

## ◎博士论坛◎

# 结构分析和设计语言 AADL 研究

王瀚博,周兴社,董云卫,唐 蕾

WANG Han-bo,ZHOU Xing-she,DONG Yun-wei,TANG Lei

西北工业大学 计算机学院,西安 710072

College of Computer, Northwestern Polytechnical University, Xi'an 710072, China

E-mail: wanghb@mail.nwpu.edu.cn

WANG Han-bo,ZHOU Xing-she,DONG Yun-wei, et al. Research on architecture analysis and design language. Computer Engineering and Applications, 2009, 45(16):1-4.

**Abstract:** The focus of embedded system development is shifted from the code level to the model level with the increasing scale, complexity and performance of embedded system and the model-driven architecture stands for a general orientation of embedded system development. AADL is a new standard of model-driven system engineering. The process, the method and the tools are discussed and the technologies including model analysis, model transformation and code generation are also discussed. At last AADL is compared with UML which is the standard modeling language of OMG.

**Key words:** Architecture Analysis and Design Language(AADL); embedded system; modeling; model driven

**摘要:**随着嵌入式系统规模、复杂度和性能需求的提升,嵌入式系统开发的重点从代码级提前到模型级,模型驱动体系结构成为嵌入式系统开发的主流。结构分析和设计语言(AADL)是一种模型驱动系统工程的新标准,从过程、方法和工具三个方面对其进行研究,讨论模型分析、模型转换和代码生成等相关技术,并将其与OMG的标准建模语言UML进行比较。

**关键词:**结构分析和设计语言;嵌入式系统;建模;模型驱动

DOI:10.3778/j.issn.1002-8331.2009.16.001 文章编号:1002-8331(2009)16-0001-04 文献标识码:A 中图分类号:TP391.8

## 1 引言

随着微电子、通信等技术发展,嵌入式系统得到广泛的应用,全球用于嵌入式系统的微处理器销售额已占总销售额的85%以上<sup>[1]</sup>。除了传统的消费电子产品,嵌入式系统越来越多地应用在工业控制、汽车、航空航天等任务关键和实时领域,系统规模、复杂度和性能需求不断提升。传统的以代码为核心的开发方法已经不能适应这种趋势,开发的重点从代码级提前到模型级,以提供在早期进行系统集成的可行性。为此,2001年OMG(Object Management Group)提出模型驱动体系结构(Model Driven Architecture, MDA),用来处理软件开发模型。MDA的核心思想是从系统使用其硬件能力的细节中分离出系统运行的规范<sup>[2]</sup>,提供独立于硬件平台的系统描述、硬件平台的描述,选择硬件平台并将平台无关的系统规范转换为平台相关的系统规范,MDA最重要的标准是统一建模语言(Unified Modeling Language, UML)。UML基于面向对象方法,以简单的符号表示和严格的语义框架描述系统的结构、行为和通信,尤其适合于对嵌入式系统建模。任务关键的实时嵌入式系统还要

求建立软件到硬件的映射,既能够满足功能属性要求又能够满足非功能属性要求,还能够对一个模型进行多元分析以便在调整系统结构时可以从多个域进行权衡<sup>[3]</sup>。为此SAE(Society of Automotive Engineers)在UML的基础上提出结构分析和设计语言(Architecture Analysis and Design Language, AADL),受到卡内基梅隆大学、空中客车公司等学术界和工业界的关注和接受,并正在被我国的航空工业界认可和采用。

## 2 AADL 标准

AADL用来设计和分析系统的软硬件结构,包括独立的组件和它们的交互,尤其适合性能关键的实时嵌入式系统。AADL的发展经历了三个阶段:首先SAE在2004年11月发布了AS5506<sup>[4]</sup>标准,并将AADL的名称由航空分析和设计语言(Avionics Analysis and Design Language)改为结构分析和设计语言,AS5506定义了AADL的核心——具有精确语义的建模概念;然后在2006年6月发布了AS5506/1<sup>[5]</sup>标准,AS5506/1为AS5506提供了四个附件——图形化符号附件、元模型及交

**基金项目:**国家自然科学基金(the National Natural Science Foundation of China under Grant No.60736017);国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2007AA010304)。

**作者简介:**王瀚博(1983-),男,博士研究生,主要研究方向:嵌入式计算,网格计算;周兴社(1955-),男,教授,博士生导师,主要研究方向:分布/并行计算,嵌入式计算等;董云卫(1968-),男,博士,教授,主要研究方向:嵌入式计算,软件工程;唐蕾(1983-),女,博士研究生,主要研究方向:普适计算,网格计算。

收稿日期:2008-12-23 修回日期:2009-01-22

换格式附件、语言顺应及应用程序接口附件和错误模型附件，这一阶段相继发布一系列 AADL 工具；目前（2008 年 10 月），AADL 的新标准 AADL V2.0 正在进行第二次投票，AADL V2.0 对子程序建模、异步系统语义和动态加载语义、虚拟总线和协议建模等进行了改进。

AADL 标准由文本和图形语言、XML/XMI 交换格式、UML2.0 轮廓和错误模型附件组成，其中 AADL 语言是 AADL 标准的核心，其元素及关系如图 1 所示<sup>[6]</sup>。AADL 语言和 UML 一样基于面向对象的思想，组件类型（Component Type）和组件实现（Component Implementation）分别与面向对象程序设计语言中的类和对象的概念相似，Component Implementation 对客观世界中嵌入式系统的组件进行抽象，Component Type 则是对 Component Implementation 的抽象和归纳。一个 Component Type 需要从下面几项描述：标识符（identifier）、特征（features）、流（flows）和属性（properties），这也与面向对象程序设计语言中类的表示符、属性和方法的概念相似，并且可以使用关键词 extends 进行继承。Component Type 的组件又细分为三类：软件类包括数据（data）、子程序（subprogram）、线程（thread）、线程组（thread group）和进程（process）；执行平台类包括存储器（memory）、设备（device）、处理器（processor）和总线（bus）；系统类包括系统（system）。AS5506 对每一种组件的表示（文本和图形）、属性和约束进行规定。Component Implementation 是 Component Type 的实现，首先需要描述其精化的 Component Type，与 Component Type 的描述相比还包含子组件（subcomponents）、连接（connections）、调用顺序（call sequences）和模式（modes）。Component Implementation 中 connections 是 Component Type 中 features 的精化，可以细分为端口（port）、访问（access）和参数（parameter），但不包括 features 中的子程序（subprogram）。modes 还可以细分为模式描述和模式转换（mode transitions）。Component Type 和 Component Implementation 可以引用属性集（Property Set）和包（Package），Property Set 是组织为具有独立名空间的独立单元的组件声明集合，Package 则把具有相同属性的元素集合起来通过 Package 名称进行引用。AADL 语言通过附件（annex）进行扩展。

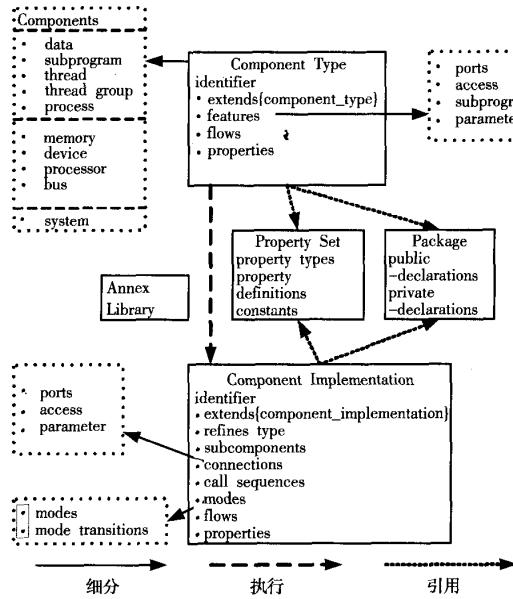


图 1 AADL 元素及关系

AADL 能够建立软件到硬件的映射，支持功能属性和非功能属性的描述，并可以对同一个模型进行多元分析，包括可用性和可靠性分析（平均故障间隔时间等）、数据质量分析（数据精度等）、实时性能分析（执行时间等）、资源消费分析（带宽等）、安全性分析（入侵检测等）。

### 3 AADL 建模

#### 3.1 建模过程

AADL 使用具有正确语法和语义声明的规范进行建模，这些规范可以表示为 AADL 文本、可扩展标记语言（Extensible Markup Language, XML）或者 AADL 图形，这三种表示法是等效的。典型的 AADL 建模过程如图 2 所示。使用 AADL 对嵌入式系统建模的前提是进行需求分析，与一般的需求分析不同，除了功能属性需求还支持非功能属性需求。AADL 不仅可以在不同的阶段进行建模，而且可以从不同的层次进行建模：顶层设计主要描述系统的各个模块及模块间的关系；底层设计主要描述模块的软硬件组成、软硬件与模块、软硬件与软硬件之间的关系。AADL 建模首先要创建模型组件，按照模块的类型创建相应的 Component Type 并按照需求分析为创建的 Component Type 添加 features，在此基础上创建每一个 Component Type 的 Component Implementation。所有模块的 Component Type 和 Component Implementation 创建后，还需要创建一个特殊组件——容器系统（Containing System）的 Component Type 和 Component Implementation，用来标明整个嵌入式系统的边界。接下来需要填充系统实现，即为 Containing System 和模型组件中 System 类型的组件添加子组件并连接子组件与子组件、子组件与模型组件的端口，这里的子组件既包括创建的模型组件也包括一些声明过的 Package。模型组件和系统实现完成顶层设计的建模，硬件组件和软件组件对底层设计进行建模，将顶层设计的模型映射到执行平台组件和应用软件组件的

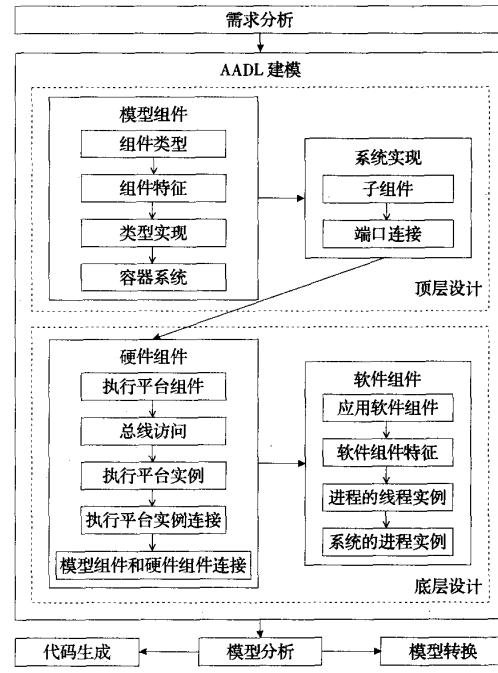


图 2 AADL 建模过程

模型。创建硬件组件和软件组件首先创建 Component Type, 并添加 features, 然后创建相应的 Component Implementation。硬件组件还需要添加所需的 bus access, 创建执行平台实例, 连接实例并把模型组件和硬件组件连接起来。软件组件还需要创建 process 组件和 thread 实例, 并把 thread 实例添加到 process 组件中, 然后创建 process 的实例并添加到 System 类型的组件中。最后, 对 AADL 模型进行分析后, 进行模型转换或者代码生成。

按照上述过程对一个简单的速度控制器建模<sup>[7]</sup>, 以下是部分模型的 AADL 文本表示, 图 3 是这个部分模型的 AADL 图形表示。

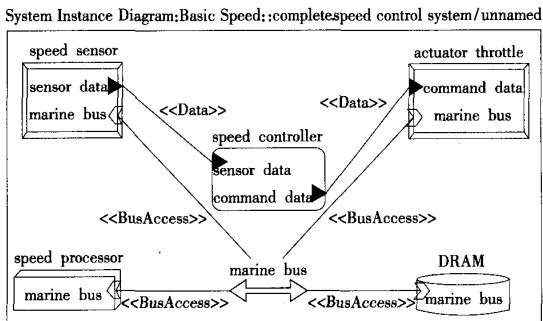


图 3 AADL 图形表示

```
system implementation complete.speed_control_system
subcomponents
speed_sensor:device sensor.speed;
actuator_throttle:device actuator.throttle;
speed_controller:system control.speed;
speed_processor:processor basic.marine_certified;
marine_bus:bus standard.marine_certified;
DRAM:memory DRAM.marine_certified;
connections
  data port speed_sensor.sensor_data->speed_controller.sensor_data;
  data port speed_controller.command_data->actuator_throttle.command_data;
  bcoon1:bus access marine_bus->speed_processor.marine_bus;
  bcoon2:bus access marine_bus->DRAM.marine_bus;
  bcoon3:bus access marine_bus->speed_sensor.marine_bus;
  bcoon4:bus access marine_bus->actuator_throttle.marine_bus;
properties
  Allowed_Processor_Binding=>reference speed_processor;
end complete.speed_control_system;
```

### 3.2 建模方法

从 AADL 的建模过程可以看出 AADL 使用一种逐步精化的建模方法, 事实上 AADL 作为模型驱动的新标准, 支持模型驱动体系结构。MDA 将模型分为两种: 平台无关模型(Platform Independent Model, PIM), 描述从执行平台抽象的功能和结构; 平台相关模型(Platform Specific Model, PSM), 描述特定执行平台上的功能和结构。AADL 构建 PIM 时在构建模型组件阶段只使用软件类 data、subprogram、thread、thread group、process 和系统类 system 的 Component Type 和 Component Implementation, 在填充系统实现阶段为 Containing System 和模型组件中 System 类型的组件添加软件类的子组件并进行连接, 并提前构建软件组件进程的线程实例和系统的进程实例。AADL

构建 PSM 时在构建硬件组件阶段完成执行平台实例, 连接属于 PIM 的模型组件与硬件组件, 实现应用软件到执行平台的映射。MDA 中一个 PIM 可以对应多个 PSM, AADL 中 PIM 和 PSM 可以是一对多的关系, 一般情况下是一对一的关系。

### 3.3 建模工具

现有的 AADL 工具主要包括:

OSATE(Open-Source AADL Tool Environment)<sup>[8]</sup>: SEI 基于 Eclipse 插件开发的开源工具, 用于 AADL 模型编辑、编译和分析的集成开发环境, 并提供 AADL 模型与 MetaH 模型的转换等相关功能。

STOOLD<sup>[9]</sup>: 欧洲 Ellidiss 公司开发的可视化建模工具, 使用自上而下的建模方法以缩小设计模型和运行代码之间的差距, 可以从 AADL 代码生成 Ada 代码和 C++ 代码。

Cheddar<sup>[10]</sup>: 法国布雷斯特大学 LISyC 小组开发的实时调度工具, 用来检查使用 AADL 描述的嵌入式系统时间约束, 既可以作为一个独立的仿真引擎又可以作为 STOOLD 的插件运行。

除了上述主要工具, 还有 AADL 仿真器 ADes、基于 AADL 的无线传感器设计工具 ANDES、AADL 的 UML 2.0 Profile 等, 这些工具按功能可以划分为 AADL 集成建模(模型分析、模型转换、代码生成)环境、基于属性的 AADL 仿真环境、面向领域的 AADL 开发环境。

## 4 AADL 技术

### 4.1 模型分析

模型分析是 AADL 最重要的技术之一, 并可以对同一个模型进行多元分析, OSATE 支持的模型分析对象等见表 1。

表 1 AADL 模型分析

对象	内容	类型	属性	方法
模型	语义、健全性	语义	可靠性	静态分析
线程	绑定、优先级	调度	可靠性	静态分析/动态分析
系统	受限调度	调度	可用性	动态分析
总线	负载、功率	结构	资源消费	静态分析
资源	分配、预算	结构	资源消费	动态分析
连接	需求、一致性	语义/结构	可靠性	静态分析
安全	safety、security	安全	安全性	静态分析
属性	循环引用	语义	可靠性	静态分析
流	反应时间	流	实时性	动态分析

AADL 可以对模型、线程、系统等进行分析, 每一个对象包括若干可以分析的内容, 可以分为结构、语义、调度、安全和流类型, 分别适用静态分析方法和动态分析方法, 支持可用性/可靠性、实时性、资源消费和安全性的分析。

AADL 模型分析包括对 AADL 模型的分析和对 AADL 实例模型的分析, AADL 实例模型是 AADL 模型中系统的实例, 具有处理器绑定、存储器绑定、线程处理协议等 AADL 属性值和带宽能力、处理能力、服务质量协议等 SAE 属性值。一些分析对象的内容既支持 AADL 模型, 又支持 AADL 实例模型, 另一些分析对象的内容仅支持 AADL 模型或 AADL 实例模型, 主要由分析对象内容具有的属性值决定。Security 和 Safety 分析支持 AADL 模型和 AADL 实例模型, Security 分析检查具有向外连接的组件安全级别是否超过所连接组件的安全级别, Safety 分析检查具有向外连接的组件安全级别是否高于连接末组件的安全级别。流反应时间分析仅支持 AADL 实例模型,

确定组件的流实现反应时间并与组件的流规范反应时间比较，流反应时间分析的另一种形式是确定组件间端到端的流规范反应时间。

## 4.2 模型转换

AADL 模型转换包括对应用软件的转换、对执行平台的转换和对应用软件与执行平台映射关系的转换，目标是保证原模型和目标模型的一致性，关键是如何减少转换带来的性质丢失。AADL 模型转换主要是建立两种模型的元素转换规则，如果两种模型的元素不是双射关系，那么建立相关元素组成的 AADL 属性集。AADL 是一种半形式化标准，目标模型的形式化程度越高，转换的难度就越大。因此，AADL 模型和 Meta 模型的转换得到 OSATE 的支持，而 AADL 模型和 UML 模型的转换还处在原型阶段。AADL 模型转换为 MetaH 模型时，首先进行 AADL 组件到 MetaH 类型的转换，然后进行 AADL 属性到 MetaH 模型属性的转换。AADL 组件转换为 MetaH 类型时，按照 MetaH 类型使用顺序，保证 MetaH 类型先声明后使用。对于 AADL 模型中 system 类型的组件如果具有执行平台层次，那么转换为 MetaH 模型的宏，否则转换为 MetaH 模型的声明<sup>[11]</sup>。

## 4.3 代码生成

AADL 代码生成指从 AADL 模型生成某种程序设计语言代码，代码生成是一个基于映射规则的精化过程，一个可用的映射规则集合至少由包的映射规则、端口的映射规则、进程组件的映射规则和线程执行的映射规则组成<sup>[12]</sup>。代码生成具有较高的自动化程度，包括语法分析自动化、映射规则自动化和代码书写自动化，使开发者不必纠缠于实现的细节。与模型转换不同，建立模型到代码的映射规则不仅需要考虑模型的平台相关性，而且必须考虑代码的平台相关性。代码生成的另一个难点是如何验证映射规则的有效性，保证代码能够实现模型的功能，并且不会产生模型没有的错误。AADL 代码生成取得很大进展，从 AADL 模型生成 Ada 代码、C 代码和 JAVA 代码已经实现，并得到 Stood 等工具支持，但是在自动化程度、平台相关性和规则有效性三个方面均存在不足。

## 5 AADL 与 UML

AADL 具有程序设计语言的风格，目标是定义运行时的系统行为，通过文本和图形描述系统的结构。UML 具有图表图形表达的风格，目标是描述系统功能的结构，通过图表和图形对功能的结构建模。AADL 具有形式化的语法和语义，可以检查 AADL 语言的语法和语义验证模型的一致性和正确性，这些分析建立在系统结构的顶层。UML 虽然可以通过图表和图形建模复杂的软件结构，但是不能定义图表、图形之间的关系，只能孤立地表达软件某一方面的性质。除此之外，AADL 通过附件进行扩展，UML 通过模板、标签和约束进行扩展。AADL 主要应用于实时嵌入式系统的设计和分析，UML 具有广泛的应用范围，除了软件还可以用来描述商业过程。AADL 和 UML 可以实现互补，通过物理模型、功能模型和结构模型的集成，共同提高开发效率<sup>[13]</sup>。 AADL 与 UML 的区别见表 2。

表 2 AADL 与 UML

	AADL	UML
基础	语言	图表
目标	定义运行行为	描述功能结构
表达	文本和图形	图表和图形
验证	自动分析	无
扩展	附件	模板、标签、约束
应用	实时嵌入式系统	软件、商业、过程

## 6 结束语

本文首先介绍 AADL 标准，然后从过程、方法与工具三个方面研究和实现基于 AADL 的嵌入式系统建模，并讨论 AADL 模型分析、模型转换和代码生成技术，最后将其与 UML 进行比较。可以看出，AADL 支持在整个开发周期中对系统进行建模，缩短开发时间，减低开发成本，升级系统能力，可以尽早发现设计中的问题尤其是非功能属性。AADL 已经成为航空电子设计的重要标准，可以更好地支持对飞控系统的设计，下一步的工作是使用 AADL 对新一代无人机飞控场景进行建模和分析。

## 参考文献：

- [1] 嵌入式与实时系统开发：使用 UML、对象技术、框架与模式[M].柳翔,译.北京：机械工业出版社,2004.
- [2] Miller J, Mukerji J. MDA guide version 1.0.1[EB/OL].[2003].http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf.
- [3] Cao Hui, Yan Ting, Pereira L R. Use AADL to analyze and design embedded systems[EB/OL].[2007].http://www.embedded.com/design/opensource/201201811?\_requestid=1246710.
- [4] SAE AS5506[S].[2004-11-05].http://www.sae.org/technical/standards/AS5506.
- [5] SAE AS5506/1[S].[2006-06-15].http://www.sae.org/technical/standards/AS5506/1.
- [6] Feiler P H, Gluch D P, Hudak J J. The Architecture Analysis & Design Language (AADL): An introduction [EB/OL].[2006].http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tn011.pdf.
- [7] The SEI AADL Team. Getting started with the AADL and OSATE: An introductory tutorial using the AADL textual and object model editors[EB/OL].[2006].http://www.aadl.info/downloads/AadlStarterKit.zip.
- [8] The SEI AADL Team. OSATE: Plug-ins for front-end processing of AADL models[EB/OL].[2008].http://la.sei.cmu.edu/aadl/currentsite/tool/osate.html.
- [9] Ellidiss. STOOD[EB/OL].[2008].http://www.ellidiss.com/stood\_detail.shtml.
- [10] LISyC. The Cheddar project: A free real time scheduling analyzer [EB/OL].[2008].http://beru.univ-brest.fr/~singhoff/cheddar/.
- [11] The SEI AADL Team. SAE AADL online help [EB/OL].[2006].http://www.aadl.info.
- [12] Ma Liang, Gui Sheng-lin, Luo Lei, et al. Research of automatic code generating technology based on AADL[C]//The 2008 International Conference on Embedded Software and Systems Symposia, 2008.
- [13] Dionisio de Niz. Diagrams and languages for model-based software engineering of embedded systems: UML and AADL[EB/OL].[2007].http://la.sei.cmu.edu/aadl/currentsite/documents/UML\_AADL\_Comparison.pdf.

# 结构分析和设计语言AADL研究

作者: 王瀚博, 周兴社, 董云卫, 唐雷, WANG Han-bo, ZHOU Xing-she, DONG Yun-wei, TANG Lei  
作者单位: 西北工业大学, 计算机学院, 西安, 710072  
刊名: 计算机工程与应用 [ISTIC PKU]  
英文刊名: COMPUTER ENGINEERING AND APPLICATIONS  
年, 卷(期): 2009, 45(16)  
被引用次数: 1次

## 参考文献(13条)

1. Ellidiss STOOD
2. The SEI AADL Team OSATE:Plug-ins for front-end processing of AADL models
3. The SEI AADL Team Getting started with the AADL and OSATE:An introductory tutorial using the AADL textual and object model editors
4. Feiler P H;Cluch D P;Hudak J J The Architecture Analysis & Design Language (AADL):An introduction
5. SAE. AS5506/1 2006
6. SAE. AS5506 2004
7. Cao Hui;Yan Ting;Pereira L R Use AADL to analyze and design embedded systems
8. Miller J;Mukerji J MDA guide version 1.0.1
9. Dionisio de Niz Diagrams and languages for model-based software engineering of embedded systems:UML and AADL
10. Ma Liang;Gui Sheng-lin;Luo Lei Research of automatic code generating technology based on AADL  
2008
11. The SEI AADL Team SAE AADL online help
12. LISyC The Cheddar project:A free real time scheduling analyzer
13. 柳翔 嵌入式与实时系统开发:使用UML、对象技术、框架与模式 2004

## 本文读者也读过(10条)

1. 陶勇, 桂盛霖, 马亮, 尹立孟. TAO Yong, GUI Sheng-lin, MA Liang, YIN Li-meng AADL模型的代码自动生成及集成技术[期刊论文]-计算机工程2009, 35(8)
2. 冯冰, 杨志义, 董云卫, 马春燕. 一种面向AADL架构的模型测试方法[期刊论文]-计算机测量与控制2010, 18(4)
3. 马春燕, 董云卫, 朱宇峰, 陆伟, Ma Chunyan, Dong Yunwei, Zhu Yufeng, Lu Wei AADL 测试模型的构造研究[期刊论文]-西北工业大学学报2010, 28(6)
4. 杨志斌, 皮磊, 胡凯, 顾宗华, 马殿富, YANG Zhi-Bin, PI Lei, HU Kai, GU Zong-Hua, MA Dian-Fu 复杂嵌入式实时系统体系结构设计与分析语言:AADL[期刊论文]-软件学报2010, 21(5)
5. 宋翠叶, 杜承烈, 李刚, SONG Cui-ye, DU Cheng-lie, LI Gang 基于AADL的软件开发技术研究[期刊论文]-计算机应用研究2009, 26(9)
6. 王庚, 周兴社, 张凡, 董云卫, WANG Geng, ZHOU Xing-she, ZHANG Fan, DONG Yun-wei AADL模型的测试方法研究[期刊论文]-计算机科学2009, 36(11)
7. 郭富磊 AADL在模型驱动中的应用研究[学位论文]2010
8. 刘倩, 桂盛霖, 李允, 罗蕾, LIU Qian, GUI Sheng-lin, LI Yun, LUO Lei 基于UPPAAL的AADL模型可调度性验证[期刊论文]-计算机应用2009, 29(7)

9. 刘雪琴. 桂盛霖. 罗蕾. 罗惠琼. LIU Xue-qin. GUI Sheng-lin. LUO Lei. LUO Hui-qiong AADL模型. 计算机应用研究[期刊论文]-计算机应用研究2008, 25 (12)
10. 杨志义. 张琛雨. 董云卫. Yang Zhiyi. Zhang Chenyu. Dong Yunwei AADL软件容错系统建模与评估[u]. 计算机测量与控制2009, 17 (4)

#### 引证文献(4条)

1. 刘博. 李蜀瑜 基于NuSMV的AADL行为模型验证的探究[期刊论文]-计算机技术与发展 2012 (2)
2. 李振松. 蒋志雄. 顾斌 AADL模式转换设计方法研究[期刊论文]-计算机工程与设计 2011 (12)
3. 李振松. 顾斌 基于AADL的中断控制设计方法[期刊论文]-微型机与应用 2011 (10)
4. 庞凤颖. 刘勇. 黄敏 接口控制文件仿真方法研究[期刊论文]-车辆与动力技术 2010 (1)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_jsjgcyy200916001.aspx](http://d.g.wanfangdata.com.cn/Periodical_jsjgcyy200916001.aspx)