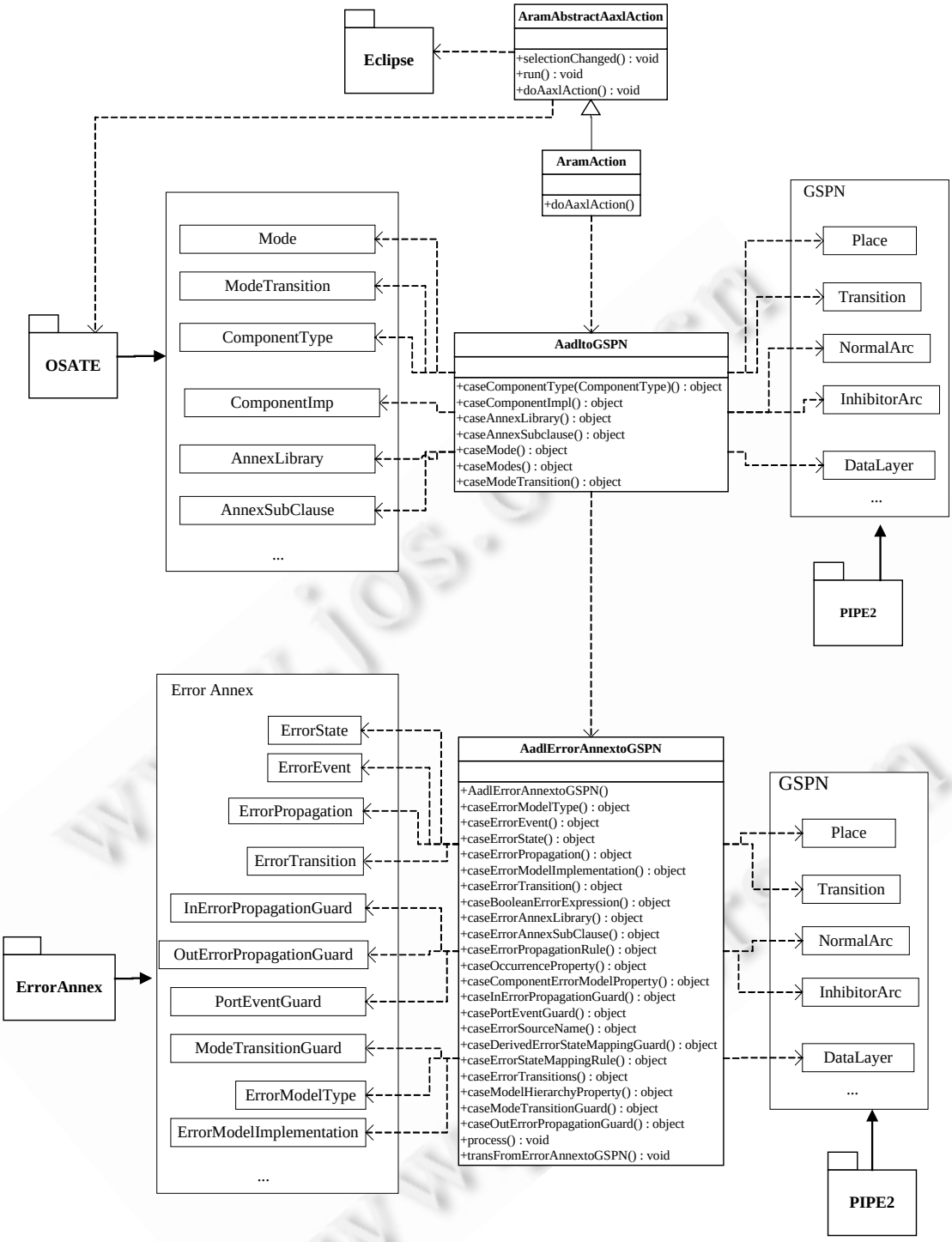


Fig.8 Class of ARAM

图 8 ARAM 工具类

5 ARAM 工具应用

为介绍 ARAM 工具的使用,本节将对飞行控制系统进行 AADL 可靠性建模,接着把飞行控制系统的 AADL 可靠性模型转换到对应的 GSPN 模型,最后对飞行控制系统可靠性进行分析与评估.飞行控制系统主要是为航天器完成飞行控制和计算,本系统主要关注其自动驾驶和导航系统,因自动驾驶系统可以完成飞行控制系统的自动驾驶,驾驶员只需要输入目的地信息而不需要给出完整的飞行路线.

飞行控制系统的 AADL 架构模型如图 9 所示,主要包括两 Nav_Autopilot_System 子系统和 HCI_System 子系统,这两部分通过 LAN 总线进行交互,互相传递信息:

a) Nav_Autopilot_System 子系统,主要功能是使 4 个作动器和 GPS 传感器交互,通过 GPS 获取当前地理位置信息,经过 P_NAV_Con 进程计算后,通过调整作动器参数值来控制飞行控制系统状态;

b) HCI_System 子系统,主要功能是显示当前地理位置,帮助驾驶员了解当前情况,那么驾驶员就可以判断当时的飞行情况是否满足要求;同时,驾驶员也可以输入自动驾驶参数和取消自动驾驶.

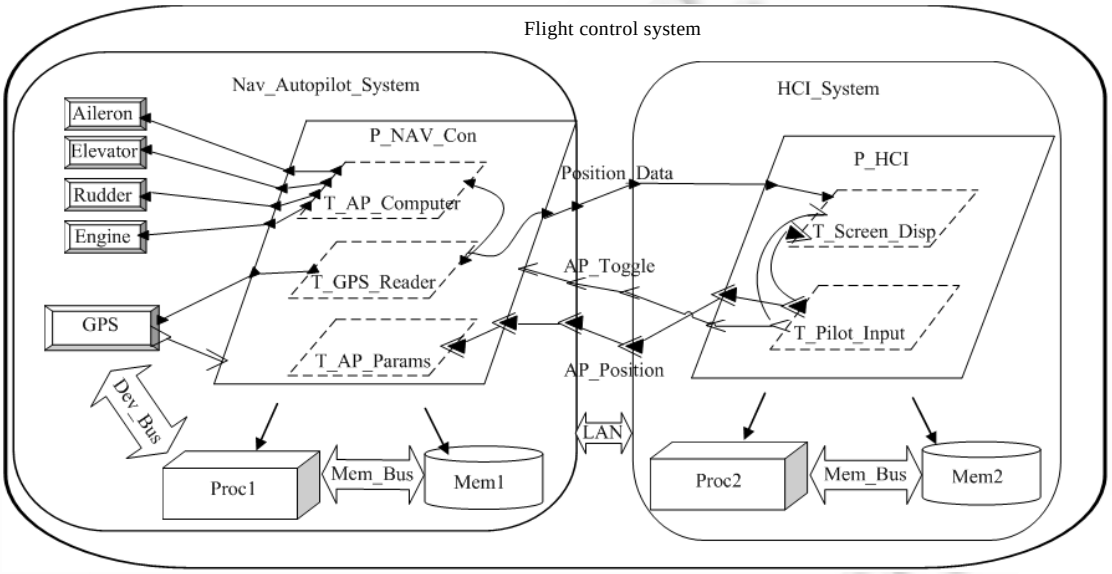


Fig.9 AADL architecture of flight control system

图 9 飞行控制系统 AADL 架构

根据飞行控制系统的 AADL 架构模型以及每个构件实际的动态运行情况,给每个构件增加相应的错误模型.图 10 中描述了处理器构件 Proc1 和线程构件 T_GPS_Reader 的错误模型,错误模型中包含了错误模型类型和错误模型实现:

- 错误模型类型中定义了错误状态、错误事件和错误传播;
- 错误模型实现中定义了错误状态之间因错误事件、向外错误传播和向内接受错误而发生迁移.

其他构件的错误模型在此省略,其中,飞行控制系统的 AADL 架构模型文本描述也在此省略.结合图 9 飞行控制系统的 AADL 架构模型,组成 AADL 可靠性模型.然后,对飞行控制系统实例化,得到其实例化模型.从飞行控制系统的实例化模型中可以得到构件之间的语义连接,如图 9 所示,进程 P_NAV_Con 内的线程 T_GPS_Reader 和进程 P_HCI 内的线程 T_Screen_Disp 是直接连接的,如果 T_GPS_Reader 发生错误,那么会把错误直接传播到线程 T_Screen_Disp.

Proc1

```
error model procEM
features
   Error_Free: initial error state;
   TempErr: error state;
   PermErr: error state;
   ErrND: error state;
   Failed: error state;
   Temp_Fault: error event {Occurrence⇒Poisson λ1};
   Perm_Fault: error event {Occurrence⇒Poisson λ2};
   Recover: error event {Occurrence⇒Poisson λ3};
   Repair: error event {Occurrence⇒Poisson λ4};
   Detect: error event {Occurrence⇒Poisson λ5};
   NonDetect: error event {Occurrence⇒Poisson λ6};
   PerceiveFail: error event {Occurrence⇒Poisson λ7};
   H_Err: out error propagation {Occurrence⇒fixed λ8};
   H_FailedVisible: out error propagation {Occurrence⇒
                                          fixed λ9};
   H_OK: out error propagation {Occurrence⇒fixed λ10};

end procEM;

error model implementation procEM.impl
transitions
   Error_Free–[Perm_Fault]→PermErr;
   Error_Free–[Temp_Fault]→TempErr;
   TempErr–[Recover]→Error_Free;
   PermErr–[Detect]→Failed;
   PermErr–[NonDetect]→ErrND;
   ErrND–[PerceiveFail]→Failed;
   Failed–[Repair]→Error_Free;
   TempErr–[out H_Err]→TempErr;
   Failed–[out H_FailedVisible]→Failed;
   Error_Free–[out H_OK]→Error_Free;
end procEM.impl;
```

T_GPS_Reader

```
error model tgrEM
features
   Error_Free: initial error state;
   Activation: error state;
   PermErr: error state;
   ErrND: error state;
   ErrD: error state;
   Failed: error state;
   InRestart: error state;
   Fault: error event {Occurrence⇒Poisson λ1};
   Detect: error event {Occurrence⇒Poisson λ2};
   NonDetect: error event {Occurrence⇒Poisson λ3};
   Eliminate: error event {Occurrence⇒Poisson λ4};
   Fail: error event {Occurrence⇒Poisson λ5};
   Recover: error event {Occurrence⇒Poisson λ6};
   Restart: error event {Occurrence⇒Poisson λ7};
   PerceiveFail: error event {Occurrence⇒Poisson λ8};
   H_Err: in error propagation;
   H_FailedVisible: in error propagation;
   H_OK: in error propagation;
end tgrEM;

error model implementation tgrEM.impl
transitions
   Error_Free–[Fault]→Activation;
   Activation–[NonDetect]→ErrND;
   ErrND–[Recover]→Error_Free;
   ErrND–[PerceiveFail]→Failed;
   Activation–[Detect]→ErrD;
   ErrD–[Eliminate]→Error_Free;
   ErrD–[Fail]→Failed;
   Failed–[Restart]→Error_Free;
   Error_Free–[in H_Err]→Activation;
   Error_Free–[in H_FailedVisible]→Failed;
  InRestart–[in H_OK]→Error_Free;
end tgrEM.impl;
```

Fig.10 Error model for Proc1 and T_GPS_Reader

图 10 Proc1 和 T_GPS_Reader 错误模型

组成飞行控制系统架构的 AADL 可靠性模型后,就可以利用 ARAM 工具实现飞行控制系统架构的 AADL 可靠性模型向 GSPN 模型的转换,然后对 GSPN 模型进行可靠性分析与评估.首先,对这里主要是说明 AADL 可靠性模型和 GSPN 模型之间的转换,飞行控制系统 AADL 可靠性模型转换到 GSPN 的结果如图 11 所示.为方便阅读,图中省去了内存构件 Mem1,Mem2 和总线的 GSPN 模型以及作动器的 GSPN 模型,其中也省去了每个 Out Propagation 的吸收;因为位置 NoOutProp 没有向外发出错误,对其他构件没有产生影响,因此也在图 11 中省略.

经过 AADL 可靠性模型向 GSPN 可靠性计算模型转换后,可以利用 PIPE2 工具对 GSPN 模型进行分析,得到模型的可靠性评估结果.分析图 9 的飞行控制系统中构件对整个系统的影响可以看出,一旦 GPS 构件出错后,那么对整个系统的影响是灾难性的.因此,本文通过改变 GPS 的错误事件 Fail 的发生概率,观察飞行控制系统中其他构件的可靠性值变化.图 12 中列举了 5 个受 GPS 构件错误事件影响的构件可靠性值,可以看出,受 GPS 构件错误事件影响的构件可靠性值变化主要可以分成 3 类:

a) 受影响最大的构件.构件 GPS 本身是受 GPS 错误事件影响最大的构件,从 GPS 读取数据的构件 T_GPS_Reader 直接受 GPS 错误传播的影响,其受影响度也比较大;

b) 受影响较小的构件.构件 T_AP_Computer 和 T_Screen_Disp 是受影响比较小的构件,因这两个构件主要是受构件 T_GPS_Reader 的影响,T_AP_Computer 从 T_GPS_Reader 读取数据进行计算,T_Screen_Disp 是从 T_GPS_Reader 读取数据进行显示,这两个构件相当于间接受到 GPS 错误事件变化的影响;

c) 不受影响的构件.构件 T_Pilot_Input 的可靠性值随 GPS 错误事件值的变化没有任何改变,没有受到

GPS 的影响.如图 9 所示,这两个构件没有任何直接或者间接的依赖关系.

构件错误传播对其他构件的影响作用不是保持不变的,从图 13 中可以看出,随着 GPS 错误事件值的变化,对其他构件的影响逐渐减少,其趋势是逐渐向零变化.也就是说,GPS 错误事件造成的影响在逐渐缩小.与上面 3 种影响一一对应:

a) 受错误事件影响比较大的构件.这些构件可靠性变化斜率的绝对值也比较大.如构件 GPS 和 T_GPS_Reader,这两个构件随着 GPS 错误事件值的影响越来越小,而且变化得也比较快;

b) 受错误事件影响比较小的构件.这些构件可靠性变化斜率的绝对值相对也比较小.如构件 T_AP_Computer 和 T_Screen_Disp,这两个构件的变化曲线如图 12 所示,发生了重叠.也就是说,这两个构件的受影响度基本一致;

c) 不受影响的构件.这些构件的受影响度应该是一直保持为零,不受到错误事件值变化的影响.如构件 T_Pilot_Input,这个构件不受 GPS 的影响,因此可以看到它的可靠性值的变化保持为 0.

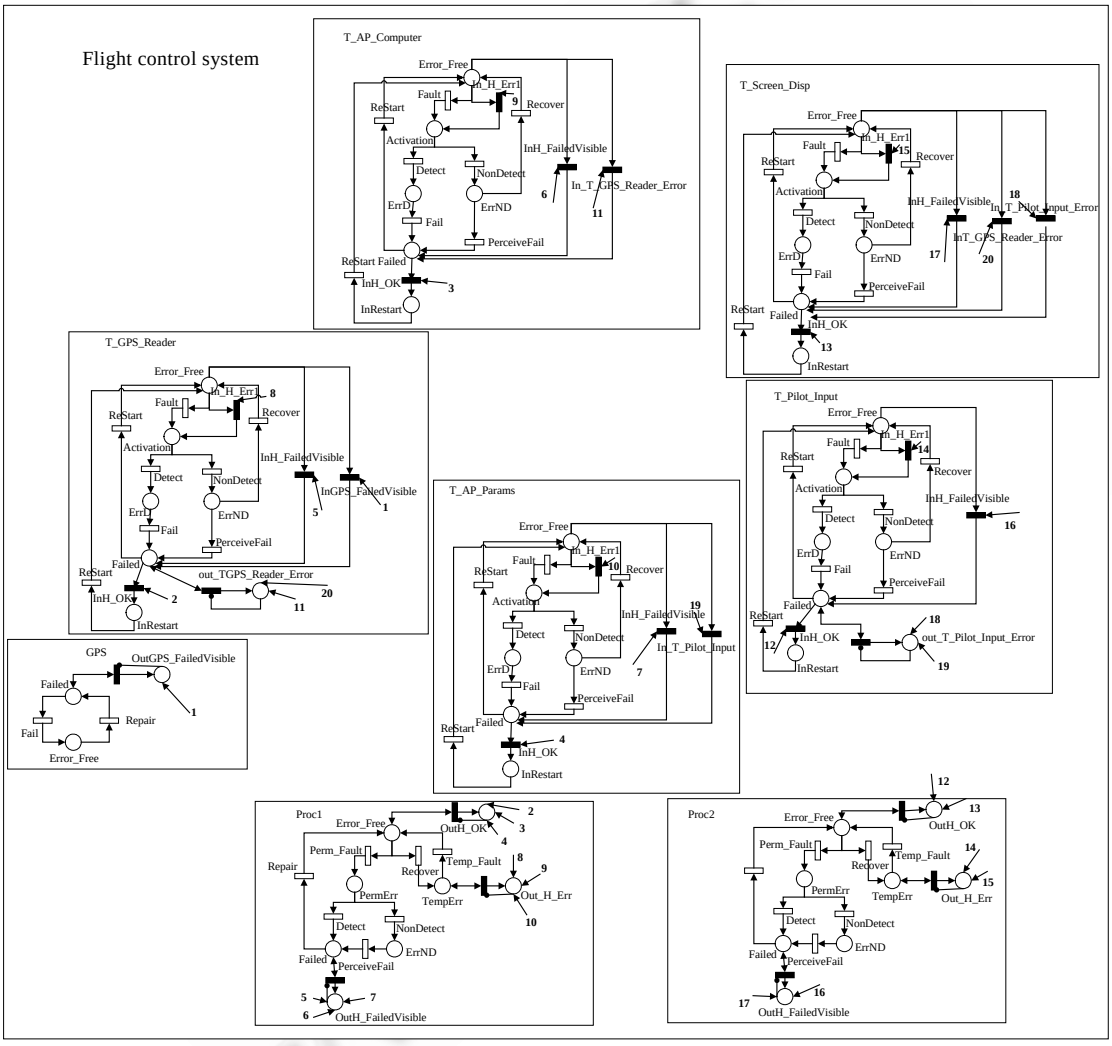


Fig.11 GSPN model of flight control system

图 11 飞行控制系统的 GSPN 模型

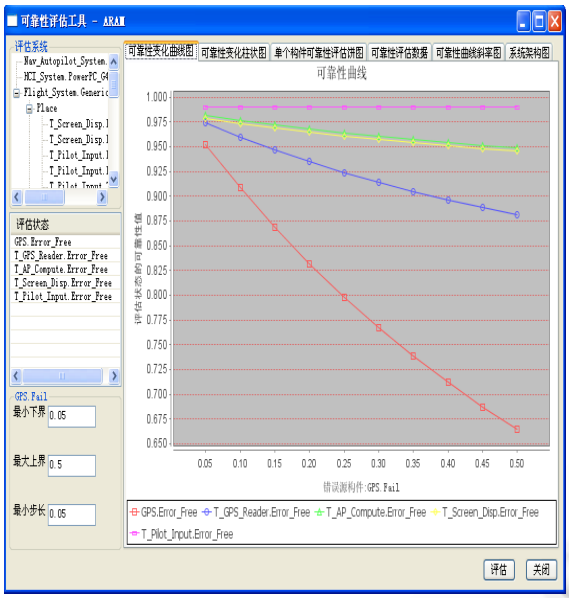


Fig.12 Reliability curve of component

图 12 构件可靠性曲线

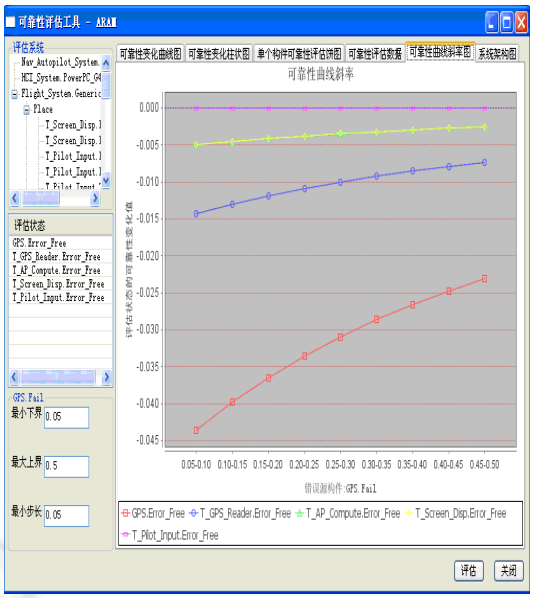


Fig.13 Reliability slope of component

图 13 构件可靠性变化斜率

通过对数据进行分析可以看出,在系统架构设计过程中,可以根据系统对各个构件的可靠性需求来调整关键构件的错误事件的发生概率,直到满足构件的可靠性需求为止.如果通过调整错误事件的发生概率不能满足需求时,那么就需要在构件这一级别对系统进行重新设计.

6 总 结

本文首先描述了 AADL 可靠性模型基本元素向 GSPN 模型元素的形式化转换,并以这些转换规则为基础,设计与实现了可靠性分析与评估工具 ARAM,实现了 AADL 可靠性模型向 GSPN 可靠性模型的自动转换,并最终实现嵌入式系统可靠性分析与评估.

本文结合飞行控制系统对 ARAM 工具的应用进行说明,使用 AADL 对飞行控制系统进行架构建模,同时根据飞行控制系统中构件实际的运行情况建立对应错误模型,利用 ARAM 工具对飞行控制系统可靠性模型进行分析与评估.根据可靠性分析结果,可以判断飞行控制系统的架构设计是否满足需求.如果不满足就可以在架构级别对飞行控制系统进行重新设计,提高系统的开发效率.

致谢 在此,我们向对本文的工作给予支持和建议的陕西省嵌入式系统技术重点实验室的张琛雨、杨俊、朱宇峰、畅绍枫、侯永、蒋虎等同学表示感谢.

References:

[1] OMG. Model driven architecture (MDA). 2010. http://www.omg.org/mda/

[2] Feiler PH, Gluch DP, Hudak JJ. The Architecture Analysis & Design Language (AADL): An Introduction. Int'l Society of Automotive Engineers, 2006.

[3] Yang ZB, Pi L, Hu K, Gu ZH, Ma DF. AADL: An architecture design and analysis language for complex embedded real-time systems. Journal of Software, 2010,21(5):899−915 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/21/3700.htm [doi: 10.3724/SP.J.1001/2010.03700]

[4] Open source AADL tool environment (OSATE). 2010. http://www.aadl.info/aadl/currentsite/tool/osate.html

[5] Varona-Gomez, R, Villar E. AADL simulation and performance analysis in SystemC. In: Proc. of the 14th IEEE Int'l Conf. on Engineering of Complex Computer Systems, 2009. 323−328. [doi: 10.1109/ICECCS.2009.11]

[6] Isograph. 2011. http://www.isograph-software.com/

[7] Liu JJ, Zhong S, Ye H. Reliability modeling for airborne equipment system using AADL. Aeronautical Computing Technique, 2009,39(2):90−94 (in Chinese with English abstract).

[8] Feiler P, Rugina A. Dependability Modeling with the Architecture Analysis & Design Language (AADL). Int'l Society of Automotive Engineers, 2007.

[9] International Society of Automotive Engineers. SAEAS5506/1: SAE Architecture Analysis and Design Language (AADL) Annex Vol.1: Annex E: Error Model Annex. 2006.

[10] Rugina AE. Dependability modeling and evaluation—From AADL to stochastic Petri nets in systèmes informatiques [Ph.D. Thesis]. Toulouse: Institut National Polytechnique de Toulouse, 2007. 151.

[11] Rugina AE, Kanoun K, Kaâniche M. A system dependability modeling framework using AADL and GSPNs. Architecting Dependable Systems IV. LNCS 4615, Springer-Verlag, 2007. 14−38. [doi: 10.1007/978-3-540-74035-3_2]

[12] Zhang CY, Yang ZY, Dong YW. Research and assessment of the reliability of a fault tolerant model using AADL. In: Advanced Software Engineering & Its Applications. 2008. 45−52. [doi: 10.1109/ASEA.2008.21]

[13] Bause F, Kritzinger PS. Stochastic Petri Nets: An Introduction to the Theory. 2002. 129−161.

[14] Platform independent Petri net editor 2(PIPE2). 2007. http://pipe2.sourceforge.net/

附中文参考文献:

[3] 杨志斌,皮磊,胡凯,顾宗华,马殿富.复杂嵌入式实时系统体系结构设计与分析语言:AADL.软件学报,2010,21(5):899−915. http://www.jos.org.cn/1000-9825/21/3700.htm [doi: 10.3724/SP.J.1001/2010.03700]

[7] 刘建军,钟珊,叶宏.基于 AADL 的机载设备系统可靠性建模.航空计算技术,2009,39(2):90−94.

董云卫(1968－),男,云南大理人,博士,教授,博士生导师,CCF 会员,主要研究领域为嵌入式软件设计与验证,信息物理融合系统建模与分析.

张凡(1979－),男,博士,讲师,主要研究领域为嵌入式软件设计与验证.

王广仁(1987－),男,硕士,主要研究领域为嵌入式系统软件可靠性验证.

高磊(1984－),男,硕士,主要研究领域为嵌入式系统软件可靠性验证.