

基于时间 Petri 网的 AADL 模型

吴育春,李蜀瑜

(陕西师范大学 计算机科学学院,陕西 西安 710062)

摘要:随着嵌入式系统设计日益复杂,对系统的描述也提出了越来越高的要求,特别是实时性能。因此如何描述实时性能逐渐受到重视。在嵌入式系统中,线程的时间属性影响着系统的实时性,因此在设计早期利用 AADL 描述线程,但是 AADL 只是一种半形式化的语言,不够精确,并可能产生二义性。因此,把 AADL 模型转化为时间 Petri 网,但是时间 Petri 网不能很好地表示 AADL 的时间属性。故在研究 AADL 模型的时间属性的基础上,针对时间 Petri 考虑区域值延时与固定延时,并把固定延时当作区域值延时的一种特殊情况。最后给出针对 AADL 模型转化为时间 Petri 网的相应规则。

关键词:AADL;时间 Petri 网;转化规则

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2014)02-0088-04

doi:10.3969/j.issn.1673-629X.2014.02.021

AADL Model Based on TPN

WU Yu-chun, LI Shu-yu

(School of Computer Science, Shaanxi Normal University, Xi'an 710062, China)

Abstract: With the increasing complexity of embedded system design, the description of the system is raised higher and higher requirements, especially real-time performance, therefore how would you describe the real-time performance is gradually paid attention. In embedded systems, thread time attribute affects the system in real time, so early in the design use the AADL to describe thread, but the AADL is only a semi-formal language, which is not enough accurate and may incur ambiguity. Thus, the model of AADL is transformed into time Petri nets, but time Petri nets cannot be a good representation of the AADL time property. On the basis of the time properties of AADL model, for the time Petri consider the regional value delay and fixed delay, and fix delay as a special case of the regional value of the delay. Finally, give the corresponding rules of the AADL transformed into the time Petri net.

Key words: AADL; time Petri nets; transformation rules

0 引言

在需求日益增强的今天,嵌入式系统的设计将会变得越来越复杂,特别是在专用领域的嵌入式系统,例如航空、航海和工业控制。这些系统的规模、复杂度和性能需求不断提升,也导致了过去以代码为核心的开发方法已经不能适应这种趋势。为此由 SAE 提出一种架构描述语言(AADL),受到工业界、学术界等的接受和认可,以便把开发的重点从代码转向到模型级。在关键性嵌入式系统设计中要求建立软件与硬件之间的映射,既能满足功能属性又能满足非功能属性。

在 AADL 标准中,大部分都是用自然语言进行描述的。因此 AADL 是一种半形式化语言。它的优点易于对嵌入式系统建模,但是也易于产生二义性。

在用 AADL 描述系统时,该如何转换为形式化语

言。在文中提出规则,以便把 AADL 模型转化为时间 Petri 网模型。

1 AADL 标准简介

现如今嵌入式系统要求越来越复杂,特别要求满足设计者的需求,这使得新的设计方法、新的设计理念和新的工具应运而生。在这样的趋势下,过去以代码为核心的设计理念逐渐淡出研究人员的视线,而模型驱动方法^[1]变成主流,因为它致力于系统的一致性描述,它包含了对系统功能和非功能属性的描述。在这样的理念下,2006年由美国的 SAE 发布一种新的结构描述语言—AADL^[2]。

AADL 用来描述系统的软硬件结构,包含各个组件之间的交互。它非常适合描述关键性嵌入式系统。

收稿日期:2013-04-25

修回日期:2013-07-28

网络出版时间:2013-11-29

基金项目:教育部博士点基金项目(200807180005)

作者简介:吴育春(1988-),男,硕士研究生,研究方向为嵌入式系统设计;李蜀瑜,副教授,硕士生导师,研究方向为嵌入式系统、云计算。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20131129.0946.040.html>

AADL 标准^[3]由文本和图形语言、XML/XMI 交换格式、行为附件和错误模型附件组成,其中 AADL 是 AADL 标准的核心,AADL 标准中建模元素以及它们之间的关系见图 1^[2]。它以组件为中心,允许系统设计师先把注意力集中在各个组件的外部通信上,然后再分别对各个组件进行细化和实现。AADL 的组件分为三种:软件、硬件和系统。

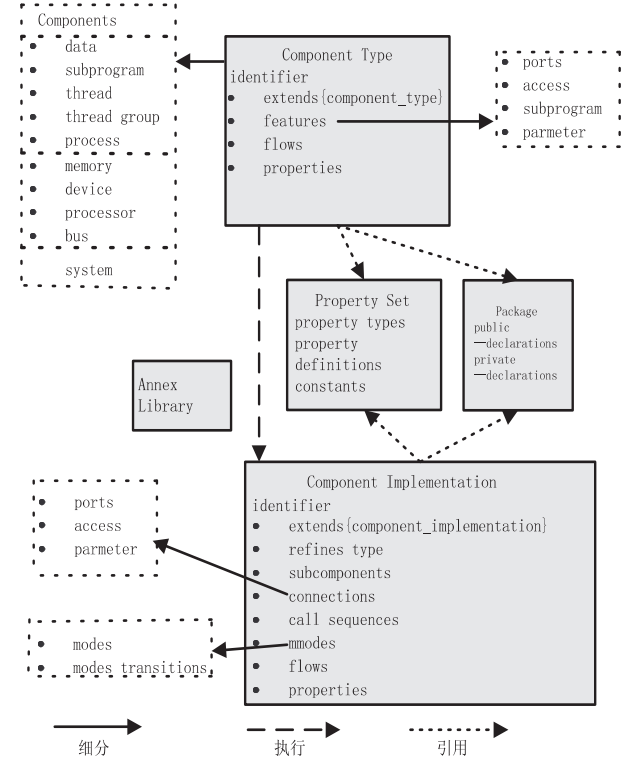


图1 AADL 建模元素和它们之间的关系

软件组件包含数据、线程、线程组、子程序和进程,硬件组件包含存储器、总线、处理器和设备,最后组合组件包含系统。借助标准中定义的“属性”来表示一个系统的行为,例如线程调度、系统模式;在标准中每一个建模元素都含有非功能属性,例如时间、周期等。这些非功能属性描述了 AADL 模型中组件的非功能方面,例如线程调度条件(周期或者非周期)、接口的说明和组件之间的交互。这些属性深深地影响着系统的行为。对于 AADL 的介绍见文献[2]。

AADL 支持对系统的不同层面和不同阶段进行建模^[4-5],AADL 首先关注于系统各个模块之间的交互,并逐渐对系统的每一个模块进行细化。AADL 建模过程如下:

1) 根据需求分析,构建组件类型,即:Component Type,并在此基础上,根据实际情况有选择性地填上 Component Type 的一些组成元素,例如 feature。

2) 为每一个组件类型创建 Component Implementation,这是对组件类型的进一步细化。这里也选择性地添加组件实现中需要的子组件、属性等。

3) 最后,建立 system 的组件类型和组件实现,以表明系统的边界,接着在系统实现中填充各个上述定义各个组件、连接组件之间的接口,以及含有的一些属性包。

AADL^[6]能很好地描述系统,但是它是半形式化语言。在下文中,介绍时间 Petri 网形式化语言,并介绍如何把 AADL 模型转化为时间 Petri 网模型。

2 时间 Petri 网简介

Petri^[7-9]网以强大的数学定义、直观的图形描述闻名,它能精确定义系统。近几十年来,Petri 网的抽象和描述能力得到了加强,从而衍生出了高级 Petri 网,其中包括时间 Petri 网。时间 Petri 网对应于系统的动态模型。

结合 AADL 线程的时间属性,下面给出时间 Petri 网的定义。

定义 1: TPN (时间 Petri) = (P, T, F, K, M_0, W, D) 为时间 Petri 网,其中:

$P = \{p_1, p_2, \dots, p_n\}$ 是一个有限库所集, $n \geq 0$ 。

$T = \{t_1, t_2, \dots, t_m\}$ 是一个有限变迁集,其中 $T = T_{\text{immediation}} \cup T_{\text{delay}}$, 且 $T_{\text{immediation}} \cap T_{\text{delay}} = \emptyset$, $T_{\text{immediation}}$ 是立即变迁, T_{delay} 是延时变迁。

$F \subseteq (P \times T) \cup (T \times P)$, 弧的集合。

$K \subseteq N \cup \{\infty\}$ 是 P 上的容量函数, N 为正整数集合。

$W: F \rightarrow N$ 为弧的容量函数, N 为正整数集合。

$D = \{D_1, D_2, \dots, D_M\}$ 为一组 T_{delay} 上相联系的时延参量。

M_0 是初始标识集。

定义 2: 在 TPN 中 t 使能的定义:

$$M(p_i) \geq W(p_i, t) \wedge M(p_j) + W(t, p_j) \leq K(p_j) \\ \forall p_i \in {}^*t, \forall p_j \in t^*$$

定义 3: $T_{\text{immediation}}$ (立即变迁), 即当某一变迁的发生条件满足时,该变迁立即从相应的输入库所中移走相应的托肯,且立即得到发生后果。定义如下:

当 $t \in T_{\text{immediation}} \wedge t$ 满足定义 2, 则:

$$M'(p) = \begin{cases} M(p) + W(t, p) & p \in t^* - {}^*t \\ M(p) - W(p, t) & p \in {}^*t - t^* \\ M(p) - W(p, t) + W(t, p) & p \in {}^*t \cap t^* \\ W(p) & \text{other} \end{cases}$$

在上文中,根据 AADL 模型考虑区域值延时与固定延时,并把固定延时当作区域值延时的一种特殊情况。对于任意的时延变迁 $t \in T_{\text{delay}}$ 都有一个对偶 $[P_{\min}, P_{\max}]$ 与之相对应,若是在全局时间 b 时 t 的发生条件成立,则 t 可在时间区间 $[b + P_{\min}, b + P_{\max}]$ 内

执行,即 t 有效时,其相应的输入库所中的托肯将至少保留 P_{\min} ,最大保留 P_{\max} ,若是超过该时间,则认为 t 变迁不能发生了。若是 $P_{\min} = P_{\max}$,则为固定值延时。

定义 4: T_{delay} (延时变迁),即当某一个变迁的发生条件满足时,该变迁要延迟一段时间后才从相应的输入库所中移走相应的托肯并得到发生后果。定义如下:

当 $\text{time}_t \in [b + P_{\min}, b + P_{\max}] \wedge t \in T_{\text{delay}} \wedge t$ 满足定义 2,则输出的结果与定义 3 相同;

若 $\text{time}_t \notin [b + P_{\min}, b + P_{\max}]$,则 t 不能再触发。

3 AADL 转化 TPN

在上文中已经简要介绍了 AADL 建模以及时间 Petri 网的知识,在 AADL 模型中,线程^[10]以及它们之间的交互影响^[11-13]着系统的活动。如何精确地描述出这种关系,时间 Petri 网能很好地补充这一点。结合两者即可容易利用 AADL 建模的优点,之后转化为 TPN,利用它精确地描述系统。

转化规则如下:

规则 1:

Thread state(状态)→P(库所)

线程所经历的状态都用库所表示,特别是线程的初始状态就用待用托肯的库所表示。

规则 2:

Thread event(事件)→T(变迁)

线程处于某种状态,被某一事件触发转为另一状态。

规则 3:

事件相关的时间属性→带有时间参量的变迁

在 AADL 标准中规定了线程的属性,在这些属性中大部分是与时间有关的。这些属性对于系统有着重大影响,因此必须在 Petri 网建模的时候考虑进去。在规则 3 中相关时间属性用带有时间参量的变迁表示,下面给出线程中重要的时间属性的具体表示:

Period:用 Period 变迁表示并使固定值延时。

Compute_Execution_Time:表明一个线程在 CPU 中所消耗的时间,用 Computer 变迁表示并使区域值延时。

Compute_Deadline:这用于说明一个线程被允许执行的最大时间,并用域值延时表示。

Priority:表明线程之间的优先级,若是一个线程在执行,一个比它具有更高优先级的线程将会被执行。

Dispatch_Protocol:用于线程之间的调度规则,在 AADL 模型中有三种调度规则:周期、零星、非周期。规则支持这三种调度模式。

上面的规则给出对一个线程转化为 TPN,除此之

外,线程必须绑定在处理器上和其他线程进行交互。

规则 4:

模式是 AADL 模型中很重要的概念,它用于系统在不同事情发生时,系统不同的配置,在时间 Petri 用 Group 表示系统的模式。不同 Group 表示一个时间 Petri 网模型在当前时间内不同的库所和变迁。

针对用于汽车防碰撞系统的部分 AADL 模型,见下面的文本描述,其中 CrashProcess 含有一个用来周期采样和计算撞击强度数据的线程,另一个线程用于接收撞击数据并确定发送点火指令。

这两个周期线程运行在一个处理器中,即 default_Protocessor,它们的调度协议是 FIFO_Within_Priorities。线程通过它们的端口进行交互。发送线程把计算后的数值发送给接收线程。

```
process Calculate_Crash_strength
features
Crash_date;in data port;
Control_date;out data port;
end Calculate_Crash_strength;
thread Send_Date
features
Crash_date;in data port;
Calculate_date;out data port;
properties
Period => 3 Ms;
Compute_Execution_Time => 1 Ms .. 2 Ms;
Compute_Deadline => 3 Ms;
Dispatch_Protocol =>FIFO_Within_Priorities;
end Send_Date;
thread Received_date
features
Calculate_date;in data port;
Control_date;out data port;
properties
Period => 1 Ms;
Compute_Execution_Time => 1 Ms .. 2 Ms;
Compute_Deadline => 3Ms;
Dispatch_Protocol =>FIFO_Within_Priorities;
end Received_date;
process implementation Calculate_Crash_strength. ProcessImpl1
subcomponents
Send_date;thread Send_Date. ThreadImpl1;
Reveived_Date;thread Received_date. ThreadImpl2;
connections
DataConnection1 ;data port Crash_date -> Send_date. Crash_date;
DataConnection2 ;data port Send_date. Calculate_date -> Reveived_Date. Calculate_date;
end Calculate_Crash_strength. ProcessImpl1;
```

thread implementation Send_Date.ThreadImpl1
end Send_Date.ThreadImpl1;
thread implementation Received_date.ThreadImpl2
end Received_date.ThreadImpl2;

文中给出转化后的 TPN 模型见图 2。

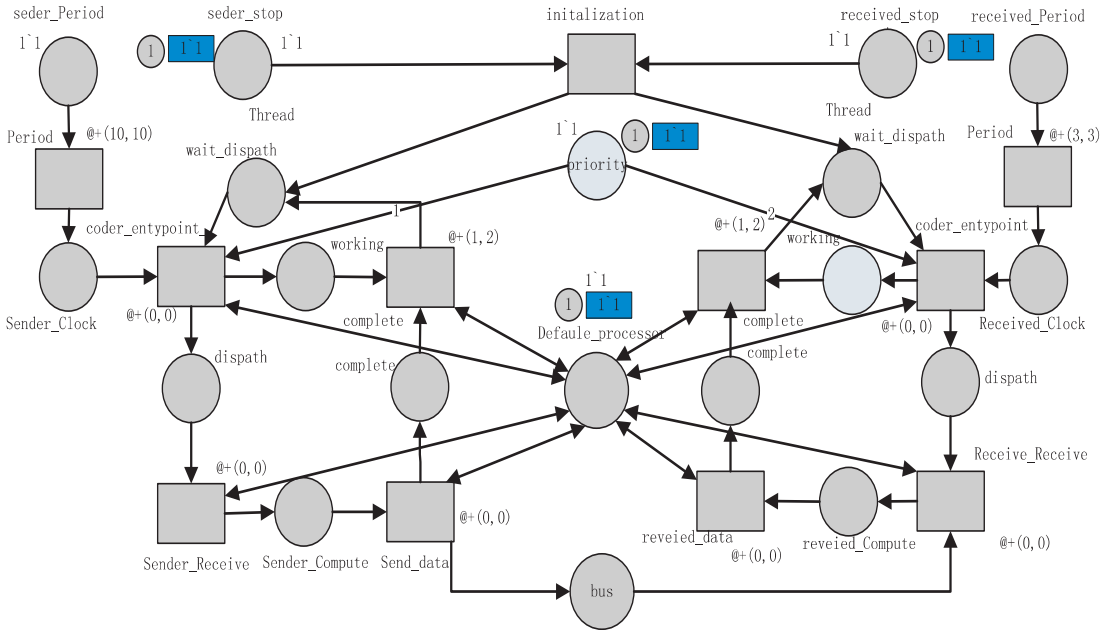


图 2 Calculate_Crash_Strength 进程的时间 Petri 网模型

在 Petri 网模型中,表示两个不同周期线程运行在一个处理器中,其中线程中不同的属性,用上面的规则进行表示。Sender_Stop、Receive_Stop 库所的类型是 Thread(设计者定义),default_Processor 库所的类型是 Process,它的初始标识为 1。为了体现 Send_Date 线程的优先级比 Receive_date 线程高,增加了 Priority 库所。

4 结束语

文中利用 SAE 发布的 AADL 标准对嵌入式系统进行建模,AADL 很容易用于建模,但是描述不太精确。在此基础上,把 AADL 模型通过规则转化为 Petri 网模型,这样就能精确描述系统,不会产生二义性。

将来的研究工作就是在原来时间 Petri 网基础上,继续对其的数学定义进行扩充,并能把描述时间 Petri 网的 CPNTools 工具集成到描述 AADL 的工具中,这样能自动把 AADL 模型转化为 Petri 网模型,并结合系统需求验证系统。

参考文献:

[1] Miller J,Mukerji J. MDA guideversion1.0.1 [R/OL]. [2003]. [http://www.omg.org/cgi-bin/apps/doc? omg/03-06-01.pdf](http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf).

[2] Feiler P H,Gluch D P,Hudak J J. The architecture analysis & design language (AADL): An introduction [R/OL]. 2006. CMU/SEI-2006-TN-011.

[3] SAE. AS5506/1 [S/OL]. [2006-06-16]. <http://www.sae.org/technical/standards/AS5506/1>.

[4] 王瀚博,周兴社,董云卫,等. 结构分析和设计语言 AADL 研究[J]. 计算机工程与应用,2009,45(16):1-4.

[5] 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.

[6] Senn E,Laurent J,Juin E,et al. Refining power consumption estimations in the component based AADL design flow[C]// Proc of the IEEE conf on specification,verification and design language. [s. l.]:[s. n.],2008:173-178.

[7] Jensen K. An introduction to the theoretical aspects of colored Petri nets[J]. Lecture notes in computer science,1994,803:230-272.

[8] Girault C,Valk R. Petri nets for systems engineering[M]. [s. l.]:Springer-Verlag,2003.

[9] 袁崇义. Petri 网原理与应用[M]. 北京:电子工业出版社,2005.

[10] 苗德成,奚建清,苏锦钿. AADL 进程子集行为语义研究[J]. 计算机工程与科学,2012,34(7):93-98.

[11] Singhoff F,Legrand J,Nana L,et al. Scheduling and memory requirements analysis with AADL[J/OL]. 2005. [http://delivery.acm.org/10.1145/1110000/1103847/p1-singhoff.pdf? ip=61.185.190.204&acc=ACTIVE %20SERVICE&CFID=128370857&CFTOKEN=71185284&_acm_=1342606365_9b3607d5db6a6bf2b746ace32b08415c](http://delivery.acm.org/10.1145/1110000/1103847/p1-singhoff.pdf?ip=61.185.190.204&acc=ACTIVE%20SERVICE&CFID=128370857&CFTOKEN=71185284&_acm_=1342606365_9b3607d5db6a6bf2b746ace32b08415c).

[12] 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.

[13] 余晃晶,李仁发,黄丽达. 基于 AADL 的汽车防滑控制系统可调度性分析[J]. 湖南大学学报(自然科学版),2012,39(3):43-47.

基于时间Petri网的AADL模型

作者：[吴育春](#)，[李蜀瑜](#)，[WU Yu-chun](#)，[LI Shu-yu](#)

作者单位：[陕西师范大学 计算机科学学院, 陕西 西安, 710062](#)

刊名：[计算机技术与发展](#)

ISTIC

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2014(2)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201402022.aspx