

PEARSON

MODEL-BASED ENGINEERING WITH **AADL**

An Introduction to the SAE
Architecture Analysis & Design Language

使用AADL的模型基工程 ——SAE体系结构分析和设计语言入门

(英) 彼得·H. 费勒 (Peter H. Feiler)

戴维·P. 格鲁茨 (David P. Gluch)

著

光电控制技术重点实验室

译

航空工业出版社

PEARSON

使用 AADL 的模型基工程

——SAE 体系结构分析和设计语言入门

(英) 彼得·H. 费勒 (Peter H. Feiler) 著
戴维·P. 格鲁茨 (David P. Gluch)
光电控制技术重点实验室 译

航空工业出版社
北 京

内 容 提 要

本书由 AS 5506 标准主编成员所著,书中简练地介绍了 AADL 的概念、结构和使用,以及 AADL 为何是验证软件系统体系结构的 MBE 的有效工具。全书为学习 AADL 的读者提供了开发和分析基本系统模型的足够细节。它是使用 AADL 促进 MBE 开发过程最佳化的首个指南,也是市面上可获得的学习 AADL 的唯一教材。本书假定读者已经具备核心软件工程实践(如软件体系结构、软件设计)的基本知识、实时系统(如并发性、调度、通信),以及计算机运行期概念(如线程、执行语义)的知识。

本书提供的设计范例工程适用性很强,既能用作嵌入式实时应用的软件体系结构或软件系统工程课程的教材,又可供广大软件工程设计人员作为手册性参考资料。

图书在版编目(CIP)数据

使用 AADL 的模型基工程:SAE 体系结构分析和设计语言入门/(英)费勒(Feiler, P. H.), (英)格鲁茨(Gluch, D. P.)著;光电控制技术重点实验室译. --北京:航空工业出版社,2014.6

书名原文:Model-based engineering with AADL: an introduction to the SAE architecture analysis and design language

ISBN 978-7-5165-0405-5

I. ①使… II. ①费…②格…③光… III. ①面向对象语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 179 号



北京市版权局著作权合同登记
图字:01-2013-6864

使用 AADL 的模型基工程
——SAE 体系结构分析和设计语言入门
Shiyong AADL de Moxingji Gongcheng
——SAE Tixi Jiegou Fenxi he Sheji Yuyan Rumen

航空工业出版社出版发行

(北京市朝阳区北苑 2 号院 100012)

发行部电话:010-84934379 010-84936343

北京地质印刷厂印刷

2014 年 6 月第 1 版

开本:787×1092 1/16

印数:1—3000

全国各地新华书店经售

2014 年 6 月第 1 次印刷

印张:20.25 字数:504 千字

定价:95.00 元

原版前言

本书中，我们向读者介绍了SAE（原来称汽车工程师协会）体系结构分析和设计语言（AADL）的概念、结构和使用，以及AADL为何是验证软件系统体系结构的模型基工程（Model-Based Engineering, MBE）的有效工具。如果你正在学习AADL，我们为你提供了能够开发和分析基本系统模型的足够细节。通过掌握本书中的材料获得的核心技能，将为你提供建立自己的AADL和MBE经验的基础。即使你是一个AADL方面学识渊博的用户，我们预计你也会发现本书是一本有价值的参考资料。

MBE和AADL是什么和有什么用

模型基工程旨在建立和分析系统模型，这样就能够预测和了解该系统的能力和工作质量属性（如性能、可靠性或信息安全性）。在整个寿命期内，通过这样做，你就能够揭示系统级的问题——而这些问题，在系统集成和验收试验之前，通常是无法发现的——同时避免在开发和维修中进行为时已晚的、昂贵的返工。过去，已经就各种各样的系统组成以及各种不同的分析建立了独立的模型。系统性的和更为连贯的方法是一种体系结构中心方法。体系结构中心方法专注于系统级的问题，并保持一系列自相容的系统分析视角，这样就能在该系列范围内保持体系结构变化中各种独立分析的有效性。

AADL是SAE国际标准（AS 5506A^①）（其第二次修订版AS 5506B已于2012年9月发布——译者注）。AADL是用于模型基软件系统工程的统一框架，你可使用它获取静态的模块化体系结构、以通信任务为基础的运行时体系结构、用于部署软件的计算机平台体系结构，以及与之进行系统交互的任何物理系统或环境。你可以在单一的体系结构模型中获取静态结构和动态，并且解释其与各种工作特性的分析相关的信息。由AADL提供的概念，如线程、进程或设备，已经明确地定义了执行语义，这种执行语义能使你既可进行不太重要的系统分析，又可进行正式的系统分析。此外，利用其扩展性的结构组成，你和工具开发商能够将自定义分析和规范技术与核心AADL能力结合起来，以便为体系结构建模和分析建立一个完整的工程环境。

在开发AADL模型时，你将你的系统体系结构表示为各交互组件的分层结构。将软件、硬件和物理组件的接口规范和实现蓝图，组织在支持大规模和基于团队开发的软件

^① AS 5506A标准原版发布于2004年11月。本书涵盖其发布于2009年1月的修订版（A版），以及于2012年批准的勘误更正。有关AADL的进一步信息可浏览网站 www.aadl.info。购买该标准的复制件，可访问网站 www.sae.org/technical/standards/AS5506A。——原书注

包内。

作为标准, AADL 向你提供了在适当的技术中常常发现不了的稳定性, 并且允许你参与定义语言增强的工作。扩展 AADL 框架的整套标准的附加要素, 可参见 SAE 体系结构分析和设计语言 (AADL) 附件卷 1 (*SAE Architecture Analysis and Design Language (AADL) Annex Volume 1*) (AS 5506/1) 和 SAE 体系结构分析和设计语言 (AADL) 附件卷 2 (*SAE Architecture Analysis and Design Language (AADL) Annex Volume 2*) (AS 5506/2)。作为标准于 2006 年 6 月发布的 SAE AS 5506/1 定义了如下附件: AADL 图形表示附件、AADL 元模型与交换格式附件、语言一致性和应用程序接口附件, 以及错误模型语言附件。作为标准于 2011 年 1 月发布的 SAE AS 5506/2 定义了如下附件: 行为建模附件、利用 AADL 实现数据建模的指南附件, 以及 ARINC 653 分区体系结构建模附件^①。

哪些人能从阅读本书中获益

如果你是一位软件依赖系统的开发者, 那么不管你是一位系统体系结构师或软件体系结构师、系统工程师, 还是一位嵌入式软件系统开发者, 都能从本书中获益。本书为你能够将 AADL 和模型基工程直接应用于工作中提供了基础。如果你是一位技术领导或项目主管, 本书中讨论的核心原理和示例, 为你提供了指导 AADL 应用技术人员所需要的知识。

对于软件工程领域的研究生和高年级本科生来说, 本书为你根据学习经验了解和应用 AADL 提供了基础。本书能够作为嵌入式实时应用的软件体系结构或软件系统工程课程的部分教材。

为从本书中获益最大, 你需要了解什么

核心软件工程实践 (如软件体系结构、软件设计) 的基本知识、实时系统 (如并发性、调度、通信), 以及计算机运行期概念的知识 (如线程、执行语义), 将帮助你从本书中获益最多。你在这些领域应具有的专业水平, 至少应与计算机科学或软件工程领域的高年级本科生相当。如果你是一位在开发嵌入式实时软件系统方面具有 2~3 年技术学科经验的软件开发者, 你会发现本书在建模软件系统体系结构方面特别有价值。

本书的结构

我们将本书的材料组成 2 个部分, 再加上 3 个附录。第 1 部分是 AADL 语言和 MBE 实践的概述。它介绍了在一个示例的系统上下文中使用 AADL 的基本软件系统建模和分析, 包括有效地应用 AADL 的准则。第 2 部分阐述了 AADL 各种要素的特性, 包括有关使用这些要素的表示、适用性和限制。附录包括作为 AADL 标准的一部分而定义的 AADL 语言要素、属性的综合性示例表、本书中使用的示例系统的描述、参考目录和索引。

^① 仍在开发的附加附件于 2012 年稍后表决, 它们是: 错误模型附件标准的修订版、要求定义和分析附件以及代码生成附件。——原书注

术语

AADL 是一个用以区分各种组件接口规范（组件类型说明）、组件实现蓝图（组件实现说明）以及组件实例（分组件说明）之间的区别的组件建模语言。组件类型和组件实现指的是组件分类符。AADL 还区分了具有特定语义的组件类别之间的区别，以实现应用软件（如线程、进程、数据）、执行平台（如处理器、总线、设备）和复合组件（系统）建模。AADL 标准文件使用诸如系统类型说明（System Type Declaration）或系统实现说明（System Implementation Declaration）的术语。在本书中，凡上下文构成意义明确之处，我们均使用系统类型（System Type）或系统（System）一类的简短术语。

示例应用系统

我们使用汽艇自动驾驶仪（PBA）控制系统作为全书示例的基础。PBA 是用于海上船只的速度、导航和制导控制的嵌入式实时系统。但是，PBA 是为提供一个验证 AADL 的背景而建立的杜撰的发明，它并不代表任一特定的民用、军用或研究系统。当 PBA 作为一个海上应用示例时，它代表一个大范围应用的飞行器（包括飞机、航天飞机、汽车和其他陆地车辆）控制的关键要素。^①

作者介绍

彼得·H. 费勒博士（Dr. Peter Feiler）是软件工程研究所（Software Engineering Institute, SEI）研究技术和系统整体方案（RTSS）项目技术团队的资深成员。他在 SEI 已工作了 27 年。他的兴趣包括安全关键性嵌入式实时系统的体系结构中心工程。他正在与卡内基-梅隆（Carnegie Mellon）综合大学和其他研究机构的研究人员合作，开发模型基体系结构技术，并且正在调查这种技术对于民用工业的适用性。他是 SAE 国际体系结构分析和设计语言（AADL）标准的作者和编者。彼得具有卡内基-梅隆综合大学授予的计算机科学博士学位，并且是 ACM、IEEE 和 SAE 国际的资深成员。他最近获得了卡内基信息技术科学奖。

戴维·P. 格鲁茨博士（Dr. David P. Gluch）是安柏瑞德（Embry-Riddle）航空航天大学的电气、计算机、软件和系统工程系的教授，也是软件工程研究所（SEI）的访问学者。他的研究重点是软件验证的复杂系统模型基软件工程的技术和实践。在与安柏瑞德大学的教师合作之前，他曾经是 SEI 技术团队的资深成员，参与了创新软件工程实践和技术的开发与转换。他的工业研究和开发经验涵盖了容错计算机、飞机电传（Fly-By-Wire）控制、航天飞机软件建模和自动过程控制系统等众多领域。他是一本关于实时 UNIX 系统的书的合著者，并且撰写了大量的技术报告和专业论文。戴维具有佛罗里达州立大学授予的物理学博士学位，并且是 IEEE 的资深成员。

致谢

我们感谢帮助本书成为现实的许多人。

^① PBA 系统的细节在附录 A 中提供。——原书注

我们感谢使 AADL 成为现实的 SAE AADL 委员会主席布鲁斯·刘易斯 (Bruce Lewis)。每季度一次的标准会议为用户提供了一个反馈 AADL 使用意见的论坛。另外, 还感谢各委员会成员, 特别是那些帮助 AADL 形成满足实际需要的语言的、来自各工业部门的成员。撰写一本将 AADL 用于模型基工程的书籍的灵感, 正是在这种背景下产生的。

我们还欣赏许多大学和工业界的研究和先进技术团体所进行的工作, 是他们将 AADL 作为一个平台用于范围广泛的正规软件相关系统的分析之中, 并且验证了使用 AADL 的模型基工程的可行性。他们的工具和技术使得遵从 AADL 的体系结构分析得以展现它的优点。

在软件工程研究所 (SEI), 还要感谢我们的项目经理崔西娅·奥本多夫 (Tricia Oberndorf) 和琳达·诺思罗普 (Linda Northrop), 他们使我们能将时间和精力集中于编写本书的尝试之中, 并鼓励我们完成编写工作。SEI AADL 团队的其他成员: 卢茨·布拉吉 (Lutz Wrage)、亚伦·格林豪斯 (Aaron Greenhouse)、约翰·赫达克 (John Hudak)、约瑟夫·塞贝尔 (Joseph Seibel)、戴奥·德尼茨 (Dio DeNiz) 和克雷格·迈耶斯 (Craig Meyers), 以各种形式贡献了使用 AADL 的基本知识, 其中一小部分业已反映在本书中。他们引导并对建立和使用 OSATE 的工具集做出贡献, 还开发和介绍了有关 AADL 的若干辅导材料和两门课程, 并将 AADL 用于两个客户项目。我们还收到了本书几版草案的反馈。

我们欣赏几位外部评注家, 特别是布鲁斯·刘易斯 (Bruce Lewis)、杰罗姆·哈古斯 (Jérôme Hugues) 和奥莱格·索科尔斯基 (Oleg Sokolsky) 对于本书草案的反馈。最后, 我们还要感谢艾迪生维سر理出版公司 (Addison - Wesley) 的彼得·戈登 (Peter Gordon) 和金·波德埃蒙 (Kim Boedigheimer) 对于本书成功出版提供的帮助。

译者的话

传统的制造后测试型系统设计方法，在构建当今的嵌入式、软件依赖性系统十分费时费力，而且成本昂贵。其结果是有超过 30 家主要的工业机构联合 SAE，开发了 SAE AS5506 系列标准，即体系结构分析和设计语言（AADL），用于在模型基工程的分析实践中，进行系统体系结构的建模和模型转换，并进而借助系统设计模型实现系统的软件验证。也就是说，利用 AADL 可以通过建模，对诸如性能、可信性、系统安全和数据完整性一类的关键性实时因素，进行简练而严格的分析，并能够在用户工程环境下集成额外建立的客户分析/规范技术，开发完全统一的体系结构模型，从而使得用户能够更容易地构建满足其要求的可靠系统。

众所周知，模型是客观事物的抽象表示。而模型基工程是通过使用规范语言构建的系统模型，完成包括软件密集型系统的分析、设计、构建、演示验证、使用和维护等各项活动在内的工程开发。为便于理解，我们用图 1 来说明这种基于模型构建的软件系统的建模和验证过程。

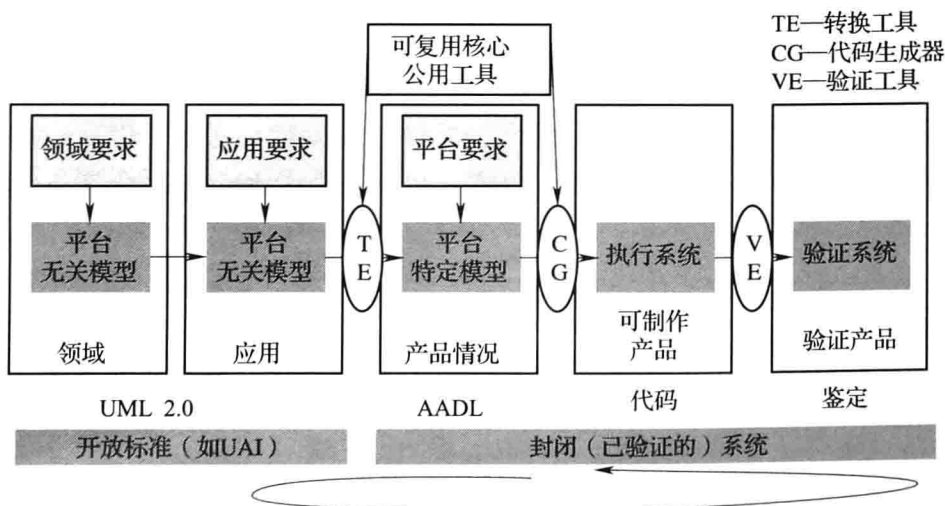


图 1 基于模型构建的软件系统的建模和验证过程

由图 1 可见，领域要求无关或应用要求无关的 PIM（平台无关模型），通常是利用 UML/xUML 构建的，而平台要求相关的 PSM（平台特定模型），则需在利用软件工具转换 PIM 的基础上，结合 AADL 构建。需要指出，图中的转换工具和代码生成器也是两个很关键的环节。目前，有不少软件开发商已开发出作为货架商品的这类模型自动转换工具和代码自动生成器。此外，这两种模型之间的转换过程是一个可逆的过程。也就是说，如果新的可执行系统没有达到规定功能和性能要求（包括互用性要求），也可将执行系统的执行代码逆向转换为 PSM，乃至 PIM，再利用 UML/xUML 修改或重新构建 PIM，继而结合新系

统的特定配置文件，转换成新的 PSM，然后再继续原流程，验证系统的规定功能和性能。故而，上述转换过程实际上是一个可迭代的过程。而在 PIM 与 PSM 之间的正反向转换是可利用转换工具自动执行的，需要指出，上述 UML/xUML 与 AADL 建模语言的区别是：UML/xUML 只是一种纯软件的建模方法；而 AADL 所建立的模型中则既包含了软件要素（软件组件），也包含了硬件要素（硬件组件）。故而，前者所建立的模型只是一个从概念数据视角和逻辑数据视角构建的平台无关的抽象。而后者所建立的模型则是一个从平台数据视角构建的平台特定的抽象。这是一种需经不断细化和迭代的建模和模型转换的过程。

《使用 AADL 的模型基工程——SAE 体系结构分析和设计语言入门》一书，是利用 AS 5506《体系结构分析和设计语言》这一国际标准，使得特定应用系统的开发和集成过程最佳化的首个指南。该书由该标准主编和技术领导彼得·H. 费勒（Peter H. Feiler）合著。这本入门性的参考和教材对于自学和课堂教学都是理想的，而且对于各类从业者，包括体系结构师，开发、集成、验证和认证人员，高层技术领导以及项目管理人员来说，本书都是一本绝佳的参考书。为在早期发现充满真实世界实例的嵌入式软件系统的问题，以降低解决问题的成本，作为体系结构中心的模型基工程方法的一部分，本书介绍了 AADL 表示的所有问题。纵观全书，作者将 AADL 与其他建模标记法和方法进行了比较，同时通过一个完整的案例研究说明了这一语言，并通过反复的细化和分析，对一个真实的示例系统进行了开发和分析。

本书的第 1 部分介绍了 AADL 语言和核心的模型基工程实践，在一个示例的系统内容中讲解了基本软件系统建模和分析，并提出了有效地应用 AADL 的实践指南。本书的第 2 部分描述了每个 AADL 要素的特性，包括它们的表示、适用性和应用限制。附录包括纳入 AADL 标准的 AADL 各要素、属性的综合性的示例表，以及本书示例的系统描述。

本书是市面上可获得的学习 AADL 的唯一教材。从头到尾地学习作者的全部工作经验与构建模型的经验，你肯定能学会 AADL 语言。因此，译者也向想要学习 AADL 的读者强烈推荐本书。

光电控制技术重点实验室

2014 年 1 月于洛阳

目 录

引言.....	(1)
---------	-------

第 1 部分 模型基工程与 AADL

第 1 章 模型基软件系统工程.....	(5)
1.1 MBE 和软件系统工程	(5)
1.1.1 MBE 用于嵌入式实时系统	(5)
1.1.2 可分析的模型和 MBE	(6)
1.1.3 MBE 和 AADL	(8)
1.2 AADL 和其他建模语言	(9)
1.2.1 AADL、MDA 和 UML	(10)
1.2.2 AADL 和 SysML	(11)
第 2 章 使用 SAE AADL 进行工作	(12)
2.1 AADL 模型	(13)
2.1.1 组件类别	(13)
2.1.2 语言的语法	(14)
2.1.3 AADL 分类	(14)
2.1.4 AADL 说明汇总	(15)
2.1.5 AADL 模型的结构	(17)
2.2 系统规范和系统实例	(18)
2.2.1 组建系统实例模型	(18)
2.2.2 AADL 文本域图形表示	(18)
2.2.3 分析模型	(20)
第 3 章 使用 AADL 建模和分析的基础	(21)
3.1 简单模型的开发	(21)
3.1.1 模型组件的定义	(21)
3.1.2 顶层模型的开发	(24)
3.1.3 控制软件的细节描述	(25)
3.1.4 添加硬件组件	(27)
3.1.5 物理连接的说明	(28)
3.1.6 软件对硬件的绑定	(29)

3.1.7 进行调度分析	(30)
3.1.8 综述	(32)
3.2 代码文件的表示	(32)
3.2.1 源代码和二进制文件的编写	(33)
3.2.2 变量名称文件的编写	(34)
3.2.3 源代码结构的建模	(34)
3.3 动态重构的建模	(35)
3.3.1 PBA 模型的扩展	(35)
3.3.2 模式的规定	(37)
3.4 抽象流的建模与分析	(38)
3.4.1 流规范的规定	(38)
3.4.2 端对端流的规定	(39)
3.4.3 流分析	(40)
3.5 概念模型的开发	(40)
3.5.1 将抽象组件部署在 PBA 模型中	(41)
3.5.2 详细的抽象实现	(43)
3.5.3 变换为运行时表示	(44)
3.5.4 增加运行时属性	(45)
3.5.5 完善规范	(47)
3.6 利用组件模式开展的工作	(48)
3.6.1 组件库和参考体系结构	(48)
3.6.2 建立组件库	(49)
3.6.3 定义参考体系结构	(51)
3.6.4 使用参考体系结构	(52)
第4章 应用AADL能力	(54)
4.1 规定系统组成	(54)
4.1.1 组件分层	(54)
4.1.2 建模执行平台资源	(55)
4.1.3 执行平台通信支持	(56)
4.1.4 系统分层	(56)
4.1.5 建立系统实例模型	(57)
4.1.6 使用系统实例模型中的连接开展工作	(57)
4.1.7 使用系统实例模型开展工作	(58)
4.2 组件交互	(59)
4.2.1 数据和控制单向交换的建模	(59)
4.2.2 建模共享数据交换	(60)
4.2.3 建模局部服务请求或功能调用	(61)
4.2.4 建模远程服务请求或功能调用	(62)

4.2.5 建模面向对象的方法调用	(64)
4.2.6 建模子程序参数	(66)
4.2.7 与外部世界的接口	(68)
4.3 数据及其使用建模	(68)
4.3.1 定义一个简单的数据类型	(68)
4.3.2 表示数据类型的变型	(69)
4.3.3 数据类型详解	(70)
4.4 组织一个设计	(71)
4.4.1 软件包使用	(71)
4.4.2 开发替代的实现	(73)
4.4.3 定义多种扩展	(74)

第2部分 AADL 的要素

第5章 定义 AADL 组件	(79)
5.1 组件名称	(79)
5.2 组件类别	(79)
5.3 说明组件类型	(80)
5.4 说明组件的外部接口	(82)
5.5 说明组件实现	(84)
5.6 汇总	(87)
第6章 软件组件	(88)
6.1 线程	(88)
6.1.1 表示	(90)
6.1.2 属性	(90)
6.1.3 限制	(91)
6.2 线程组	(92)
6.2.1 表示	(92)
6.2.2 属性	(93)
6.2.3 限制	(93)
6.3 进程	(94)
6.3.1 表示	(94)
6.3.2 属性	(95)
6.3.3 限制	(95)
6.4 数据	(96)
6.4.1 表示	(97)
6.4.2 属性	(97)
6.4.3 限制	(97)

6.5	子程序	(98)
6.5.1	表示	(99)
6.5.2	属性	(99)
6.5.3	限制	(99)
6.6	子程序组	(100)
6.6.1	表示	(100)
6.6.2	属性	(101)
6.6.3	限制	(101)
第 7 章	执行平台组件	(102)
7.1	处理器	(102)
7.1.1	表示	(103)
7.1.2	属性	(103)
7.1.3	限制	(104)
7.2	虚拟处理器	(104)
7.2.1	表示	(104)
7.2.2	属性	(105)
7.2.3	限制	(105)
7.3	存储器	(106)
7.3.1	表示	(106)
7.3.2	属性	(107)
7.3.3	限制	(107)
7.4	总线	(108)
7.4.1	表示	(108)
7.4.2	属性	(109)
7.4.3	限制	(109)
7.5	虚拟总线	(109)
7.5.1	表示	(109)
7.5.2	属性	(110)
7.5.3	限制	(110)
7.6	设备	(110)
7.6.1	表示	(111)
7.6.2	属性	(112)
7.6.3	限制	(112)
第 8 章	复合组件和通用组件	(113)
8.1	系统	(113)
8.1.1	表示	(113)
8.1.2	属性	(114)
8.1.3	限制	(114)

8.2 抽象	(115)
8.2.1 表示	(115)
8.2.2 属性	(116)
8.2.3 限制	(116)
第9章 静态和动态体系结构	(118)
9.1 分组件	(118)
9.1.1 说明分组件	(118)
9.1.2 使用分组件说明	(119)
9.1.3 作为阵列说明的分组件	(120)
9.2 模式	(121)
9.2.1 说明模式和模式转换	(121)
9.2.2 说明模态组件类型和实现	(122)
9.2.3 使用可选组件配置的模式	(123)
9.2.4 继承模式	(125)
9.2.5 模式特定的属性	(127)
9.2.6 调用序列的模态配置	(127)
第10章 组件交互	(129)
10.1 端口和连接	(129)
10.1.1 端口说明	(129)
10.1.2 端口与端口连接的说明	(131)
10.1.3 端口与端口连接的使用	(131)
10.1.4 端口与端口连接的限制	(134)
10.1.5 端口通信时序	(136)
10.1.6 数据流的采样处理	(137)
10.1.7 确定性采样	(138)
10.1.8 混合端口基和共享数据通信	(141)
10.1.9 端口与端口的连接属性	(144)
10.1.10 集合数据通信	(144)
10.2 数据访问和连接	(146)
10.3 总线访问和连接	(148)
10.4 特征组和连接	(151)
10.4.1 特征组类型说明	(152)
10.4.2 将特征组作为组件特征来说明	(153)
10.4.3 特征组连接说明	(154)
10.5 抽象特征和连接	(157)
10.5.1 抽象特征说明	(157)
10.5.2 抽象特征细化	(158)
10.6 阵列和连接	(159)

10.6.1	明确规定的阵列连接	(159)
10.6.2	阵列连接模式	(160)
10.6.3	使用阵列连接属性	(161)
10.7	子程序调用、访问和实例	(162)
10.7.1	说明调用和调用序列	(163)
10.7.2	将远程子程序调用作为绑定来说明	(164)
10.7.3	将远程子程序调用作为访问连接来说明	(165)
10.7.4	建模子程序实例	(166)
10.8	参数连接	(168)
10.8.1	参数说明	(168)
10.8.2	参数连接的说明	(169)
第 11 章	系统流与软件开发	(171)
11.1	流	(171)
11.1.1	流规范说明	(171)
11.1.2	流实现说明	(173)
11.1.3	端对端流说明	(177)
11.1.4	端对端流处理	(178)
11.2	软件与硬件的绑定	(179)
11.2.1	绑定属性的说明	(179)
11.2.2	处理器绑定	(181)
11.2.3	存储器绑定	(181)
11.2.4	连接绑定	(181)
11.2.5	远程子程序调用的绑定	(182)
第 12 章	组建模型	(184)
12.1	模型要素的命名和引用	(184)
12.1.1	软件包的命名和引用	(184)
12.1.2	分类符的命名和引用	(184)
12.1.3	对模型要素的引用	(185)
12.1.4	属性集的命名和引用	(185)
12.2	利用软件包组建模型	(186)
12.2.1	软件包说明	(186)
12.2.2	软件包中引用的要素	(188)
12.2.3	用于软件包和类型参考的别名	(190)
12.3	通过细化改进模型	(191)
12.3.1	分类符扩展说明	(192)
12.3.2	模型要素细化说明	(192)
12.3.3	用于细化的分类符替换规则	(194)
12.3.4	类别的细化	(196)

12.4 作为分类符参数的原型	(197)
12.4.1 原型说明	(197)
12.4.2 原型使用	(199)
12.4.3 提供原型实际	(200)
12.4.4 属性	(202)
第13章 注释模型	(203)
13.1 编写模型要素文件	(203)
13.1.1 评注并描述属性	(203)
13.1.2 空白组件段	(204)
13.2 使用属性	(204)
13.2.1 分配属性值	(205)
13.2.2 AADL 属性类型和数值	(206)
13.2.3 确定属性值	(208)
13.2.4 包容属性关联	(209)
13.2.5 确定属性值：一个示例	(210)
第14章 扩展的语言	(213)
14.1 属性集	(213)
14.1.1 说明的属性集	(213)
14.1.2 属性集说明	(214)
14.1.3 属性定义	(217)
14.1.4 属性常数说明	(219)
14.2 附件子语言	(219)
14.2.1 附件库中说明的附件概念	(220)
14.2.2 分子句中使用的附件概念	(221)
第15章 创建和验证模型	(222)
15.1 模型创建	(222)
15.2 模型创建工具	(223)
15.3 系统验证和生成	(224)
15.4 系统验证和生成工具	(225)
附录A 语法和属性汇总	(227)
A.1 AADL 语法	(227)
A.2 组件类型和实现要素	(240)
A.3 基本属性类型和类型构造器	(245)
A.4 AADL 备用字	(245)
A.5 AADL 属性	(246)
A.5.1 部署属性	(246)
A.5.2 线程相关属性	(256)
A.5.3 时序属性	(260)

- A. 5. 4 通信属性 (267)
 - A. 5. 5 存储器相关属性 (272)
 - A. 5. 6 编程属性 (276)
 - A. 5. 7 建模属性 (283)
 - A. 5. 8 项目特定常数和属性类型 (284)
- A. 6 运行时服务 (289)
 - A. 6. 1 应用运行时服务 (289)
 - A. 6. 2 运行时执行服务 (291)
- A. 7 汽艇自动驾驶仪系统 (294)
 - A. 7. 1 描述 (294)
 - A. 7. 2 增强版 PBA 系统 (295)
 - A. 7. 3 PBA 系统的 AADL 组件 (295)
 - A. 7. 4 一个备选的 AADL 表示法 (296)
- 附录 B 增补资源 (297)
 - B. 1 系统体系结构建模 (297)
 - B. 2 案例研究 (298)
- 附录 C 参考文献 (300)