一种 AADL 故障模型到动态故障树的转换方法

张晓策,燕雪峰,周 勇
(南京航空航天大学 计算机科学与技术学院,江苏 南京 211106)

摘 要:在基于模型驱动的嵌入式软件开发中,需要使用工程技术和工具保证其安全性和可靠性。在设计阶段确定系统的可靠性是非常重要的,因其可为设计决策提供重要信息,以减少系统的开发成本。应用 AADL 建立嵌入式系统模型时,存在着 AADL 对时序故障描述不足的问题。为解决该问题,将 AADL 的故障附件进行扩展,使其能够完善地描述时序故障,并提出了扩展的 AADL 故障模型到动态故障树的转换规则及方法。从扩展的 AADL 故障模型到动态故障树的转换过程共分为三步:将 AADL 模型中的组件转换为单个动态故障树;以组件为基本元素建立数据或事件的故障转移图;根据故障转移图和组件的动态故障树建立动态故障树。通过导弹发射系统实例证明了该转化规则及方法的可行性和实用性。

关键词:AADL 模型;动态故障树;转换方法;错误附件
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2017)11-0110-05
doi:10.3969/j.issn.1673-629X.2017.11.024

A Method for Conversion of AADL Model into Dynamic Fault Tree

ZHANG Xiao-ce, YAN Xue-feng, ZHOU Yong
(School of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China)

Abstract: In the embedded software development based on the model driven, it needs to use the engineering techniques and tools to ensure its safety and reliability. It's very important to determine the system reliability at the design stage, because it can provide important information for design decisions and reduce the development cost of the system. When adopting the AADL to model for embedded software, there have been a problem that the AADL is poor to describe the sequential error. In order to solve this problem, the error model annex of AADL has been modified to describe the sequential fault perfectly. The conversion rule and method of extended AADL fault model to dynamic fault tree is proposed, which is divided into three steps: transforming the components in the AADL model into a single dynamic fault tree; establishing the failover diagram of the data or event; based on the dynamic fault tree of component and the failover diagram building the dynamic fault tree. Finally, its feasibility and practicability are verified by the example of missile launch system.

Key words: AADL model; dynamic fault tree; conversion method; error model annex

0 引 言

随着嵌入式系统的飞速发展,嵌入式系统的结构越来越复杂,规模越来越大,对系统的开发成本、开发周期及可靠性的要求也越来越高。传统的嵌入式系统开发方法已不能满足嵌入式系统开发的需要。因此,业界引入了一种新的方法,模型驱动结构方法(Model Driven Architecture, MDA)[1]。嵌入式系统的开发被提升到模型级。模型成为开发过程中的核心。在系统设计阶段对系统架构进行判断和修改,提高了系统的可靠性,缩短了开发周期,节约了开发成本。针对这样的需求,美国自动化工程师协会发布了一种架构分析与设计语言(Architecture Analysis and Design Language, AADL)[2-3]。AADL 能够详细地描述复杂的嵌入式系统架构。它不关心构件内部的具体实现,仅仅通过构件与构件间的交互、软件构件与硬件构件的绑定,对嵌入式系统进行描述与分析。AADL 语言虽然为安全实时系统提供了强大的表达能力,但其大部分语义仍然采用自然语言和例子进行解释。目前,模型转换是 AADL 故障模型形式化验证与分析的主要途径。例如,将 AADL 模型转换到 BIP、Petri 网、故障树

收稿日期:2016-10-24 **修回日期**:2017-02-24 **网络出版时间**:2017-08-01
基金项目:"十三五"重点基础科研项目(JCKY2016206B001)
作者简介:张晓策(1991-),男,硕士研究生,研究方向为系统建模与仿真;燕雪峰,教授,研究方向为软件工程方法论、系统建模与仿真等;周勇,副教授,研究方向为人工智能、专家系统、智能推理等。

等,目的是为了重用这些模型上已有的验证和分析能力[4-5]。

时序故障是一类由底事件发生的顺序产生的故障。AADL 描述嵌入式系统体系结构时,虽然考虑了事件或进程的结束时间,但是缺乏对事件或进程的时序故障描述。通过对 AADL 错误模型附件的扩展,增加了时序故障的描述。并对系统故障进行建模,通过转换规则和算法将系统故障模型转换为时序故障树,用于分析系统的安全问题。

1　AADL 模型及错误附件

AADL 可以支持一个完整的基于模型的嵌入式软件的开发生命周期,包括文档设计、架构分析、代码生成、系统集成指导及系统优化和升级等。

错误模型附件[6](Error Model Annex)是 AADL 模型的一种标准扩展,主要用于描述系统及构件的可靠性,为 AADL 构件错误和构件通信错误提供建模依据。它可以建立单个构件的错误模型,也可以建立构件通信产生的错误传播模型。

AADL 可靠性模型由两部分组成[6]:AADL 架构模型和 AADL 错误模型。AADL 架构模型可以从软件和硬件两个方面描述系统,主要描述系统各构件的属性、构件间的联系及软件和硬件构件的绑定关系。它不关心这些功能的实现过程,只是描述构件实现怎样的功能。构件的描述由类(type)和实现(implementation)构成[7]。构件类中定义构件的属性,如端口、子程序、参数等;构件实现中定义构件实现的功能,如包含的子构件、子程序调用、连接、流、模态等。AADL 错误模型主要描述的内容包括:错误状态、错误事件、错误变迁以及相关参数等可靠性信息。错误模型与构件相对应,每一个构件都通过错误模型子附录与相应的错误模型联系。

错误模型子附录通常在架构模型构件的实现中声明。由于错误是通过构件间的连接、软硬件的绑定或子程序调用进行传播的,需要在错误模型子附录中定义错误的过滤与屏蔽规则(guard_in,guard_out)和连接错误状态(guard_event,guard_transition,activate\deactivate transitions)。这些语法规则很好地描述了错误是如何在构件间传播的。

2　扩展的 AADL 错误附件

AADL 中的构件对时间有相关的描述,包括线程的周期、执行时间、截止时间等属性。在 AADL 错误模型中,构件的错误传播过滤与屏蔽规则缺少对多个事件和数据发送到构件的先后顺序的描述,也就是缺少对事件和数据的时序关系的描述。例如,当某个构件有一个事件输入和一个数据输入,当事件或数据的错误,或者事件输入发生在数据输入之后,会导致该构件失效。在 AADL 中,虽然对输入事件和数据的构件有相关的时间描述,但是执行时间和截止时间的描述会有一部分重叠,造成无法判断事件和数据的输入先后顺序。针对这个问题,对 AADL 错误模型进行扩展,加入对时序错误的描述。

AADL 错误模型中的错误事件、错误状态及状态间的变迁可以采用一个三元组模型描述。三元组为:EM=(ES,EE,TR)。其中,ES 是所有错误状态的集合,$ES=\{es_1,es_2,\cdots,es_m\}$;EE 是所有错误事件的集合,$EE=\{ee_1,ee_2,\cdots,ee_m\}$;TR 是所有错误状态间变迁的集合,转移函数 $TR(es_i,ee_j)=es_k$。其中 ee_j 可以是 EE 中某些错误事件的组成,包括 $ee_i \cap ee_j$,$ee_i \cup ee_j$,$ee_i < ee_j$,$ee_i \,\&ee_j$,$ee_i \mid ee_j$;包含 3 种时序情况:$ee_i < ee_j$ 表示 ee_i 在 ee_j 之前发生造成错误;$ee_i \,\&ee_j$ 表示 ee_i 和 ee_j 同时发生造成错误;$ee_i \mid ee_j$ 表示 ee_i 发生,ee_j 不发生或者 ee_i 在 ee_j 之前发生造成错误。

3　转换规则

目前,AADL 模型的形式化验证与分析的主要途径是模型转换,主要是将 AADL 模型转换到 BIP、Petri 网、故障树等。文献[8-11]描述了从 AADL 模型到静态故障树的转换方法,但是这些方法中缺少对数据组件等的转换描述。文献[12]描述了从 AADL 故障模型到动态故障树的转换方法,但该模型中缺少对时序故障的描述。

采用故障树模型作为系统架构模型的可靠性计算模型,并建立 AADL 可靠性模型与故障树可靠性模型之间的对应关系。动态故障树能够准确描述系统中各种事件之间的因果关系。

AADL 模型是通过组件及组件之间的交互来描述和分析系统结构。AADL 模型的组件按照其性质可以大致分为 3 类:系统组件、软件组件和硬件组件。AADL 模型中构件间的依赖关系包括两种,分别是端口\连接实现和调用实现。调用实现包括内存调用、函数调用等。端口\连接包括线程构件、进程构件等。文中将从两种不同的依赖关系出发,讨论两种不同的构件转换到动态故障树的规则。

3.1　转换规则一

在 AADL 模型的组件中,以端口\连接实现为依赖关系的构件包括:进程组件、线程组件、硬件构件等。本节主要描述这些构件到动态故障树的转换规则。

扩展的 AADL 错误模型与故障树模型之间存在映射关系。其关系如下:

定义 1:动态故障树可以用一个 6 元组表示,$F=$

{BE,ME, G,T,I ,TE}。动态故障树是以 TE 为根节点,由(BE∩ G ,ME, I)组成的有向无环图。其中,BE 是基本事件的集合;ME 是故障树的中间状态集合; G 是门的集合,即时序故障树中出现的门; T 是每个门的解释, T ={And,Or,Sand,Pand,Por}; I 是每个门的输入;TE 是时序故障树的根节点。

根据定义 1,实现 AADL 基本错误模型元素向故障树模型元素转换规则,如下所述:

(1)EM(EE)= F (BE),错误事件的集合和初始状态转化为基本事件的集合。

(2)EM(ES)= F (ME),所有错误状态转换为故障树的中间状态集合。

(3)EM(TR)= F (I ∩ G),所有错误状态间变迁的集合转换为故障树的门节点及其输入。

错误状态间的变迁集合分为两部分:构件内部的状态变迁和构件间故障传播的条件。对于转换规则(3),主要是构件间故障的传播。对应 AADL 中的过滤与屏蔽规则 guard_in 和 guard_out。

把错误过滤/屏蔽 guard_in 及 guard_out 中定义的规则按照其逻辑关系转换为相对应的逻辑口,状态和传播转换为中间事件。错误过滤/屏蔽后的组件状态为结果事件,其余为原因事件。通过逻辑门将各个元素连接起来。对于错误过滤 guard_in 来说,当输入为两种以上的数据或事件时,这些输入的时序会对系统产生不同的影响。根据 AADL 错误模型中的 TR,所有错误状态间变迁的集合,转换成相应的逻辑门。例如 $ee_i < ee_j$,转换为优先与门(PAND 门), ee_i 和 ee_j 作为 PAND 门的输入事件。

3.2 转换规则二

在 AADL 模型的组件中,以调用实现为依赖关系的构件包括数据访问、子程序调用等。本节主要描述这些构件到动态故障树的转换规则。

在数据访问中,外部构件通过关键字 data access 访问数据构件。数据构件属于软件构件。当其向外传播错误时,本身的状态也发生变迁。数据构件定义两个最基本的错误状态:Data_Error Free、Data_Failed,分别代表数据构件是否处于错误状态。在系统架构模型中,根据外部构件访问数据构件的关键字 data access 可以确定错误传播的方向,从数据构件到外部调用构件。数据访问为系统模型提供数据共享的功能,当多个子构件访问同一个数据构件时,将其拆分为一对一的数据访问来进行错误模型的转换。

对于数据访问,将数据访问事件转换为条件事件,将总线所连接的构件转换为中间事件(即访问构件和被访问构件)。访问构件和访问事件所转换的事件为原因事件;被访问构件所转换的事件为结果事件。各个元素之间通过相应的逻辑口 AND 相连接。一个数据访问实例转换为相应的构件故障树如图 1 所示。

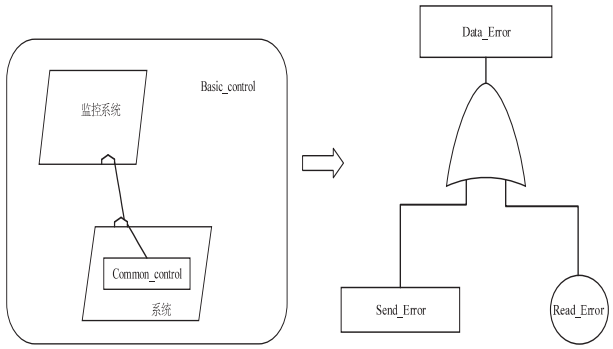

图 1 内存构件转换图

4 转换流程

故障树是对系统进行安全分析的一种重要方法。从 AADL 故障模型中提取动态故障树,将动态故障树输入到已有的故障树分析软件中,可以自动地分析嵌入式系统模型的可靠性。

时序故障树的生成是一个递归过程,主要分成三个步骤:

(1)单个构件的故障树生成。

构件包括 AADL 模型中定义的进程、子程序、总线、设备、存储器等。每种构件中都定义了所有错误状态、错误事件、错误状态间的转移等。根据基本构件转换规则、内存构件转换规则和总线构件转换规则,将构件转换为单个构件的故障树。

(2)建立构件间的故障传播图。

AADL 模型中的构件可以分两种,分别是依赖关系为端口\连接实现的构件和依赖关系为调用实现的构件。构件的故障传播也分两种:构件内部的故障传播和构件间的故障传播。在该步骤中,将所有的构件抽象为一个点,根据依赖关系将这些构件连接,建立故障传播图。

构件包括进程组件、线程组件、子程序组件、数据组件、硬件构件等。硬件构件和进程构件可以由线程构件、数据构件等组合而成。将这些构件分解为线程构件、数据构件等不可分隔的构件,通过依赖关系完成这些构件的转换,建立单个构件的故障传播图。最后在整个系统中,根据各构件的依赖关系建立整体的故障传播图。

(3)故障树的生成。

将故障传播图中的各点替换为步骤 1 中生成的构件故障树,完成整体故障树的建立。

5 实 例

本节将对导弹发射流程系统进行 AADL 可靠性建

模,接着把导弹发射流程系统的 AADL 可靠性模型转换到对应的时序故障树模型。导弹发射流程系统主要是导弹发射的控制和数据的传送。该系统主要关注其导航信息的传送和发射流程的控制。系统主要是上端发送基本的发射指令,包括导航信息和电源控制命令两部分,下端根据发射指令,将导航信息进行计算并封装下发到导弹中,控制导弹的电源开启。导弹的电源开启必须在导航信息发送之前完成,因为导弹必须在电源开启的状态下才能正常工作,即可以接收导航信息。导弹发射流程系统的 AADL 架构模型如图 2 所示。

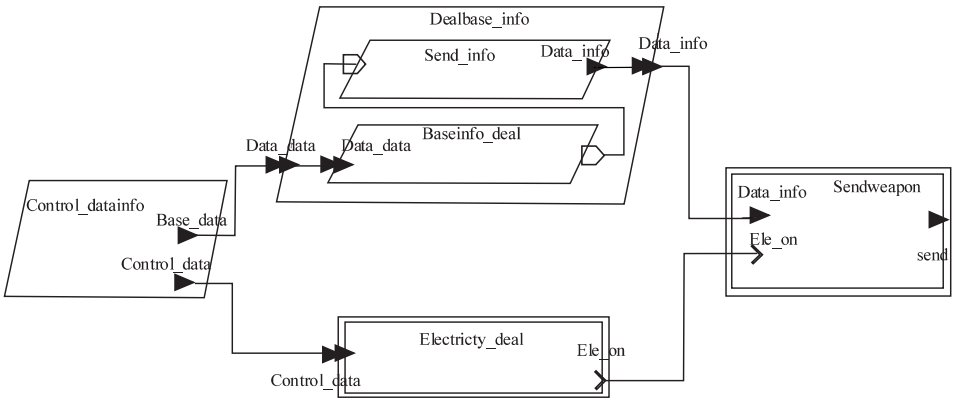

图 2　导弹发射系统的 AADL 图

(1)单个构件的故障树生成。

以 Sendweapon 构件为例,构件代码如下:

```
Error_model Sendweapon
Features
EroorFree:initial error state;
Faulse,Failed:error state;
Sendfail:error event;
Transitions:
ErrorFree-[sendfaile]->Failed;
Faulse->Failed;
Guard_in
ErrorFree->faulse when mixdata_error or ele_on_error or(mixdata pand ele_on)
End error model
```

将该构件转换为构件的故障树,如图 3 所示。

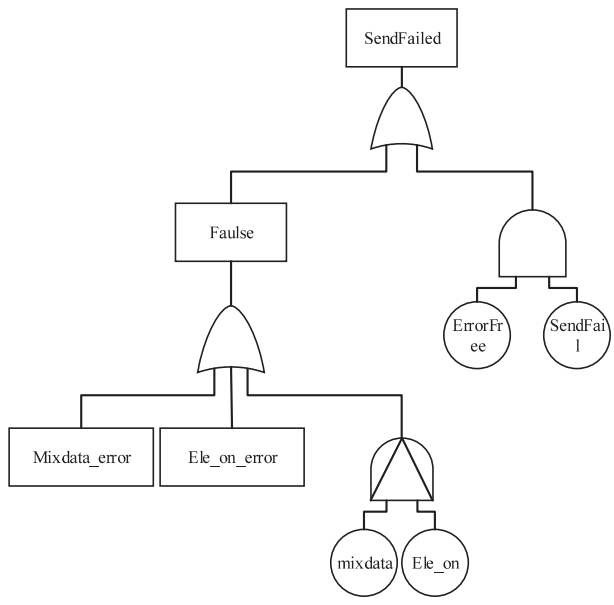

图 3　Sendweapon 构件转换后的故障树

(2)构件的故障传播图。

根据转换流程的故障传播图生成过程,建立导弹发射流程系统的故障传播图。Control_datainfo,Electricty_deal 和 Sendweapon 组件转换为相应的节点。Dealbase_info 组件可以分为线程 Baseinfo_deal 和 Send_info,以及数据访问构件 Data_rap。将 Dealbase_info 组件转换为相应的节点,得到构件的故障传播图,如图 4 所示。

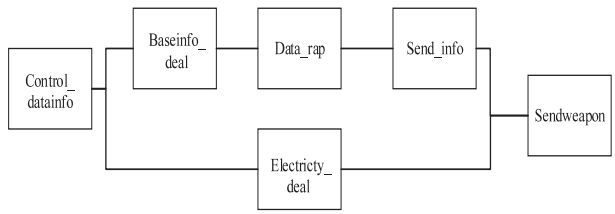

图 4　故障传播图

(3)故障树的生成。

根据故障传播图建立导弹发射流程系统的故障树,如图 5 所示。

6　结束语

针对 AADL 故障模型中缺少时序故障描述的问题,对 AADL 的错误附件进行了相应扩展,增加其对时序故障的描述,进而提出从扩展的 AADL 故障模型到动态故障树的转换规则和转换方法。以导弹发射流程为例,详细描述了从 AADL 故障模型到动态故障树的转换过程,证明了该转换规则与转换方法的可行性以及有效性。

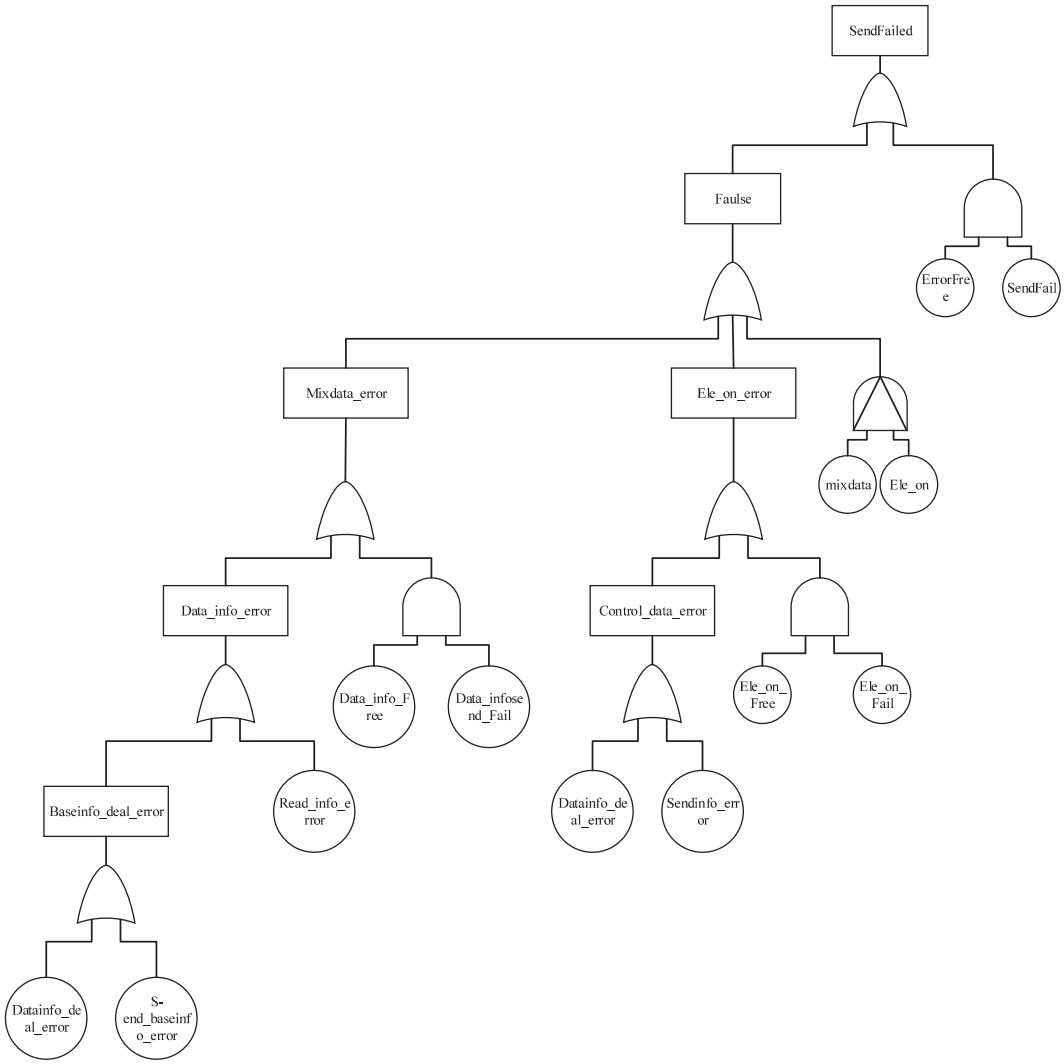

图 5 导弹发射流程的故障树

参考文献:

[1] Soley R. Model driven architecture:three years on[M]//CoopIS,DOA,and ODBASE. Berlin:Springer-Verlag,2003:1048-1049.

[2] Singhoff F,Plantec A. AADL modeling and analysis of hierarchical schedulers[C]//Proceedings of the 2007 ACM international conference on SIGAda annual international conference. New York:ACM,2007:41-50.

[3] 杨志斌,皮 磊,胡 凯,等. 复杂嵌入式实时系统体系结构设计与分析语言:AADL[J]. 软件学报,2010,21(5):899-915.

[4] 董云卫,王广仁,张 凡,等. AADL 模型可靠性分析评估工具[J]. 软件学报,2011,22(6):1252-1266.

[5] 苏 威. 基于 AADL 的嵌入式软件系统验证技术研究[D]. 西安:陕西师范大学,2015.

[6] Delange J, Feiler P. Architecture fault modeling with the AADL error-model annex[C]//Software engineering and advanced applications. [s. l.]:IEEE,2014:361-368.

[7] 高 磊,董云卫,张 凡,等. 一种 AADL 系统可靠性模型转换方法[J]. 计算机工程,2011,37(14):21-26.

[8] Sun H, Hauptman M, Lutz R. Integrating product-line fault tree analysis into AADL models[C]//10th IEEE high assurance systems engineering symposium. [s. l.]:IEEE,2007:15-22.

[9] Xiang J,Yanoo K,Maeno Y,et al. Automatic synthesis of static fault trees from system models[C]//Fifth international conference on secure software integration and reliability improvement. [s. l.]:IEEE,2011:127-136.

[10] Joshi A,Vestal S,Binns P. Automatic generation of static fault trees from AADL models[C]//The 37th annual IEEE/IFIP international conference on dependable systems and networks. [s. l.]:IEEE,2007.

[11] 刘 玮,李蜀瑜. 基于 AADL 模型的静态故障树的自动生成[J]. 计算机技术与发展,2013,23(10):99-102.

[12] Dehlinger J,Dugan J B. Analyzing dynamic fault trees derived from model-based system architectures[J]. Nuclear Engineering and Technology,2008,40(5):365-374.