基于构件交互自动机的 AADL 模型转换方法研究

李揭阳,李 勇,张福高
(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:构件交互自动机(Component-Interaction Automata,Co-IA)是扩展了构件之间交互描述的自动机。体系结构分析设计语言(Architecture Analysis and Design Language,AADL)是一种基于构件的半形式化体系结构分析和设计语言,是嵌入式系统体系结构建模和设计的标准,但无法直接进行形式化模型的检测工作。为了形式化描述系统交互过程中产生的大量数据,更好地描述模型中的状态集合、状态变迁和数据约束的性质,在构件交互自动机研究和发展的基础上,提出了一种扩充的构件交互自动机,将形式化规格说明语言 Z 引入构件交互自动机 Z-CoIA,描述模型中包含状态和状态变迁。为检测与验证所建立的模型,基于具体实例进行了由 AADL 模型向经扩充的构件交互自动机模型的转换。验证结果表明,所提出的方法推动了 AADL 的形式化进程。

关键词:体系结构分析设计语言;构件交互自动机;Z 语言;模型转换
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2017)07-0068-04
doi:10.3969/j.issn.1673-629X.2017.07.016

Investigation on AADL Model Transformation Method Based on Component-interaction Automata

LI Jie-yang,LI Yong,ZHANG Fu-gao
(School of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,
Nanjing 211106,China)

Abstract:Component-Interaction Automata (Co-IA) is a automata which has expanded the interaction between components. Architecture Analysis and Design Language (AADL) is a kind of component-based semi-formal architecture analysis and design language and is the modeling and design standard of embedded system architecture. But it cannot detect the formal model directly. In order to formally describe the large amounts of data produced by the interaction in the system and the characteristics of state sets,state change,data constraints in the model,the expended CoIA has been proposed on the basis of research and development of CoIA. And the formal specification language Z is introduced into the CoIA,named Z-CoIA,to describe the status and status changes in the model. So as to test and verify the model,transformation from the AADL to CoIA is carried out by a concrete example. The verification results show that the proposed method promotes the AADL formal process.

Key words:AADL;component-interaction automata;Z language;model transformation

0 引 言

近年来,AADL 逐渐发展成为复杂嵌入式实时系统的体系结构设计与分析语言标准。随着 AADL 的不断发展,针对其模型的验证工作也成为研究热点。其主要的研究方向是将 AADL 模型转换为扩展的自动机等形式的模型并进行验证。文献[1]将抽象的 AADL 子集模型转换为时间自动机模型并进行验证,并基于 OSATE 开发了 AADL 的模型验证工具。文献[2]将 AADL 模型转换为广义随机 Petri 网,并依托随机 Petri 网分析技术进行可靠性评估验证。

1987 年,I/O 自动机被提出。2001 年,L. de Alfaro 等提出了接口自动机的概念,此后又提出了时间自动机的概念,这些自动机并没有对构件之间的交互行为进行描述,不支持构件之间的接口交互。文献[3-5]提出了构件交互自动机,介绍了其优缺点并指出了将基于构件的模型自动化地转换为构件交互自动机的研究方向,并给出了基于时序逻辑的验证方法。文献[6]提出了基于时间构件交互自动机的建模方

收稿日期:2016-07-27 **修回日期**:2016-11-03 **网络出版时间**:2017-04-28
基金项目:国家"973"重点基础研究发展计划项目(2014CB744900);航空科学基金(20150652008)
作者简介:李揭阳(1992-),女,硕士生,研究方向为形式化方法。

法,并给出了时间构件交互自动机的相关定义、组合和验证算法。文献[7]在构件交互自动机的基础上,引入 QoS 约束条件,提出一种基于约束交互自动机的构件模型。

为了更好地描述系统交互过程中产生的大量数据,以及模型中的状态集合,状态变迁,数据约束的性质,文献[8]将 Z 语言引入接口自动机,提出一种结合自动机和 Z 语言的 ZIA 语言,并给出形式化的语法和语义定义。

在基于构件自动机演化的基础上,结合 AADL 的特点,提出一种基于 AADL 的扩充的构件交互自动机,将形式化规格说明语言 Z 引入构件交互自动机,给出 AADL 模型到构件交互自动机的形式化转换规则。通过具体实例进行了验证,为 AADL 模型形式化和自动化检测奠定了基础。

1　带 Z 的构件交互自动机及实例

1.1　构件交互自动机

自动机的概念自提出后,出现了接口自动机[9]、时间自动机[10]等多种扩充形式,这些自动机具有一般形式化的形式,能够用时序逻辑公式对其直接进行形式化验证,但不支持构件之间的接口交互,因此又提出了构件交互自动机描述构件间的交互行为。

定义 1:构件交互自动机可以刻画为一个五元组 $M=(Q,Q_0,A,\Gamma,H)$,其中,Q 表示非空的有限状态节点的集合;$Q_0 \subseteq Q$ 表示初始状态的集合;Σ 表示自动机中所有动作的集合,包括输入动作、输出动作和内部动作;$\Gamma \subseteq Q \times \Sigma \times Q$ 表示状态之间的转换关系集合;H 表示构件组合层次设计的构件名字的集合。

1.2　带 Z 的构件交互自动机

20 世纪 80 年代,英国牛津大学程序研究组的 Jean Raymond Abrial 等提出了 Z 语言。软件形式规格说明语言 Z 是基于一阶谓词逻辑和集合论的形式化语言,采用严格的数学理论,因此可以产生简明、精确、无歧义且可证明的规格说明。

Z 语言的核心是 Z 模式[11],有助于让规范说明变得结构化和模块化。一个模式由一些变量的声明和限制这些变量的值的谓词两部分组成,模式有垂直和水平两种形式。垂直模式更加简洁明了,垂直型的模式通用式定义如下:

S

$D_1;\cdots;D_m$

$P_1;\cdots;P_n$

其中,S 为模式名称;$D_1;\cdots;D_m$ 为声明部分;$P_1;\cdots;P_n$ 为谓词部分。

对应的水平模式表示法为:$S \triangleq [D_1;\cdots;D_m \mid P_1;\cdots;P_n]$。

此外,可将“?”、“!”和“'”添加在变量的后面,在 Z 语言中成为变量的修饰。同样,模式也可以修饰,将修饰应用到被修饰模式的声明部分中所有的变量,该模式的谓词部分中对应的变量也都被修饰。有了修饰,Z 语言就可以刻画下个状态中的变量与当前状态中的变量的关系。将 Z 语言引入自动机,接下来给出带 Z 的构件交互自动机的定义。

定义 2:带 Z 的构件交互自动机 $M=(Q,Q_0,A,V,\Gamma,F^A,F^V,H)$。其中,$Q$ 表示非空的有限状态节点的集合;$Q_0 \subseteq Q$ 表示初始状态的集合;Σ 表示自动机中所有动作的集合,包括输入、输出和内部动作;V 表示自动机中所有变量的集合,包括输入、输出和内部变量;$\Gamma \subseteq Q \times \Sigma \times Q$ 表示状态之间的转换关系集合;F^A 表示动作的映射函数,将动作集合 Σ 中的任意动作映射为某个 Z 中的操作模式,同时可以具体地将输入动作、输出动作、内部动作一一映射为输入操作模式、输出操作模式、内部操作模式;F^V 表示状态的映射函数,将集合 Q 中的任意状态映射为某个 Z 中的状态模式;H 表示构件组合层次设计的构件名字的集合。

1.3　带 Z 的构件交互自动机实例

给出一个构件交互自动机 M 的实例,包括以下元素:

(1) $Q=\{q_0,q_1,q_2,q_3\}$;

(2) $Q_0=\{q_0\}$;

(3) $A^I=\{a,c\}$,$A^O=\{d\}$,$A^H=\{b\}$;

(4) $V^I=\{x_1,x_2\}$,$V^O=\{y\}$,$V^H=\{z\}$;

(5) $\Gamma=\{(M,a,+),(M,b,M),(M,c,+),(M,d,-)\}$;

(6) $F^V(p_0)=S_0 \triangleq [z:N \mid z=0]$,

$F^V(p_1)=S_1 \triangleq [x_1?:N \mid x_1 \in N;z=0]$,

$F^V(p_2)=S_2 \triangleq [x_1:N;z:N \mid z=x_1]$,

$F^V(p_3)=S_3 \triangleq [x_2?:N;y!:N \mid x_2 \in N;z=x_1]$;

(7) $F^A(a)=A_a \triangleq [x_1?:N \mid x_1 \in N]$,

$F^A(b)=A_b \triangleq [x_1:N;z:N \mid z'=z+x_1]$,

$F^A(c)=A_c \triangleq [x_2?:N \mid x_2 \in N]$

$F^A(d)=A_d \triangleq [x_2 \in N;y!:N;z:N \mid y!=z*x_2]$;

(8) $H=\{M\}$;

自动机如图 1 所示。

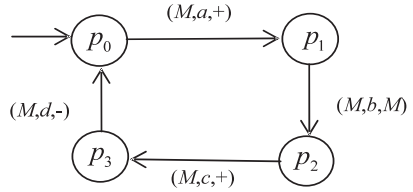

图 1　一个带 Z 的构件交互自动机

2 带 Z 的构件交互自动机的组合与验证

构件交互自动机具有层次性,可以组合形成更高层次的自动机。必须匹配所有的输入输出动作来构造新的自动机。

定义 3:构件交互自动机的组合 $S=(Q_C,Q_{C0},A_C,V_C,\Gamma_C,F_C^A,F_C^V,H_C)$,是构件交互自动机集合 $\{M_i=(Q_i,Q_{0i},A_i,V_i,\Gamma_i,F_i^A,F_i^V,H_i)\}$ 组合而成的构件交互自动机。其中,$i\in N$ 且参与组合的构件名集合 $\{(H_i)\}$ 互不相交;Q_C,Q_{C0} 为组合后的状态集合及初始状态,$A_C=\bigcup_{i\in N}A,V_C=\bigcup_{i\in N}V$;$\Gamma_C\subseteq\{X\times A_C\times(X\cup\{\pm\})\}$,$X\in\{(H_i)\}$ 是新的动作迁移的集合;F_C^A,F_C^V 为组合后的到 Z 语言的动作映射和变量映射;$H_C=\{(H_i)\}$,$i\in N$。

在组合构件交互自动机时,首先要考虑动作的匹配,如两个自动机互为输入输出模块,则可匹配为一个内部动作,在不需要考虑内部动作细节的情况下,内部动作可以记为 (X,τ,X)。输入为 $M_1=(Q_1,Q_{10},A_1,V_1,\Gamma_1,F_1^A,F_1^V,H_1)$ 和 $M_2=(Q_2,Q_{20},A_2,V_2,\Gamma_2,F_2^A,F_2^V,H_2)$ 两个构件自动机,输出为组合的构件自动机 $S=(Q_C,Q_{C0},A_C,V_C,\Gamma_C,F_C^A,F_C^V,H_C)$。同步动作变为内部动作,其余动作交互,生成新的状态集合 Q_C、动作集合 A_C 和标记转换系统 Γ_C,但要遍历所有状态,合成了新的状态集和动作集就可以将其重新映射为 Z 语言中的状态模式和操作模式,最后 $H_C=(H_1,H_2)$,生成带 Z 的组合构件交互自动机。

3 AADL 模型转换实例

2004 年,美国汽车工程师协会(SAE)在 MetaH,UML 的基础上提出了嵌入式实时系统体系结构分析与设计语言—AADL,并发布了 SAE AS5506 标准[12],为设计与分析嵌入式实时系统的软、硬件体系结构及功能与非功能性质提供了一种标准,将系统设计、分析、验证、自动代码生成等关键环节融合于统一框架之下。AADL 具有语法简单、功能强大、可扩展等优点[13]。介绍了如何将 AADL 模型的具体实例用构件交互自动机来描述。地铁门控制系统[14]结构如图 2 所示,包括列车控制系统、列车门控制系统、站台屏蔽门控制系统部分。列车门控制系统接收列车控制系统的开门及关门命令,列车门控制系统接收到命令后将数据进行处理并发送给站台屏蔽门控制系统,控制车门开关,同时,接收到车门状态信息后将其反馈给列车控制系统。

对列车控制系统用 AADL 进行建模,将整个系统用系统构件 train_system 描述,包括两个子构件:train_controller 进程构件和 door_system 系统构件。其中,door_system 包括 door_controller 和 door 两个进程构件,每个进程构件包含各自的线程构件,列车控制系统的部分 AADL 模型如图 3 所示。

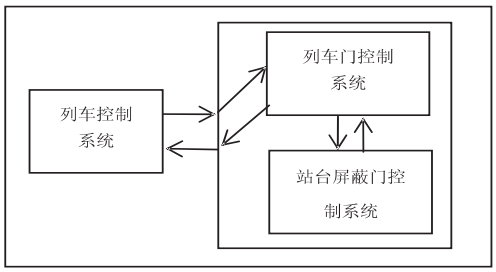

图 2 列车控制体系结构图

```
process door
features
x: in data port;
end door;

process door_controller
features
current_data:in data port;
new_feature:out data port;
end door_controller;

process train_controller
Features
current_data:out data port;
new_feature:in data port;
end train_controller;
door:system door_part;
train_controller:process train_controller;
end train_system.impl;
```

图 3 列车控制系统的 AADL 模型

根据列车控制系统的具体实例,说明将 AADL 模型转换为带 Z 的构件交互自动机的转换方法。

对进程构件 train_controller 用带 Z 的构件交互自动机 M_1 表示如下:

$$M_1=(\{p_0,p_1\},\{p_0\},\{a,b\},\{(M_1,a,+),(M_1,b,-)\},F^A,F^V,(M_1))$$

对进程构件 door_controller 用带 Z 的构件交互自动机 M_2 表示如下:

$$M_2=(\{p_0,p_1,p_2,p_3\},\{p_0\},\{a,b,c,d\},\{(M_2,a,+),(M_2,b,-),(M_2,c,+),(M_2,d,-)\},F^A,F^V,(M_1))$$

对进程构件 door 用带 Z 的构件交互自动机 M_3 表示如下:

$$M_3=(\{p_0,p_1\},\{p_0\},\{a,b\},\{(M_3,a,+),(M_3,b,-)\},F^A,F^V,(M_3))$$

运用构件交互自动机的组合方式,将进程构件 door_controller 和 door 组合为系统构件 door_system,将

内部动作隐藏,可以用构件交互自动机描述如下:

$$M_4 = (\{p_0, p_1\}, \{p_0\}, \{a, b\}, \{(M_4, a, +), (M_4, b, -)\}, F^A, F^V, (M_4))$$

4　结束语

为了形式化描述系统中大量的数据约束,在 AADL 建模语言及构件交互自动机研究的基础上,结合 Z 语言和自动机的优点,对构件交互自动机运用 Z 语言进行扩充,通过具体实例给出 AADL 模型到构件交互自动机的形式化转换规则,推动了 AADL 模型的形式化验证的进程。

参考文献:

[1] Yang Z, Hu K, Ma D, et al. From AADL to timed abstract state machines: a verified model transformation[J]. Journal of Systems & Software, 2014, 93(2): 42-68.

[2] 吴育春. 基于 AADL 的嵌入式软件形式化验证研究[D]. 西安:陕西师范大学,2014.

[3] Vareková P, Zimmerova B. Component-interaction automata for specification and verification of component interactions [C]//Proceedings of the IFM 2005 doctoral symposium on integrated formal methods. [s. l.]: [s. n.], 2005: 71-75.

[4] Brim L, Ivana Č, Vařeková P, et al. Component-interaction automata as a verification-oriented component-based system specification[J]. ACM SIGSOFT Software Engineering Notes, 2005, 31(2): 19-21.

[5] Zimmerova B, Vařeková P, Beneš N, et al. Component-interaction automata approach (CoIn)[M]//The common component modeling example. Berlin: Springer, 2008: 146-176.

[6] 贾仰理,张振领,李舟军. 基于自动机的构件实时交互行为的形式化模型[J]. 计算机科学,2010,37(9):151-156.

[7] 张玉玉. 基于约束交互自动机的构件行为一致性研究[D]. 哈尔滨:哈尔滨工程大学,2012.

[8] Cao Z. Temporal logics and model checking algorithms for ZIAs[C]//2nd international conference on software engineering and data mining. [s. l.]: IEEE, 2010: 57-62.

[9] 张　岩,胡　军,于笑丰,等. 接口自动机-一种用于组件组合的形式系统[J]. 计算机科学,2005,32(11):212-217.

[10] 陈　伟,薛云志,赵　琛,等. 一种基于时间自动机的实时系统测试方法[J]. 软件学报,2007,18(1):62-73.

[11] Bowen J P. Formal specification and documentation using Z: a case study approach [M]. London: International Thomson Computer Press, 1996.

[12] 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.

[13] 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.

[14] 舒新峰,张炎龙,孙林泽. 基于 Spin 的地铁门控制系统建模与验证[J]. 西安邮电大学学报,2015,20(5):57-61.

(上接第 67 页)

势,对图像中目标的轮廓边缘进行锐化,使目标清晰,提高了图像的主观可视效果和客观参数。通过客观数据证明,与其他三种主流的去雾算法相比,处理后的图像中目标轮廓明显,细节清楚,噪声明显减少,可以保障图像用于后期处理,使其处理后的图像边缘纹理更加清晰。

参考文献:

[1] 李　滚,吴劼夫,雷志勇. 图像雾霾等级评价及去雾技术研究进展[J]. 激光杂志,2014,35(9):1-6.

[2] 李庆义. 计算机图像处理技术综述[J]. 科技情报开发与经济,2007,17(11):226-228.

[3] 方　莉,张　萍. 经典图像去噪算法研究综述[J]. 工业控制计算机,2010,23(11):73-74.

[4] Land E H. The Retinex theory of color vision[J]. Scientific American, 1977, 32: 108-128.

[5] 毕　杨. 一种快速的曲波变换图像去噪算法[J]. 科技信息,2009(16):11-12.

[6] Kimmel R, Elad M, Shaked D, et a1. A variational framework for Retinex [J]. International Journal of Computer Vision, 2003, 52(1): 7-23.

[7] 林笑君,梁凤梅. 基于 Retinex 的一种图像去雾算法[J]. 电视技术,2013,37(17):155-158.

[8] Candes E J, Donoho D L. Recovering edges in III-posed inverse problems: optimality of curvelet frames[J]. Annals of Statistics, 2002, 30(3): 784-842.

[9] Candes E J, Demanet L, Donoho D L. et al. Fast discrete curvelet transforms [J]. Multiscale Modeling and Simulation, 2005, 5(3): 861-899.

[10] 谷秀平. 基于 Curvelet 变换的图像去噪和增强[D]. 天津:天津理工大学,2010.

[11] 颜　兵,王金鹤,赵　静. 基于均值滤波和小波变换的图像去噪技术研究[J]. 计算机技术与发展,2011,21(2):51-53.

[12] 郭　璠,蔡自兴,谢　斌,等. 图像去雾技术研究综述与展望[J]. 计算机应用,2010,30(9):2417-2421.

[13] 李冠章,罗武胜,李　沛. 一种高效地修正 Retinex 图像自适应对比度增强算法[J]. 测试技术学报,2009,23(5):445-451.

[14] 马云飞,何文章. 基于小波变换的雾天图像增强方法[J]. 计算法应用与软件,2011,28(2):71-72.

[15] 占　俊. 几种计算机数字图像技术的处理效果研究[J]. 现代电子技术,2015,38(21):32-35.