

基于 UPPAAL 的 AADL 行为模型验证方法研究

李振松 顾 斌

(北京控制工程研究所 北京 100190)

摘 要 为了实现 AADL(体系结构分析与设计语言)行为模型的分析验证,基于行为附件的文法结构以及行为描述方式,提出了 AADL 行为模型与 UPPAAL 下时间自动机模型之间的模型转换规则。在转换规则的基础上,设计和实现了自动转换的原型工具。最后以航天器控制系统中制导、导航与控制计算机从陀螺取数的 AADL 模型为例,经自动转换得到时间自动机模型,并在 UPPAAL 下仿真、验证其行为正确性,同时证明了模型转换的有效性。

关键词 AADL,行为模型,模型转换,UPPAAL,验证

中图法分类号 TP302 文献标识码 A

Research on Verification Method of AADL Behavior Model Based on UPPAAL

LI Zhen-song GU Bin

(Beijing Institute of Control Engineering, Beijing 100190, China)

Abstract To analyse and validate AADL(architecture analysis and design language) behavior model, a mapping regulation between AADL behavior model and timed automata model of UPPAAL was proposed based on the syntax definition of AADL behavior annex and the descriptive way of behavior. On the basis of the transformation rules, a prototype tool for automatic conversion was designed and implemented. Finally an AADL model of guidance navigation and control computer getting data from gyroscope in spacecraft control system was translated into a timed automata model using the tool, to simulate and verify its behavior using UPPAAL. The experiment demonstrates the validity of the model transformation.

Keywords AADL, Behavior model, Model transformation, UPPAAL, Verify

AADL^[1]语言是 2004 年由美国汽车工程师协会 SAE (Society of Automotive Engineers)体系结构描述语言附属委员会、嵌入式计算系统委员会、航空电子系统公司共同提出的一种应用于嵌入式系统领域的体系结构建模语言,是用于设计和分析安全关键的嵌入式实时系统的软、硬件体系结构的建模语言^[2],并发布为 SAE AS5506 标准。

行为附件^[3](behavior annex)是对 AADL 标准语言的语义扩展,用于描述构件内部的具体行为。行为附件通过状态及状态的迁移来描述构件行为,状态的迁移可以定义使能条件和执行动作,并且可以通过层次、并发状态来支持更加复杂的行为描述。行为附件的出现使 AADL 可更好地支持功能行为的验证和自动代码的生成。

模型转换是模型驱动开发方法(MDD)的核心,也是 AADL 模型分析和验证的基础。现阶段对 AADL 行为模型的研究主要涉及 3 个方面:其形式语义的研究^[4]、应用行为附件建模^[5]和行为模型的转换及验证。在行为模型的转换及验证方面,文献^[6]提出将 AADL 模型转换到 BIP(Behavior Interaction Priority)语言,但相应工具只能进行死锁方面的检测。文献^[4,7]将 AADL 的同步执行模型和行为模型的语义转换到时间抽象状态机(TASM)进行形式化语义描述,并进行时间属性的验证,相应自动转换工具还在开发中。

综合考虑现有工具应用的成熟度和实现可行性,本文选

择将 AADL 行为模型通过映射规则直接转换到时间自动机模型,利用世界先进水平的实时系统模拟和验证工具 UPPAAL 对 AADL 行为模型进行仿真验证,探索 AADL 模型与 UPPAAL 验证工具衔接的新途径。

1 AADL 行为模型与 UPPAAL

1.1 AADL 行为模型

AADL 行为模型是对 AADL 标准语言所构建模型的进一步精化,是以 AADL 的行为附件为基础,描述 AADL 标准构件内部详细行为的模型。AADL 的行为附件是基于自动状态机的转换理论提出的。行为附件的文法定义^[3]包含 6 个可选块,如图 1 所示。

```
annex behavior_specification{ * *  
<state variables>?  
<initialization>?  
<states>?  
<transitions>?  
<connections>?  
<composite_declaration> *  
* * };
```

图 1 行为附件的文法定义

其中,<state variables>和<initialization>分别表示类型变

到稿日期:2011-03-12 返修日期:2011-05-27 本文受国家自然科学基金重大研究计划项目(90818024)资助。

李振松(1983—),男,硕士生,主要研究方向为嵌入式软件体系结构设计,E-mail:zhensong_2005@yahoo.com.cn;顾斌(1968—),男,研究员,主要研究方向为软件工程、嵌入式软件设计。

量声明及相应变量的初始化部分, $\langle \text{states} \rangle$ 是对构件内部行为变化的状态声明, 状态的类型可以是初始(initial)状态、完成(complete)状态、返回(return)状态、紧迫(urgent)状态或者组合(composite)状态。 $\langle \text{transitions} \rangle$ 是对状态之间迁移过程的描述。构件内行为状态的迁移过程的描述格式如下:

$\langle \text{state} \rangle - [\langle \text{guard} \rangle ?] - \> \langle \text{state} \rangle [\langle \text{action} \rangle *]$

在行为附件中, 状态迁移的发生可以定义使能条件 $\langle \text{guard} \rangle$ 和执行的动作 $\langle \text{action} \rangle$, 从而通过状态迁移过程的描述来刻画具体的动作行为和约束条件。 $\langle \text{connections} \rangle$ 部分通过连接关系的设定来描述构件运行时数据、事件的传输路径。 $\langle \text{composite_declaration} \rangle$ 部分是对组合状态的详细阐述, 通过子状态机的层次化及并发状态的描述来支持构件内更加复杂的行为描述。

1.2 UPPAAL 概述

UPPAAL^[8]是由瑞典 Uppsala 大学和丹麦 Aalborg 大学联合开发的, 主要通过快速搜索机制来分析和验证状态的可达性和时钟的约束性, 具有高效性和使用方便的优点, 能够对较复杂的系统进行验证。UPPAAL 工具的图形界面由 3 部分组成: 编辑器、仿真器和验证器。编辑器用于以图形化的方式创建和编辑要分析的系统; 仿真器可以单步运行或者随机运行所建系统模型, 判断执行是否有错; 验证器通过需求规范语法 CTL 语言(分支时态逻辑)对系统进行验证, 检查时钟约束和状态可达是否满足要求。

UPPAAL 的分析模型是一种扩展的时间自动机, 其可以声明一般值变量、全局时钟、用于同步的通道以及一些执行函数。时间自动机通过使用有限的真值时钟变量提供了一个简单而又全面的方法来表示有时间约束的状态转换图。时间自动机的操作语义通过联合一个转换系统 S_A 来定义。 S_A 的一个状态是一个二元组 (s, v) , s 是 A 的一个位置, v 是满足该位置时钟约束 $I(s)$ 的一个时钟解释。转换系统 S_A 的转换语义为:

① 状态可以因为时间流逝而改变。 对一个状态 (s, v) 和一个实数时间增量 $\delta \geq 0$, 如果对于所有 $0 \leq \delta' \leq \delta$, $v + \delta'$ 满足 $I(s)$, 那么 $(s, v) \xrightarrow{\delta} (s', v + \delta')$ 。

② 状态可以因为一个位置转换而改变。 对于一个状态 (s, v) 和一个转换 $\langle s, a, \varphi, \lambda, s' \rangle$, 如果 v 满足 φ , 那么 $(s, v) \xrightarrow{a} (s', v[\lambda := 0])$ 。

概括来说, 时间自动机的状态迁移形式为 $\langle s, v \rangle \xrightarrow{\varphi, a, r} \langle s', v' \rangle$, 其中 φ 为迁移使能条件, a 为触发迁移的动作, r 为迁移发生后重置的时钟变量集合。 φ, a, r 必要时都可以省略。

2 AADL 行为模型验证

2.1 AADL 行为模型转换规则

AADL 模型转换主要是建立 AADL 模型与目标模型元素的转换规则。本文从 AADL 的行为模型出发, 建立其到时间自动机模型的转换规则。AADL 行为模型用于描述构件内部的详细行为模式, 与其他构件通过端口连接等方式形成的流(包括数据流、事件流)进行交互。AADL 模型中每个构件内的行为模型相对独立, 映射到 UPPAAL 的时间自动机模板, 多个构件交互的行为模型映射后成为 UPPAAL 下的时间自动机网络, 避免了构造时间自动机的积的繁杂过程。

从 AADL 行为附件的状态迁移描述和时间自动机的状态迁移过程的语义对照来看, 行为附件状态迁移的使能条件 $\langle \text{guard} \rangle$ 和迁移过程的动作 $\langle \text{action} \rangle$ 分别对应于时间自动机的迁移使能条件 φ 和迁移过程发生的动作 a 。简化起见, 对每个时间自动机模板仅设定一个局部时钟变量, 因此状态迁移中重置的时钟变量集合 r 仅可能包含本模板定义的局部时钟变量, 或者为空。

根据行为附件的文法定义, 行为模型在 $\langle \text{state variables} \rangle$ 块中定义变量, 在 $\langle \text{initialization} \rangle$ 块中对变量进行初始化。根据 AADL 的标准语义, 行为附件定义了 3 种基本的数据类型: 整型(integer)、实型(float)和布尔型(boolean)。用户也可以根据需要用 AADL 标准语言定义自己需要的数据类型, 如包含多个基本数据类型的结构型数据类型。UPPAAL 中定义的一般值变量类型包含整型(int)、布尔型(bool)、数组类型(array)和结构型数据(struct)。在建立映射时, 行为模型中整型、布尔型以及包含基本数据类型的结构数据可以直接转换到 UPPAAL 下变量声明中的整型、布尔型及结构型数据, 并根据行为模型 $\langle \text{initialization} \rangle$ 块的声明进行变量的初始化。由于 UPPAAL 下没有对应的实型变量类型, 因此现阶段对行为模型的构建暂且避免实型变量的声明, 从而忽略其映射关系的建立。

UPPAAL 下以位置(location)来表述状态的存在, 状态的迁移等同于位置的迁移。因此, 本文将行为模型 $\langle \text{states} \rangle$ 块中声明的自动机状态映射到 UPPAAL 中的位置, $\langle \text{transitions} \rangle$ 块中状态的迁移映射到 UPPAAL 下的位置迁移。行为模型中, 初始状态作为构件调度、执行的最初状态, 与 UPPAAL 下模板的初始位置对应, 完成状态和返回状态表示构件内行为变化的结束。为了使模型转换后的时间自动机模板成为一闭环的迁移过程, 同时保持与行为模型所描述的语义一致, 将完成状态和返回状态映射为时间自动机模板的坚定位置, 同时增加从该坚定位置到初始位置的迁移。坚定位置的语义是该位置没有时间延迟, 且下一个迁移必须离开该位置。这样就完整地表述了当构件到达完成/返回状态时, 预示着已经回到了初始状态, 并等待下一次执行。行为模型的紧迫状态表示该状态下没有时间延迟, 与时间自动机模板的紧迫位置语义一致。行为模型的组合状态是用于表示层次化的状态机而设定的, 由于在 UPPAAL 下无法表示层次化的状态机, 因此只能将组合状态(composite state)进行展平, 将行为模型中子状态机的迁移过程进行展平后映射到新的时间自动机模板。

行为模型中状态迁移过程的使能条件 $\langle \text{guard} \rangle$ 根据其表述方式分为 3 类: On 条件、When 条件和端口事件。执行动作 $\langle \text{action} \rangle$ 分为一般动作、端口动作和延时动作 3 类。使能条件表示迁移发生的前提条件, 其中 On 条件和 When 条件均表示当条件为布尔真值时迁移发生, 与 UPPAAL 下位置迁移的使能条件(guard)语义一致。而端口事件表示当构件某端口接收到输入事件时, 状态迁移发生, 与输出该事件的构件进行行为交互, 因此将行为模型状态迁移使能条件中的端口事件映射为 UPPAAL 下对应时间自动机模板交互的同步通道。执行动作 $\langle \text{action} \rangle$ 中描述的端口动作表示构件通过端口输出事件或者变量值, 同理也映射为同步通道。一般动作包

含变量的赋值和子程序调用,这里将一般动作映射到 UP-PAAL 下位置迁移的更新(update)。行为模型中的延时动作包含 delay 和 computation 两种,格式分别为 delay(min, max)、computation(min, max),表示动作延迟时间在 min 和 max 之间。将 $clock \geq min$ 作为时间自动机迁移动作的使能条件之一,而 $clock \leq max$ 作为迁移原位置的时钟约束,其中 clock 指代模板局部时钟变量,如图 2 所示。

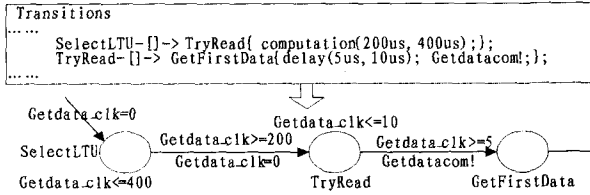


图 2 状态迁移延时动作到时间自动机的转换实例

从图 2 可以看出,computation(200us,400us)转换到时间自动机模板后,将 400us 作为 SelectLTU 位置的时钟约束,以 Getdata_clk ≤ 400 表示,其中 Getdata_clk 为该时间自动机模板的局部时钟变量,将 200us 作为 SelectLTU 位置到 TryRead 位置迁移的使能条件,即 Getdata_clk ≥ 200。

行为模型中 <connections> 块描述构件之间连接、交互的关系,这里先将每个连接所关联的两个构件行为映射到时间自动机模板,再将该连接映射为其模板间交互的同步通道。<composite_declaration> 块是对所声明的组合状态的详细描述,一般是进行子状态机的展平化操作后,映射为新的时间自动机模板。但是当存在多个子状态机并发运行时,为实现模型转换后的并发操作,将组合状态映射为触发每个子状态机模板运行的广播通道(以连续的坚定位置迁移来实现一个模板同时与其它多个模板交互的方法)形式,来保持其并发执行的语义,如图 3 所示。

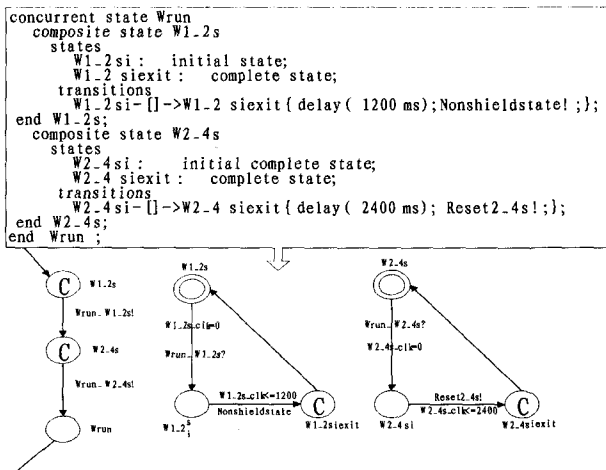


图 3 并发组合状态到时间自动机的转换实例

2.2 AADL 行为模型验证

2.2.1 模型转换工具的实现

根据 2.1 节所建立的模型转换的映射规则,基于 AADL 的行为模型开发了模型转换的原型工具 AADLBA2UPPAAL,以实现 AADL 行为模型到 UPPAAL 时间自动机模型的自动转换,进而采用 UPPAAL 工具对模型进行形式化的行为验证,如图 4 所示。

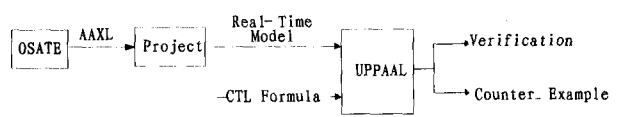


图 4 UPPAAL 工具验证 AADL 行为模型的过程

OSATE 环境下,AADL 模型以 AAXL 格式存储。AAXL 是一种自有格式类 XML 文件,通过转换规则将 AAXL 文件转换到 UPPAAL 下的 XML 格式文件,实现模型转换的工具化。

原型工具 AADLBA2UPPAAL 的运行需要输入两个参数,分别是输入的 AAXL 格式的 AADL 模型文件名和生成的 XML 格式的时间自动机模型文件名。运行命令为:

```
java -jar AADLBA2UPPAAL.jar <aaxl input file> <upaal output file>
```

即在相应目录下生成模型转换后的 XML 文件,可用 UPPAAL 工具直接打开,并进行分析验证。

2.2.2 行为模型的转换与验证实例

针对航天器控制系统中制导、导航与控制计算机(GNCC)通过串口从陀螺取数的过程采用 AADL 进行建模,主要是构建其构件内的行为模型,如图 5 所示。系统构件 GNCCGetGyrodatasys 内描述了 GNCC 从陀螺取数的行为,系统构件 LTUGyrosys 内部描述了陀螺发送数据给 GNCC 的行为,系统构件 RS422BASys 将串口的行为进行抽象描述。同时为了构成一闭环系统,模型设计时增加了 SubGNCAct 和 SubLTUover 两个构件,以便转换到时间自动机进行验证,避免存在死锁。

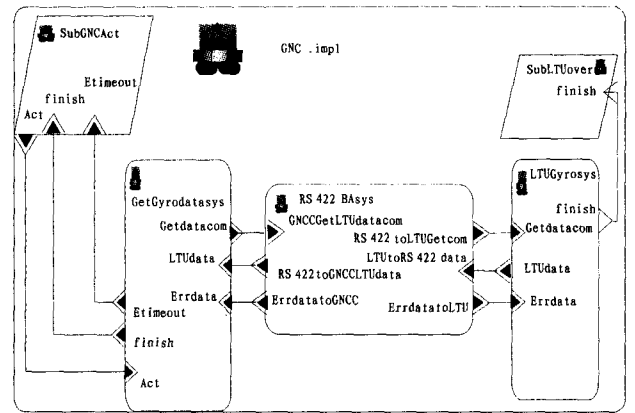


图 5 GNCC 从陀螺取数过程的 AADL 模型

执行模型转换后生成的 XML 文件在 UPPAAL 中打开,在编辑器中可以看到生成的 5 个时间自动机模板,表明模型的转换正常执行。通过在仿真器中单步和随机运行,可以看到 GNCC 与陀螺之间能够进行正常的数据发送和接收。

同时,UPPAAL 提供了一种对模型进行验证的 BNF 语法:

$$\varphi ::= A[\Box\beta | E\langle\gamma\rangle\beta | E[\Box\beta]A\langle\gamma\rangle\beta] \beta \rightarrow \gamma$$

式中, β, γ 为描述系统性质的逻辑表达式;A 表示对于所有的路径均满足所给性质 β ;E 表示至少有一条路径满足该性质 β ;符号 $\Box$ 表示对于路径上的所有状态均满足所给性质 β ; $\langle\gamma\rangle$ 表示路径上至少有一个状态满足给定所给性质 β 。

对转换后的模型进行验证,如图 6 所示。通过验证,表明所建 AADL 行为模型符合设计需求。

(下转第 169 页)

- [24] Martin D, Burstein M, Hobbs J, et al. OWL-S: Semantic Markup for Web Services[EB/OL]. <http://www.w3.org/submission/2004/SUBM-OWL-S-2041122/>
- [25] Akkiraju R, Farrell J, Miller J, et al. Web Service Semantics-WSDL-S[R]. W3C Member Submission 7, November 2005
- [26] Ipate F, et al. Testing(Stream) X-machines. Applicable Algebra in Engineering, Communication and Computing[M]. Springer-Verlag, 2003:217-237
- [27] Ma Chun-yan, Wu Jun-sheng, Zhang Tao. Web Service Sequence Testing Based on Stream X-machine[C]//The 10th International Conference on Quality Software(QSIC 2010). 2010, 7:232-239
- [28] Ipate F, Holcombe M. Testing data processing-oriented systems from stream X-machine models[J]. Theoretical Computer Science, 2008, 403:176-191
- [29] Ipate F. Testing against a non-controllable stream X-machine using state counting[J]. Theoretical Computer Science, 2006, 353:291-316
- [30] Alves A, et al. Web Services Business Process Execution Lan-

- guage Version 2.0, OASIS Standard[S]. April 2007. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- [31] Kavantzias N, Burdett D, Rizinger G, et al. Web Service Choreography Description Language Version 1.0[Z]. <http://www.w3.org/TR/ws-cdl-10/>, 2005-11
- [32] Ma Chun-yan, Wu Jun-sheng, Zhang Tao, et al. Testing BPEL with Stream X-Machine[C]//2008 International Symposium on Information Science and Engineering(ISISE). 2008:578-582
- [33] Open Service Access(OSA). Parlay X Web Services; Part 12: Multimedia Conference [OL]. <http://www.parlay.org/en/specifications/pxws.asp>
- [34] <http://api.google.com/GoogleSearch.wsdl>
- [35] <http://seeker.dice.com/jobsearch/results/US/Detroit-Metro/Communication/w3r+Consulting/WSDL>
- [36] Ma Chun-yan, Wu Jun-sheng, Zhang Tao, et al. Automatic Test Case Generation for BPEL Using Stream X-machine[J]. International Journal of u- and e-Service, Science and Technology, 2008, 1(1):27-35

(上接第 161 页)

A[] not deadlock; 通过验证表明系统能够正常运行,不存在死锁;
 E(>)GNC_impl_GetGyrodatasys. GNC_impl_GetGyrodatasys_Getdataover; 通过验证表明系统能够正常传输数据;
 E(>)GNC_impl_GetGyrodatasys. GNC_impl_GetGyrodatasys_Timeoutstate; 通过验证表明数据传输中可能存在超时;
 A(>)GNC_impl_GetGyrodatasys. GNC_impl_GetGyrodatasys_Timeoutstate imply GNC_impl_GetGyrodatasys. GNC_impl_GetGyrodatasys_idle;
 通过验证表明在传输数据过程中如果出现超时,系统总是能够复位;
 E(>)GNC_impl_GetGyrodatasys. GNC_impl_GetGyrodatasys_pre_Errorstate; 通过验证表明数据传输中可能存在出现错误的情况;
 A(>)GNC_impl_GetGyrodatasys. GNC_impl_GetGyrodatasys_pre_Errorstate imply GNC_impl_GetGyrodatasys. GNC_impl_GetGyrodatasys_idle;
 通过验证表明在传输数据过程中如果出现数据错误,系统总是能够复位。

图 6 UPPAAL 下 GNCC 对陀螺取数行为的验证

2.2.3 局限性分析

我们通过建立 AADL 行为模型与 UPPAAL 下时间自动机模板之间的映射关系,实现两种模型之间的自动转换;采用形式化的方法验证构件行为,增加了验证 AADL 模型的途径。但是由于两种模型之间存在一些差异,模型转换中的一些语义、性质可能丢失,如实型变量无法在 UPPAAL 下表示等,因此现阶段在构建 AADL 行为模型时,需要避免使用实型变量等情况,从而实现模型转换中语义、性质的保持。原型工具 AADLBA2UPPAAL 在对(connections)块、子程序调用及层次状态机的转换方面还有待实现,且只能对单一模型文件进行转换,需进一步完善。

结束语 本文通过分析 AADL 行为附件文法的语义,建立了 AADL 行为模型与 UPPAAL 时间自动机模板之间的转换映射关系,实现了模型转换的原型工具。根据某型号航天器的需求,构建了 GNCC 对陀螺取数的 AADL 行为模型,通过工具自动转换为时间自动机模型,并在 UPPAAL 下对其

取数过程的协议设计进行了验证。通过验证表明,数据在 GNCC 与陀螺之间能够正常地发送和接收;在遇到数据丢失及校验错误的情况下也能够恢复到初始状态,继续进行下一次数据传输。

AADL 行为附件的提出增强了 AADL 语言对构件内部行为的描述能力。本文利用形式化的方法对 AADL 行为模型进行行为验证,可信度较高。但是模型转换过程中语义、性质的一致性保持及证明还有待进一步研究。

参 考 文 献

- [1] SAE Aerospace. SAE AS5506A; Architecture Analysis and Design Language V2.0[EB/OL]. <http://www.sae.org/technical/standards/AS5506A>, 2009
- [2] 刘倩,桂盛霖,李允,等. 基于 UPPAAL 的 AADL 模型可调度性验证[J]. 计算机应用, 2009, 29(7):1820-1824
- [3] SAE Aerospace. SAE AS5506 Annex; Behavior Specification V1.6 [EB/OL]. http://www.aadl.info/aadl/documents/Behaviour_Annex1.6.pdf, 2006
- [4] Yang Z B, Hu K, Ma D F, et al. Towards a formal semantics for the AADL behavior annex[C]//Proceedings of the IEEE/ACM Design, Automation and Test in Europe 2009. Nice; EDAA, 2009:1166-1171
- [5] França R B, Bodeveix J P, Filali M, et al. The AADL behaviour annex-experiments and roadmap[C]//Proceedings of the 12th IEEE Int'l Conf. on Engineering Complex Computer Systems. Washington; IEEE Computer Society, 2007:377-382
- [6] Chkouri M Y, Robert A, Bozga M, et al. Translating AADL into BIP-Application to the verification of real-time systems[C]//Proceedings of the ACESMB 2008 Workshop Conjunction with MODELS 2008. Berlin, Heidelberg; Springer-Verlag, 2009:5-19
- [7] 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
- [8] Behrmann G, David A, Larsen K G. A tutorial on UPPAAL[C]//Proceedings of the 4th Int'l School on Formal Methods for the Design of Computer, Communication, and Software Systems. Bertinoro; Springer-Verlag, 2004:200-236