

基于 AADL 的数据流转换与验证

孙 健,徐 敏

(南京航空航天大学 计算机科学与技术学院,江苏 南京 210016)

摘 要: AADL 在嵌入式实时系统领域,支持系统软、硬件结构建模的同时又能对可靠性、实时性等非功能属性进行描述,可以在模型驱动开发过程中的早期模型建立阶段,通过形式化的模型检验方法对系统模型的关键属性进行验证,从而能够及早地发现在设计过程中存在的潜在错误,对保证系统实时性和提高开发效率来说都具有十分重要的意义。针对数据流时延特性问题,文中提出将 AADL 数据流的分析形成数据流的形式化描述的方法,建立这种形式化描述到时间自动机语义的映射关系作为映射法则的定义,并将时间自动机的转换按单一和混合两种类型分别给出了转换法则和转换实例的说明。在混合数据流转换中,新建了非周期线程的模板,以支持数据流的综合分析。最后给出了数据流性质验证的参考查询语句,并对数据流转换到的时间自动机模型进行了必要的实验检验。

关键词: AADL; 数据流时延; 形式化描述; 时间自动机; 性质验证

中图分类号: TP302

文献标识码: A

文章编号: 1673-629X(2016)04-0041-05

doi:10.3969/j.issn.1673-629X.2016.04.009

Transformation and Verification of Data Flows Based on AADL

SUN Jian, XU Min

(School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China)

Abstract: AADL supports structural modeling for software and hardware, and imports non-functional attributes description such as real-time and reliability in embedded real-time system. During the process of MDD (Model-Driven Development), it is of great significance for ensuring the system real-time performance and improving the efficiency of system development to find potential design problems on critical aspects in the model design stage. In order to analyze the flow latency of AADL models, a method is proposed taking the analysis of data flows of AADL to form a formal description of data flows. The mapping relationship from formal description to time automaton semantics is regarded as the definition of mapping rules. On building the timed automata of data flows, methods and samples are given to transform both simple and mixed flows into timed automata. In the transformation of mixed flows, a template of non-periodic thread is presented to support the comprehensive analysis of data flows. At last the reference query statements is given to verify the properties of the data flows, and the necessary experimental tests of time automaton model converted from data flows are carried out.

Key words: AADL; data flow latency; formal description; time automata; property verification

1 概 述

在 AADL 模型中,数据流是数据和事件信息传递的通道。一条完整的端到端的数据流通常由一个采集器设备发出,传递给一个中间组件(例如线程组件)处理后,将处理结果传递给控制器做出响应。在这个过程中,信号从采集到控制具有一定的时效性,若数据流的延迟过大,将会导致关键的数据不能及时送达或关键的任务不能按时触发,从而影响整个系统的实时性^[1-2]。

在 AADL 数据流时延特性的研究方面, AADL 标准的制定者之一 Peter Feiler 在他的文章^[3]中提出了影响数据流时延的因素及其分析方法,他指出影响数据流时延的因素主要有四点:

- (1) 线程或设备的计算;
- (2) 在不同组件之间传输的时延;
- (3) 数据采样速率和设备端口上数据队列的处理方式;
- (4) 传输协议对于数据等待队列的处理方式。

收稿日期:2015-07-19

修回日期:2015-10-20

网络出版时间:2016-03-22

基金项目:国家“973”重点基础研究发展计划项目(2014CB744900)

作者简介:孙 健(1990-),男,硕士研究生,研究方向为人工智能;徐 敏,副教授,研究方向为人工智能、软件工程。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160322.1522.084.html>

尽管时延的影响因素得到了全面分析,但 Peter Feiler 也只是选取了 AADL 属性集合中的 Latency 属性作为其分析的依据,同时在其开发的 OSATE 插件中,利用该属性对模型的数据流做了简单的模拟,并将模拟计算得出的时延结果与预期时延进行比较。另外,在 Lee Su-Young 等^[2]的研究中,结合了周期和非周期线程对于数据流的影响,分析给出了线程的数据流的时延与线程的计算时间、截止时间等的关系,得出了最优时延与最差时延的计算公式。但是这种计算没有考虑到实际的系统调度对数据流中数据流向的影响,是很不精确的。谯婷婷等^[4]基于一个飞行控制系统的实例给出了流分析的结果,但没有给出一般的方法或通用的解决方案。

通过以上的分析,文中将数据流时延特性作为研究的目标,构建数据流到时间自动机模型之间的转换,对于流经计算线程的数据流,可以结合调度模型的时间自动机来进行综合分析,得到混合模式下数据流的时延及其模型的可调度性验证方法。通过以上方法来达成下列目的:

- (1) 检查数据流的完整性,即从源端口到目的端口能否形成完整的逻辑通路;
- (2) 数据流中各段所累加的时延能否满足端到端数据流时延的设计要求;
- (3) 数据流流经计算线程后,时延的要求能否满足设计要求;
- (4) 数据流流向非周期线程的 Dispatch 端口时,在通过事件触发线程的派发的情况下系统能否满足可调度性设计要求。

2 AADL 数据流

2.1 AADL 简介

AADL 可以详细描述嵌入式系统性能相关的属性,如可靠性、有效性、时间性、响应性、吞吐量、安全性^[1]。这些属性使系统的设计者能够完成对组件和系统的分析,例如系统的可调度性分析、粒度分析、可信性分析。

从这些分析中,设计者可以测评体系结构的平衡和改变^[5]。AADL 满足了安全关键嵌入式实时系统的相关特殊需要。

AADL 的主要组件分为软件组件和执行平台组件两大类。软件组件主要包括数据、线程、进程、子程序等;执行平台组件主要包括处理器、存储器、总线、外设等。

作为 AADL 的执行实体,线程可分为周期线程、非周期线程和零星线程三类。它们的分派策略及其对外部事件的响应各不相同。

2.2 数据流简介

AADL 模型的组件是通过流(flow)来进行连接的,端到端的流(end to end flow)描述了系统内部数据和事件的抽象信息路径。与 AADL 中的组件一样,完整的 flow 定义包含两个部分:流声明(flow specification)和流实现(flow implementation)^[5-6]。流声明在组件声明中完成,它定义了从组件输入到输出的逻辑路径,含有三种标记类型:flow source(流的起源)、flow sink(流的终点)和 flow path(流的路径)。流实现在组件实现中完成^[7],通过串联组件与子组件、子组件与子组件之间的连接,形成了从输入端口到输出端口之间的串行序列,描述了组件中确切的数据流向。与流声明对应,流实现也有三种类型,其定义方式与流声明略有不同,每个实现都代表一条路径。端到端的数据流(end to end flow)作为流实现中的特例,实现了跨组件之间数据流的连接,它的起点是 flow source,终点是 flow sink,中间可以串联多个组件之间的 flow implementation 和 connections。

在 AADL 的属性集合中,定义了与数据流时延直接相关的属性 Latency,用来规定在流或者连接上所允许的时间延迟,描述的对象包括数据流(flows)、连接(connections)等。

2.3 数据流的形式化描述

一个端到端的数据流 F_p 的定义可描述为一个六元组, $F_p = \{F, f_s, f_d, P_o, C_o, L_a\}$, 各元素说明如下:

F : 流声明用作组件内部的数据流表述 $F = \{P_o \times P_o, L_a\}$, 在流声明中 $F = \{P_o \times F \times P_o, L_a\}$ 。从这个意义上, F_p 还可以表述为 $F_p = \{F \times F, f_s, f_d\}$ 。

f_s : $f_s \in F$, 即 flow source, 是端到端数据流的起点。

f_d : $f_d \in F$, 即 flow sink, 是端到端数据流的终点。

P_o : 端口组件的集合, 其中 $P_o = \{\text{data port, event port, event port}\}$ 。

C_o : $C_o = \{P_o \times P_o, \varphi\}$, 是不同组件端口连接的集合, φ 为连接约束的条件, 禁止 event port 指向 data port 或者 event port, 而其他指向的连接都为合法的。

L_a : Latency 属性的集合, 作用于 F 和 C_o 。

2.4 时间自动机理论

时间自动机是在传统的有限状态自动机的基础上扩充了时钟、时钟约束和不变条件而得到^[8]。系统当中定义的时钟在一定取值范围内按照相同的速率同步连续的增长,也可以在任意的时刻被复位为初始值 0。系统的状态转移约束条件由时钟和整型变量构成,只有在时钟满足一定的条件下,状态的转移才能够发生,而对于不变条件是对状态停留在一个位置上的约束,

满足不变条件的情况下,时钟才会在一个位置停留并持续增长^[9-13]。时间自动机的系统由于引入了时钟的概念后,对系统的时间概念具有了很强的表达能力,成为了描述实时系统模型的一个很好的工具。

作为一种形式化的方法,时间自动机有其严格的定义,以下做以说明^[10]:

定义1:状态转移系统。

一个状态转移系统是一个四元组 $S = \langle L, I_0, A, E \rangle$ 。其中: L 是自动机状态的集合; $I_0(I_0 \in L)$ 是一个初始的状态; A 是触发转移的事件集合; $E \subseteq L \times A \times L$ 是所有状态转移集合。

系统中的一个状态转移 $l \xrightarrow{a} l'$ 或者 $\langle l, a, l' \rangle$,表示在一个激励事件 a 的作用下,发生了从 l 到 l' 的状态转移。如果这样的转移是可发生的,则说明状态 l' 是从 l 状态可达的,记为 $l \longrightarrow * l'$ 。对于整个系统而言,从初始状态 I_0 开始,经过有限次的状态转移,所有可到达状态集合构成了状态转移系统的状态空间。

定义2:时钟约束。

对于一个时钟变量集 C ,时钟约束 φ 的集合 $\varphi(C)$ 的定义如下:

$$\varphi := x \triangleright \langle n \mid \varphi_1 \wedge \varphi_2$$

其中: x 是一个时钟变量; n 是自然数集 N 中的一个常量。

定义3:时钟解释。

一个时钟解释是时钟集合 C 到自然数集的映射, $v: C \rightarrow N$ 。

对于一个 $\delta \in R$, R 表示的时钟解释是对于时钟变量集合 C 中的每一个时钟变量 x 赋值为 $v(x) + \delta$; 对于一个 $\delta \in R$, $\delta \cdot v$ 表示的时钟解释是对 C 中的每个时钟变量 x 赋值为 $\delta \cdot v(x)$; 对于 $X \subseteq C$, $v[X := 0]$ 表示对于满足 $x \in X$ 的时钟 x 复位为0,其余时钟保持增长。

定义4:时间自动机。

时间自动机是在有限状态自动机上扩充了时钟变量形成的,时钟由整型变量代表,且所有的时钟变量在自动机中是同步递增的^[12-14]。一个时间自动机可表示为一个六元组, $TA = \langle L, I_0, C, Var, E, I \rangle$ 。其中:

L :有穷的位置(location)集合。

$I_0: I_0 \in L$,表示自动机的初始位置,即初始状态。

C :时钟的集合(Clock),时钟默认从0开始,不断自增加1,可以在任意时刻被重新赋值。

Var :一系列的变量集合。

E :边(Edge)的集合, $E \subseteq L \times G \times Act \times U \times L$ 。

其中: G 代表约束条件的集合,只有满足约束条件,转移才会发生; $Act = I \cup O$,构成了触发转移的使能条

件; U 是对时钟变量或者整型变量的更新操作。

I :不变条件(invariant),是状态转移的约束函数的集合。

3 数据流到时间自动机的转换

3.1 单一数据流转换方法

单一数据流指的是所有数据端口都在同一层级的组件之间,数据不流向其子组件,且不受线程计算及系统任务调度的影响,设备对数据及其事件信号的发送和接收是即时的,数据流时延的产生只考虑流属性中所定义的 Latency。例如,图1为汽车防滑控制系统部分 AADL 模型^[14],Sensor 表示一个轮速传感器,它实时采集汽车车轮的转动速度,通过 Speed_Out 数据端口发送给处理器 Processor,处理器通过 Speed_In 端口接收到数据,计算后通过 SpeedControl_Out 端口将控制信息发送给速度控制器 SpeedActuator 的接收端口 SpeedControl_In,对速度做出实时调整。在这个控制系统中,可以看出一个端到端的数据流包含了四个数据端口,两个组件之间端口的连接以及一个组件内部的数据流,所连接的部分均处在同一个组件层级。该数据流到时间自动机的转换规则阐述如下。

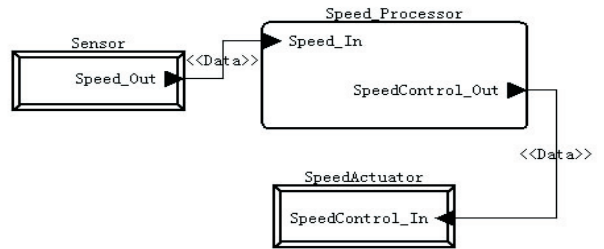


图1 汽车防滑控制系统数据流模型图

定义转换规则 Γ_{FP} , $\Gamma_{FP}(FP) = TA$, 即 $\{F, f_s, f_a, P_o, C_o, L_a\} \rightarrow \langle L, I_0, C, Var, E, I \rangle$, 系统单个端到端的数据流映射到单一时间自动机,而数据流的集合映射到时间自动机集合,具体的映射关系说明如下:

(1) $P_o \rightarrow L$,端口映射为基本状态,根据数据的流向通过自动机中的边串联起来,加入一个 Start 初始化状态来表示流的开始,并在最后加入中止状态 End 作为可达性验证的参考状态。

(2) $L_a \rightarrow \langle C, Var \rangle$,时延对应于时钟变量和整型变量,一个端到端的流中设置了几个分段的时延,就会设置几个时钟变量,同时设置相应的整型变量来记录约定的约束的值。

(3) $\langle C_o, L_a \rangle \rightarrow \langle E, I \rangle$,每个 connection 的时延转换为三部分:一个时钟赋值、一个转移条件以及一个不变条件。时钟赋值在端口对应状态的进入条件中设置,将其时钟置为0。转移条件在由连接起点端口

状态转移到终点端口状态的边上进行设置,判断条件为 $clock = latency$, 表示连接上数据的传递在 Latency 设置的时间点上完成。相应的不变条件为 $clock \leq latency$, 是对转移前状态的约束。

(4) 流声明中定义的 F 代表的是组件内部的数据流, 这部分的转换方法与 connection 的转换方法 $\langle C_o, L_a \rangle \rightarrow \langle E, I \rangle$ 相同。流实现中跨组件的 flow path 的转换中, Latency 起到验证的作用, 检查 flow path 内部流的累加是否超过了设置的 Latency 值。为了能够实现这种转换, 文中设置了相应的时钟和整型变量。时钟从 flow path 的开始状态之前开始计时, 另外在 flow path 终点端口的状态之后加入了 committed 状态, 由终点端口的状态指向该状态。转移条件为所设置时钟的值小于或等于 Latency 的值, 该 committed 状态的后继状态为下一个端口状态或 end 状态。

对于图 1 所示的模型, 按照转换规则转换到的时间自动机如图 2 所示。

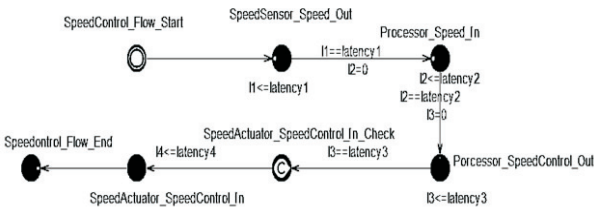


图 2 汽车防滑控制系统数据流时间自动机

3.2 混合数据流转换方法

在实际的系统中, 数据的采集、发送往往是有一定周期或是触发操作的, 数据流向线程的计算也需要一定的时间, 且这个时间与线程计算时间以及系统的调度情况都是密切相关的。将这种由采集设备产生、经线程计算或者触发非周期事件派发的数据流称为混合数据流。本节将首先分析信号源周期产生, 并且经过线程计算的数据流的时延情况, 接着在调度模型中引入由事件引起的非周期派发的线程并对系统的调度情况加以分析。

在图 1 所示的系统的基础上, 文中做了一些扩充。数据流源头的数据采集设备 Sensor 是周期工作的, 同时数据信号的发送也是周期的。此外, 数据流经了一个计算线程, 在设定的计算时间后, 由输出端口将结果返回数据流, 继续传递给控制器。在这种情况下, 数据流的时延必须要考虑到线程计算时间的因素, 而线程从触发到执行结束的时间不是一个确定的值, 它和系统所处的调度状态有关。相对于单一数据流的转换法则, 混合数据流到时间自动机的转换法则扩充为:

(1) $Po_0 \rightarrow TA$, 根据源端口 Po_0 所在设备的派发形式和派发周期来设计一个数据产生器时间自动机, 通过同步通道通知数据流时间自动机数据的产生。

(2) $F_T \rightarrow \langle Ch, Var \rangle$, 在指向线程输入端口对应状态的边上, 通过设置数据流时间自动机与非周期线程模板通信的同步通道来通知相应线程派发, 在从线程输入端口状态转移到输出端口状态的边上设置线程执行完成的接收同步信号。

根据扩充的转换法则, 混合数据流的时间自动机如图 3 所示。

对于更复杂的情况来说, 由数据流连接起来的多个线程, 线程之间具有级联的关系, 线程 1 产生的结果作为触发线程 2 派发的信号, 数据流时延和对调度模型的影响具有更大的不确定性。对于这种 AADL 模型的转换仍然遵照上述的转换规则, 线程转换成单独的时间自动机模板, 数据流中按照基本的单一数据流转换并加入同步通道和线程通信。

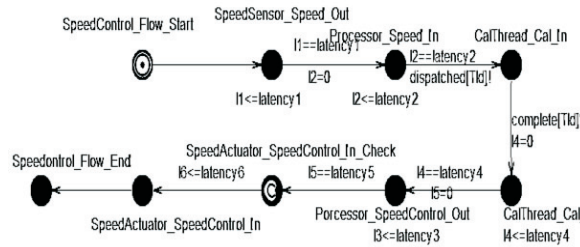


图 3 汽车防滑控制系统混合数据流时间自动机

4 数据流性质验证语句

数据流验证所关心的主要是流逻辑的正确性和设计所要求的时延特性, 通过实际的时间自动机的结束状态和添加的端口检查状态的状态可达性可以验证这两个性质。例如 $E \langle \rangle SpeedControlFlow. Speedcontrol_Flow_End$, 若性质满足则说明整个数据流在模型转换的过程中已经正确地连接起来, 能够形成逻辑上的通路, 同时也证明了在该数据流上每个点对应的时延约束是可以满足的, 单独验证某端时延特性也可用 $E \langle \rangle SpeedActuator_SpeedControl_In_Check$ 。而调度模型中相关性质的验证可以采用表 1 的验证查询语句, 数据流事件自动机和非周期线程自动机模板的引入并不对其性质验证产生影响。

5 结束语

文中建立了 AADL 数据流到时间自动机模型间的映射, 利用模型转换得到的时间自动机分析了单一数据流和混合数据流的时延性质, 并与调度模型的时间自动机结合构成时间自动机网络, 以更好地模拟 AADL 模型所描述的系统的运转情况, 从而更准确地验证可调度性和数据流时延的特性。最后给出了数据流时延性质验证查询语句的参考并设计了模型转换方法检验的实验, 证明该转换方法是可行的。

表1 模型可调度性性质验证语句

性质验证语句	验证性质说明
A[] not deadlock	检查原模型设计的系统是否能够连续运转,是否会进入死锁状态
A<> Thread0. Running	检查某线程(该语句是线程0)是否确定会进入执行状态
A[] not Thread0. Error	检查某线程(该语句是线程0)是否会进入错误状态
E<> Processor. MissedDeadline	检查调度器是否会进入超时状态
Thread0. Preempted --> Thread0. Running	检查某线程(该语句是线程0)能否由被抢占状态转移到执行状态

文中在基于AADL的数据流转换与验证中取得了一定的进展,但数据流分析考虑的因素还较少,还不支持端口队列上时延的分析,没有考虑流经周期线程的数据流。未来对数据流分析的工作将考虑更多的影响因素,从而实现更准确的分析与验证。

参考文献:

[1] 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.

[2] Lee S Y, Mallet F, de Simone R. Dealing with AADL end-to-end flow latency with UMLMARTE[C]//Proc of 13th IEEE international conference on engineering of complex computer systems. [s. l.]:IEEE,2008.

[3] Feiler P H, Hansson J. Flow latency analysis with the Architecture Analysis & Design Language (AADL)[R]. [s. l.]: [s. n.],2008.

[4] 谯婷婷,王乐,耶国栋.基于AADL的软件可靠性验证[J].计算机应用,2012,32(S2):92-95.

[5] 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.

[6] Johnsen A. Architecture-based verification of dependable embedded systems[D]. Sweden:Mälardalen University,2013.

[7] Feiler P H, Lewis B A, Vestal S. The SAE Architecture Analysis & Design Language (AADL) a standard for engineering performance critical systems [C]//Computer aided control system design,2006 IEEE international conference on control applications,2006 IEEE International symposium on intelligent control. Munich, Germany:IEEE,2006:1206-1211.

[8] Alur R. Timed automata[M]//Computer aided verification. Berlin:Springer,1999.

[9] 徐仁佐.软件可靠性工程[M].北京:清华大学出版社,2007.

[10] 童超.基于时间自动机的RBC控车流程研究[D].成都:西南交通大学,2009.

[11] 朱雪阳,唐稚松.基于时序逻辑的软件体系结构描述语言XYZ/ADL[J].软件学报,2003,14(4):713-720.

[12] 李振松,顾斌.基于UPPAAL的AADL行为模型验证方法研究[J].计算机科学,2012,39(2):159-161.

[13] 周清雷,姬莉霞,王艳梅.基于UPPAAL的实时系统模型验证[J].计算机应用,2004,24(9):129-131.

[14] 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.

[1] 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.

[2] Lee S Y, Mallet F, de Simone R. Dealing with AADL end-to-end flow latency with UMLMARTE[C]//Proc of 13th IEEE international conference on engineering of complex computer systems. [s. l.]:IEEE,2008.

[3] Feiler P H, Hansson J. Flow latency analysis with the Architecture Analysis & Design Language (AADL)[R]. [s. l.]: [s. n.],2008.

[4] 谯婷婷,王乐,耶国栋.基于AADL的软件可靠性验证[J].计算机应用,2012,32(S2):92-95.

[5] 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.

[6] Johnsen A. Architecture-based verification of dependable embedded systems[D]. Sweden:Mälardalen University,2013.

[7] Feiler P H, Lewis B A, Vestal S. The SAE Architecture Analysis & Design Language (AADL) a standard for engineering performance critical systems [C]//Computer aided control system design,2006 IEEE international conference on control applications,2006 IEEE International symposium on intelligent control. Munich, Germany:IEEE,2006:1206-1211.

[8] Alur R. Timed automata[M]//Computer aided verification. Berlin:Springer,1999.

[9] 徐仁佐.软件可靠性工程[M].北京:清华大学出版社,2007.

[10] 童超.基于时间自动机的RBC控车流程研究[D].成都:西南交通大学,2009.

[11] 朱雪阳,唐稚松.基于时序逻辑的软件体系结构描述语言XYZ/ADL[J].软件学报,2003,14(4):713-720.

[12] 李振松,顾斌.基于UPPAAL的AADL行为模型验证方法研究[J].计算机科学,2012,39(2):159-161.

[13] 周清雷,姬莉霞,王艳梅.基于UPPAAL的实时系统模型验证[J].计算机应用,2004,24(9):129-131.

[14] 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.

(上接第40页)

[1] 谯婷婷,王乐,耶国栋.基于AADL的软件可靠性验证[J].计算机应用,2012,32(S2):92-95.

[3] 汪晓庆,郑彦兴,史美林.一种有效的数据共享环境多数据源选择算法[J].软件学报,2008,19(2):314-322.

[4] 诸云强,冯敏,宋佳,等.基于SOA的地球系统科学数据共享平台架构设计与实现[J].地球信息科学,2009,11(1):1-9.

[5] 陈跃国,王京春.数据集成综述[J].计算机科学,2004,31(5):48-51.

[6] Wu D, Håkansson A. Applying a knowledge based system for metadata integration for data warehouses [M]//Knowledge-based and intelligent information and engineering systems. Berlin:Springer,2010:60-69.

[7] 张兴会.数据仓库与数据挖掘技术[M].北京:清华大学出版社,2011.

[8] 周长春,徐宏炳,张小伟.基于共享数据库的数据集成方案的改进[J].计算机工程与设计,2007,28(8):1917-1919.

[9] Chang F, Dean J, Ghemawat S, et al. Bigtable: a distributed storage system for structured data [J]. ACM Transactions on Computer Systems,2008,26(2):4-4.

[10] 刘刚,侯宾,翟周伟.Hadoop 开源云计算平台[M].北京:北京邮电大学出版社,2011.

[11] 陆嘉恒.Hadoop 实战[M].北京:机械工业出版社,2011.

[12] White T. Hadoop: the definitive guide [M]. [s. l.]:O'Reilly Media, Inc.,2012.

[13] 李光焰.数据仓库中元数据分类及管理系统研究进展分析[J].情报科学,2004,22(7):889-892.

[14] 冯少荣,肖文俊.并行分布环境下的黑板模型[J].华东理工大学学报:自然科学版,2008,34(1):96-102.

[15] 张松懋.关于黑板模型和分布式黑板模型[J].微电子学与计算机,1993,10(8):1-4.

[16] McManus J W. Design and analysis techniques for concurrent blackboard systems [J]. IEEE Transactions on System, Man and Cybernetics Part A: Systems and Humans,1996,26(6):669-680.

[17] 唐晓光.基于订阅机制的数据共享平台的研究与设计[D].大庆:东北石油大学,2013.

Architecture Analysis & Design Language (AADL)[R]. [s. l.]: [s. n.],2008.

[4] 谯婷婷,王乐,耶国栋.基于AADL的软件可靠性验证[J].计算机应用,2012,32(S2):92-95.

[5] 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.

[6] Johnsen A. Architecture-based verification of dependable embedded systems[D]. Sweden:Mälardalen University,2013.

[7] Feiler P H, Lewis B A, Vestal S. The SAE Architecture Analysis & Design Language (AADL) a standard for engineering performance critical systems [C]//Computer aided control system design,2006 IEEE international conference on control applications,2006 IEEE International symposium on intelligent control. Munich, Germany:IEEE,2006:1206-1211.

[8] Alur R. Timed automata[M]//Computer aided verification. Berlin:Springer,1999.

[9] 徐仁佐.软件可靠性工程[M].北京:清华大学出版社,2007.

[10] 童超.基于时间自动机的RBC控车流程研究[D].成都:西南交通大学,2009.

[11] 朱雪阳,唐稚松.基于时序逻辑的软件体系结构描述语言XYZ/ADL[J].软件学报,2003,14(4):713-720.

[12] 李振松,顾斌.基于UPPAAL的AADL行为模型验证方法研究[J].计算机科学,2012,39(2):159-161.

[13] 周清雷,姬莉霞,王艳梅.基于UPPAAL的实时系统模型验证[J].计算机应用,2004,24(9):129-131.

[14] 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.

[1] 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.

[2] Lee S Y, Mallet F, de Simone R. Dealing with AADL end-to-end flow latency with UMLMARTE[C]//Proc of 13th IEEE international conference on engineering of complex computer systems. [s. l.]:IEEE,2008.

[3] Feiler P H, Hansson J. Flow latency analysis with the Architecture Analysis & Design Language (AADL)[R]. [s. l.]: [s. n.],2008.

[4] 谯婷婷,王乐,耶国栋.基于AADL的软件可靠性验证[J].计算机应用,2012,32(S2):92-95.

[5] 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.

[6] Johnsen A. Architecture-based verification of dependable embedded systems[D]. Sweden:Mälardalen University,2013.

[7] Feiler P H, Lewis B A, Vestal S. The SAE Architecture Analysis & Design Language (AADL) a standard for engineering performance critical systems [C]//Computer aided control system design,2006 IEEE international conference on control applications,2006 IEEE International symposium on intelligent control. Munich, Germany:IEEE,2006:1206-1211.

[8] Alur R. Timed automata[M]//Computer aided verification. Berlin:Springer,1999.

[9] 徐仁佐.软件可靠性工程[M].北京:清华大学出版社,2007.

[10] 童超.基于时间自动机的RBC控车流程研究[D].成都:西南交通大学,2009.

[11] 朱雪阳,唐稚松.基于时序逻辑的软件体系结构描述语言XYZ/ADL[J].软件学报,2003,14(4):713-720.

[12] 李振松,顾斌.基于UPPAAL的AADL行为模型验证方法研究[J].计算机科学,2012,39(2):159-161.

[13] 周清雷,姬莉霞,王艳梅.基于UPPAAL的实时系统模型验证[J].计算机应用,2004,24(9):129-131.

[14] 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.

[1] 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.

[2] Lee S Y, Mallet F, de Simone R. Dealing with AADL end-to-end flow latency with UMLMARTE[C]//Proc of 13th IEEE international conference on engineering of complex computer systems. [s. l.]:IEEE,2008.

[3] Feiler P H, Hansson J. Flow latency analysis with the Architecture Analysis & Design Language (AADL)[R]. [s. l.]: [s. n.],2008.

[4] 谯婷婷,王乐,耶国栋.基于AADL的软件可靠性验证[J].计算机应用,2012,32(S2):92-95.

[5] 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.

[6] Johnsen A. Architecture-based verification of dependable embedded systems[D]. Sweden:Mälardalen University,2013.

[7] Feiler P H, Lewis B A, Vestal S. The SAE Architecture Analysis & Design Language (AADL) a standard for engineering performance critical systems [C]//Computer aided control system design,2006 IEEE international conference on control applications,2006 IEEE International symposium on intelligent control. Munich, Germany:IEEE,2006:1206-1211.

[8] Alur R. Timed automata[M]//Computer aided verification. Berlin:Springer,1999.

[9] 徐仁佐.软件可靠性工程[M].北京:清华大学出版社,2007.

[10] 童超.基于时间自动机的RBC控车流程研究[D].成都:西南交通大学,2009.

[11] 朱雪阳,唐稚松.基于时序逻辑的软件体系结构描述语言XYZ/ADL[J].软件学报,2003,14(4):713-720.

[12] 李振松,顾斌.基于UPPAAL的AADL行为模型验证方法研究[J].计算机科学,2012,39(2):159-161.

[13] 周清雷,姬莉霞,王艳梅.基于UPPAAL的实时系统模型验证[J].计算机应用,2004,24(9):129-131.

[14] 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.