ANSYS原厂策划　万水精心出品

ANSYS核心产品系列

ANSYS®

万水ANSYS技术丛书

ANSYS SCADE Suite 建模基础

荆　华　沈轶烨　等编著

中国水利水电出版社
www.waterpub.com.cn

万水 ANSYS 技术丛书

ANSYS SCADE Suite 建模基础

荆　华　沈轶烨　等编著

中国水利水电出版社
www.waterpub.com.cn
·北京·

内 容 提 要

SCADE 产品模块众多，适用于安全关键领域的嵌入式系统和软件的研制，涵盖功能安全分析、系统架构设计、控制算法设计、人机界面设计、多学科仿真应用等多个方面。本书主要讲解其中的控制算法设计软件 SCADE Suite，并重点围绕三个角度进行编写：从初学者的角度出发，循序渐进地安排内容结构和知识点分布；从使用者的角度出发，介绍 SCADE 的基本使用方法和常用操作技巧；从工程人员的角度出发，讲述有代表性的实例、介绍通用的行业经验。

本书收录了大量有较强代表性的示例，示例中的模型都力求准确无误，可以在 PC 环境下仿真运行。

本书可作为理工科高校计算机类研究生、航空宇航类研究生、导航制导与控制类研究生以及高年级本科生的教学用书，也可供相关领域的师生、科研和工程技术人员参考。

本书所有示例和练习题参考答案可以从万水书苑以及中国水利水电出版社网站下载，网址为：http://www.wsbookshow.com 和 http://www.waterpub.com.cn/softdown/。

图书在版编目（CIP）数据

ANSYS SCADE Suite建模基础 / 荆华等编著. -- 北京 : 中国水利水电出版社, 2018.8
（万水ANSYS技术丛书）
ISBN 978-7-5170-6803-7

Ⅰ. ①A… Ⅱ. ①荆… Ⅲ. ①计算机仿真－系统建模－应用软件 Ⅳ. ①TP391.92

中国版本图书馆CIP数据核字(2018)第202138号

策划编辑：杨元泓　　责任编辑：张玉玲　　封面设计：李　佳

书　名	万水 ANSYS 技术丛书 ANSYS SCADE Suite 建模基础 ANSYS SCADE SUITE JIANMO JICHU
作　者	荆　华　沈铁烨　等编著
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路 1 号 D 座　100038） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（营销中心）、82562819（万水）
经　售	全国各地新华书店和相关出版物销售网点
排　版	北京万水电子信息有限公司
印　刷	三河市鑫金马印装有限公司
规　格	184mm×260mm　16 开本　23.5 印张　582 千字
版　次	2018 年 8 月第 1 版　2018 年 8 月第 1 次印刷
印　数	0001—3000 册
定　价	82.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

版权所有 · 侵权必究

前　言

SCADE 诞生于 20 世纪 80 年代的法国，从欧洲的航空业与核能业的工程应用起步，经过 30 多年的发展，逐渐成为在航空航天、国防军工、轨道交通、核能重工、汽车电子等行业具有广泛应用的商业产品。

由于 SCADE 专注于流程规范、标准严苛的安全关键行业，行业的特性使得其多应用于研制具有相当密级的、高难度的重大项目，因此 SCADE 在国内仍处于“养在深闺人未识”的状况。时至今日，市场上鲜有中文版的 SCADE 书籍可供大家参考学习，SCADE 的推广应用也就有些“高处不胜寒”了。

在业内众多基于模型的研制工具中，作为唯一在多个安全关键行业中以开发工具形式通过最高等级鉴定的产品，SCADE 的独特优势逐渐被越来越多的国内企业认可。随着 SCADE 在包括 C919 大型客机、高铁列控系统、第三代核电仪控系统、军工先进战机等重大高端装备项目中的开展应用，业界亟需 SCADE 的相关培训教程。

“要写怎样的一本 SCADE 书使它更适合于初学者呢？”

这是我们写作本书时反复问自己的一个问题。SCADE 产品模块众多，适用于安全关键领域的嵌入式系统和软件的研制，涵盖功能安全分析、系统架构设计、控制算法设计、人机界面设计、多学科仿真应用等多个方面。本书主要介绍控制算法设计软件 SCADE Suite，并重点围绕三个角度进行讲解：

- 从初学者的角度出发，循序渐进地安排内容结构和知识点分布。
- 从使用者的角度出发，介绍 SCADE 的基本使用方法和常用操作技巧。
- 从工程人员的角度出发，讲述有代表性的实例，介绍通用的行业经验。

本书收录了大量有较强代表性的示例，示例中的模型都力求准确无误，可以在 PC 环境下仿真运行。

作为 ANSYS 公司 SCADE 产品的高校合作计划伙伴之一，我们电子科技大学不揣冒昧，在已经做过三年课堂教学的基础上，结合业内多家 SCADE 用户的经验，编写了这本 SCADE 软件基础教程。希望借此抛砖引玉，吸引更多的专家学者加入到推广这款优秀产品的队伍中来。

本书可作为理工科高校计算机类研究生、航空宇航类研究生、导航制导与控制类研究生以及高年级本科生的教学用书，也可供相关领域的师生、科研和工程技术人员参考。

本书章节介绍

读者可以按部就班地顺序阅读全书章节，系统地学习 SCADE Suite。有使用经验的读者，也可以根据目录直接到相应章节查询相关内容。

第 1 章主要介绍 SCADE 产品的背景、特点和未来的发展方向。本章包含一个简单的 Getting Start 案例，让读者快速上手。

第 2 章主要介绍 SCADE Suite 的界面布局、基本语法和常用建模操作。学习完本章内容后，可进行以数据流为基础的建模操作。

第 3 章主要介绍 SCADE Suite 中的状态机建模操作。SCADE 的状态机又称安全状态机（Safe State Machine，SSM），适用于精确描述具有抢占、并发、同步等复杂迁移转换特性的控制流。

第 4 章主要介绍 SCADE Suite 的高级建模操作，主要包括数组建模操作、结构体建模操作、迭代器建模操作、条件激活操作、多态建模操作。

第 5 章主要介绍基于 SCADE Suite 模型的验证，阐述使用 SCADE 模型后对机载软件验证流程的简化及对应的活动项，内容包括模型仿真、模型覆盖分析、认证级测试环境、形式化验证和编译器验证。

第 6 章主要介绍 SCADE Suite 代码与其他目标的生成，包括模型生成代码的配置、生成代码在PC平台的集成、可结合其他第三方工具的文件生成与模型对应的详细设计文档的生成。

第 7 章主要介绍 SCADE Suite 模型的优化操作，包括模型优化的目标、基准和推荐方法，并介绍如何使用 AbsInt 工具进行优化结果分析。

第 8 章主要介绍 SCADE Suite 模型开发中的项目管理，包括项目组织、配置管理、追踪管理和建模规范。

第 9 章综合案例，描述取中位数的算法。

附录是 SCADE Suite 关于 DO-178C/DO-331 目标的符合性矩阵。

致谢

任何一部书的成功出版都离不开多方面的努力。感谢参与本书编写的陈小平、张程灏、邢多庆、邱晓晗、吴丹杨、杨坤、王喆、毛伟、王文杰、姜强等人，他们为本书的模型示例设计、章节校对等工作耗费了很多业余时间；感谢合作伙伴 ANSYS 公司大中国区系统事业部的马金梭、董如怡、傅金泉、杨瑾婧、应中伟、侯东、姜平、许周文、周霄、孙晓晗、王文全，在多次的交流讨论中，他们从工程应用角度出发，结合不同行业客户的使用经验与常见问题，给出了许多有益的建议；感谢中国水利水电出版社杨元泓编辑的倾力协助，她为本书的顺利出版倾注了极大的心血；感谢电子科技大学航空航天学院的刘强、须玥、曾艳、梁伟等人在本书撰写过程中给予的关怀、鼓励与支持，他们的坚定支持是我们能完成本书的最大动力。最后，由衷地感谢所有参与过本书撰写和审阅的各位朋友。

由于写作团队水平有限及时间仓促，书中纰漏错误之处在所难免，望各位读者不吝赐教，以便再版时我们采纳读者的宝贵建议修正不足，我们的电子邮箱是 ScadeBasicTextBook@126.com。

试读结束：需要全本请在线购买： www.ertongbook.com

目　　录

1 开启 SCADE 之旅

1.1 背景概念简介

1.1.1 嵌入式系统

嵌入式系统，是一种嵌入机械或电气系统内部、具有专一功能和实时计算性能的计算机系统。区别于可以执行多重任务的通用型计算机，嵌入式系统是为某些特定任务而设计的。有些嵌入式系统的需求强调尽可能地简单、低成本，有些则要求满足非常高的实时性要求。一般来说，嵌入式系统通常具有内核小、专用性强、系统精简、实时性高等特点。

1.1.2 安全关键系统

当产生故障或者失效后，能引起下述情况的系统可认为是安全关键的系统，如图 1-1 所示。

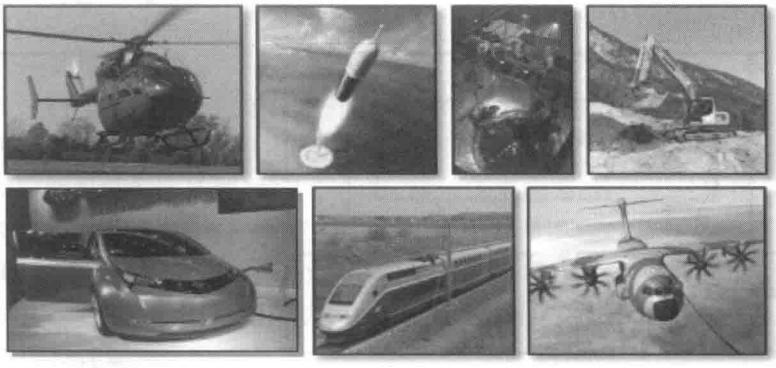

图 1-1 安全关键系统

- 人身的重大损伤，甚至死亡：例如航空、高铁、地铁、汽车……
- 环境的大范围恶化：例如核电、化工、炼油……

- 财产的巨额损失：例如通信、电力、重工……
- 关键任务的耽误和失败：例如航天、军事、指挥……

安全关键软件（Safety Critical Software）的定义带有主观性。IEEE 定义安全关键软件为：用于一个系统中，可能导致不可接受的风险的软件[1]。为了指导和规范企业开发出安全关键软件，相关行业提出了专门的软件安全性工程标准，包括机载软件的适航标准 DO-178C、铁路领域的 EN 50128 标准、电子电气领域的 IEC 61508、汽车领域的 ISO 26262 等。

SCADE 的英文全称为 Safety Critical Application Development Environment，意思是安全关键的应用开发环境，是一款可通过多个安全关键行业软件工程标准认证的工具，适用于开发嵌入式安全关键系统的软件。由于机载软件适航标准较为成熟和严苛，是在航空领域普遍认可和接受的标准。而 SCADE 产品在航空领域的应用也更加知名和广泛，本书将主要遵循机载软件的适航标准来介绍 SCADE 产品。

1.1.3 机载软件的适航标准

1. DO-178C 概览

机载软件适航标准的第一版 DO-178 发布于 1982 年。历经 30 多年的发展，相继在 1985 年发布了 DO-178A，1992 年发布了 DO-178B，2011 年发布了 DO-178C。

值得一提的是，机载软件的适航标准不是单独存在的，它和 ARP 4754A、ARP 4761、ARP 5150、DO-254、DO-297 等文件一起构成了现代机载系统安全性设计与评估的一组指南材料[2]，如图 1-2 所示。

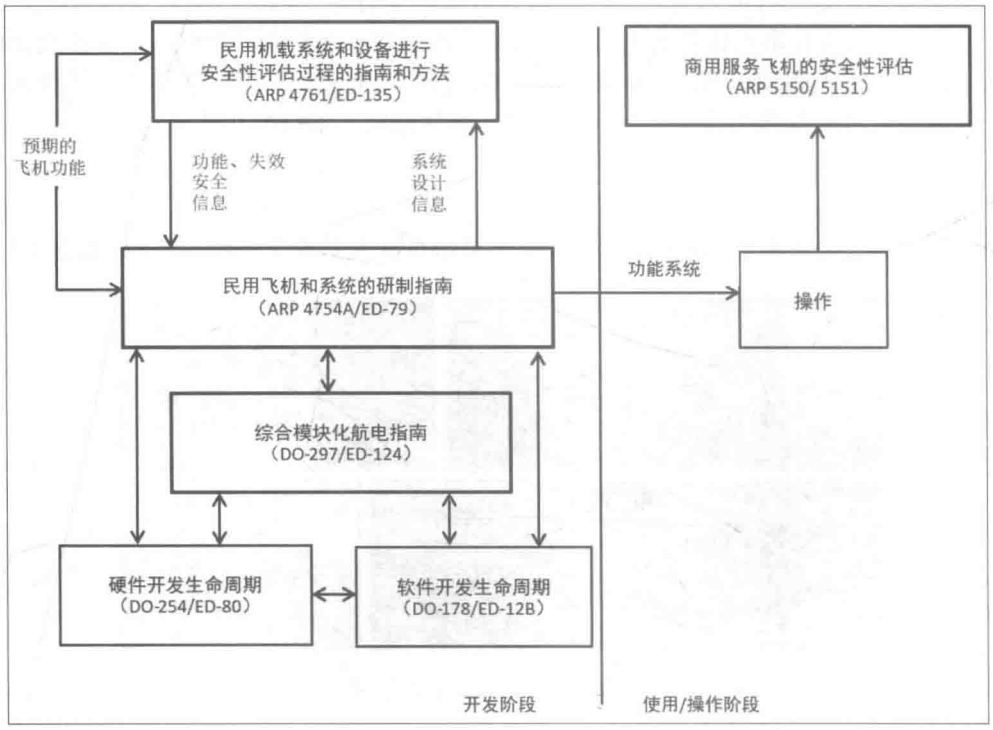

图 1-2　现代机载系统安全性设计与评估的一组指南材料

ARP 4761 和 ARP 4754 通过对航空器、系统和功能不同层级的安全分析，得出危害和失效条件，并以此分配出待研制软硬件的研制保证等级（DAL），如表 1-1 所示。

表 1-1 待研制软硬件的研制保证等级

研制保证等级	危害等级
A	灾难性的
B	危险的
C	重要的
D	次要的
E	无影响的

DO-178C 就是机载软件开发生命周期中对不同研制等级软件进行安全开发的指南，这些指南的绝大多数也都适用于其他的安全关键领域。DO-178C 的名称是《机载系统和设备合格审定的软件考虑》。它是在美国航空无线电技术委员会（RTCA）和欧洲民用航空设备组织（EUROCAE）的发起下，由国际团体形成共识，从而产生的一组建议，于 2011 年 12 月 13 日发布。DO-178C 有 7 份文件，分别是：

- DO-178C：机载系统和设备合格审定中的软件考虑。
- DO-278A：通信、导航、监视和空中交通管理系统软件完整性保证指南。
- DO-248C：DO-178C 和 DO-278A 的支持信息。
- DO-330：软件工具鉴定考虑。
- DO-331：DO-178C 和 DO-278A 的基于模型的开发与验证补充。
- DO-332：DO-178C 和 DO-278A 的面向对象技术与相关技术补充。
- DO-333：DO-178C 和 DO-278A 的形式化方法补充[1]。

2. DO-178C 的过程简介

DO-178C 标准是面向过程和目标的，通过以下 3 种形式指导软件开发者的工作：

- 规定航空机载软件生命周期中各个过程的目标。
- 推荐达到这些目标的活动和工程实现考虑。
- 规定确认这些目标已经实现的证据记录[2]。

DO-178C 中的过程主要分为以下 3 类：

- 软件计划过程：用于指导软件开发过程和软件综合过程。DO-178C 中规定了需要准备的 5 个计划（软件合格审定计划、软件开发计划、软件验证计划、软件配置管理计划和软件质量保证计划）和 3 个标准（软件需求标准、软件设计标准和软件编码标准）。
- 软件开发过程：分为软件需求过程、软件设计过程、软件编码过程和软件集成过程 4 个子过程，如图 1-3 所示。图中的高层需求（HLR）和低层需求（LLR）是 DO-178 系列标准特有的提法，可粗略地将它们类比为传统软件工程中的概要设计和详细设计。
- 软件综合过程：分为软件验证过程、软件配置管理过程、软件质量保证过程和审定联络过程 4 个子过程。

试读结束：需要全本请在线购买： www.ertongbook.com

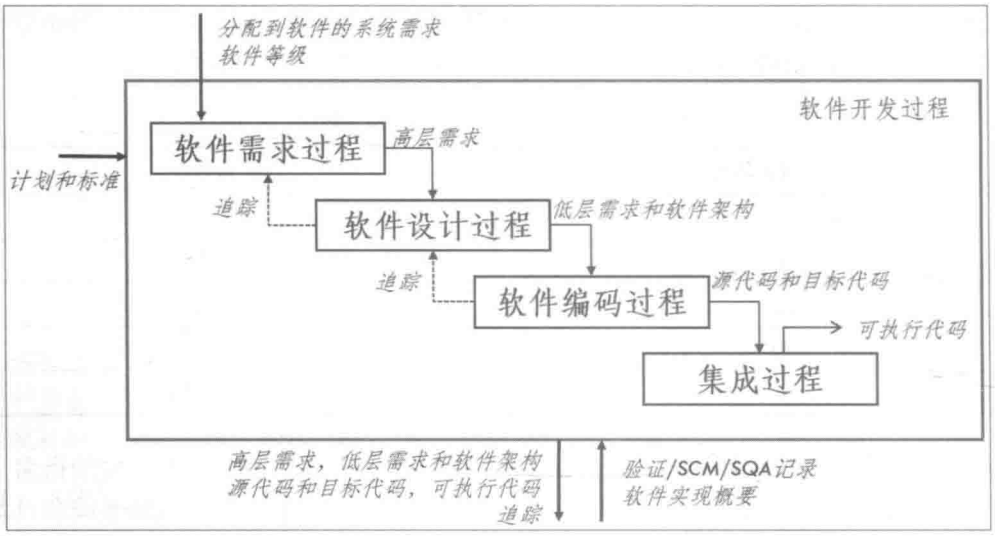

图 1-3　软件开发过程

三类过程的关系如图 1-4 所示，它们不是独立地进行的，过程之间存在着交互，例如软件综合过程贯穿于整个软件生命周期的始终，与软件计划过程和软件开发过程都存在着交互[2]。

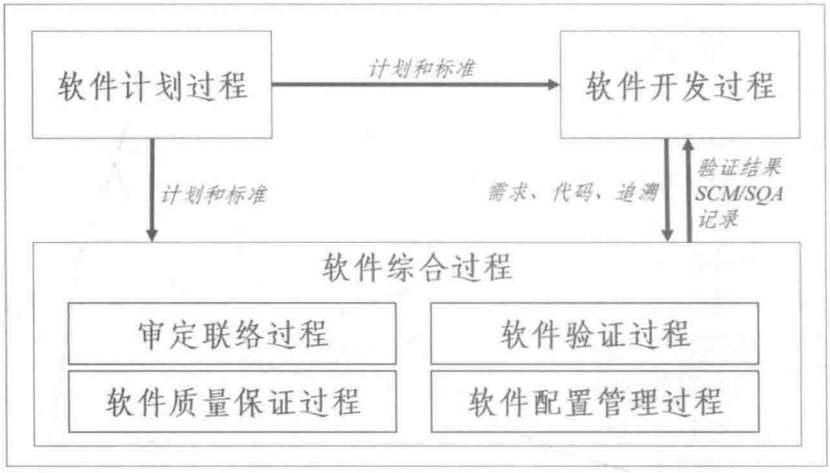

图 1-4　DO-178C 中的三类过程关系图

3. DO-178C 的目标简介

软件无法作为一种唯一且独立的产品进行审定，它必须要和安装在航空器、发动机上的机载系统和设备一起接受适航审定当局的审定。作为审定工作的一部分，机载系统或设备中的嵌入式软件应得到审定当局的批准通过[2]。

DO-178C 标准以目标来定量刻画机载软件的研制，只要在研发过程中实现了标准中规定的某一级别软件的全部目标，就可以有理由相信该软件满足了适航审定当局的适航性要求。

DO-178C 标准中，A 级软件要实现 71 个目标，B 级软件要实现 69 个目标，C 级软件要实现 62 个目标，D 级软件要实现 26 个目标[1]。其中，这些级别软件的目标中都含有共同的部分，只是随着软件级别严格程度的提高，目标数和独立性的要求相应增加，完成的工作量和难

度也逐渐提高[2]。

以最高安全等级的 A 级软件为例，DO-178C 结合三类过程提出了 10 组目标及完成这些目标推荐进行的指南和活动。这 10 组目标是：

- A-1 软件计划过程（7 个目标）
- A-2 软件开发过程（7 个目标）
- A-3 软件需求过程输出的验证（7 个目标）
- A-4 软件设计过程输出的验证（13 个目标）
- A-5 软件编码和集成过程输出的验证（9 个目标）
- A-6 集成过程输出的测试（5 个目标）
- A-7 验证过程结果的验证（9 个目标）
- A-8 软件配置管理过程（6 个目标）
- A-9 软件质量保证过程（5 个目标）
- A-10 合格审查联络过程（3 个目标）

4. DO-330 的简介

DO-178C 的目标不可省略，但是完成目标的活动通过工具的介入可以省略或者自动化地执行。但这样的工具必须满足 DO-178C 的补充文件 DO-330 软件工具鉴定考虑的要求。

DO-330 中引入了工具鉴定准则和工具鉴定级别的概念。

- 准则 1：该工具的输出是结果软件的一部分，因此可能注入一个错误。
- 准则 2：该工具自动化了验证过程，因而可能检测不出某个错误，并且其输出的作用是去除或减少下列过程：
  - 没有被工具自动化的验证过程
  - 可能对机载软件有影响的开发过程
- 准则 3：该工具在其意图的使用范围内可能检测不出某个错误。

其中的“准则 1 工具”与 DO-178B 中的开发工具的鉴定过程非常相像。准则 1 工具的例子包括自动代码生成器、配置文件生成器、编译器、链接器等。“准则 3 工具”与 DO-178B 中的验证工具的鉴定过程基本相同。例子包括测试用例生成器、自动化测试工具、结构覆盖工具、数据耦合分析工具和静态代码分析器。而“准则 2 工具”可视为一种“超级验证工具”，是 DO-178C 中专门引入的概念[1]。

基于三个准则以及工具支持的软件级别，可为工具分配一个工具鉴定级别（Tool Qualification Level，TQL）。TQL 决定了鉴定过程中要求的严格程度。共有 5 个 TQL，TQL-1 要求最严格，TQL-5 要求最低[1]，如表 1-2 所示。

表 1-2 工具鉴定过程中要求的严格程度

工具类别	软件级别	工具鉴定级别
开发工具	A	TQL-1
开发工具	B	TQL-2
开发工具	C	TQL-3
开发工具	D	TQL-4
验证工具	E	TQL-4 或 TQL-5

对于研制不同软件级别的项目，需要选用恰当的工具。SCADE 系列产品中的代码生成器、DF 文件生成器是可通过 TQL-1 鉴定的开发工具；覆盖分析器、认证级测试环境和文档生成器是可通过 TQL-5 鉴定的验证工具。

1.1.4 基于模型的开发与验证

1. DO-331 的模型

业内模型设计开发相关的提法不少，有模型驱动的设计（Model Driven Design，MDD）、基于模型的设计（Model Base Design，MBD）、基于模型的系统工程（Model Base System Engineering，MBSE）等，侧重点各有不同。本书主要使用 DO-178C 补充文件 DO-331 中基于模型的开发与验证（Model Based Development and Verification，MBDV）的提法。

模型可以用于架构设计、硬件描述、虚拟仿真等领域，而 DO-178C 中的模型主要是用于系统和软件开发的，是软件生命周期数据的一种抽象描述方式，用来更好地支持软件开发过程和软件验证过程。软件生命周期中并非所有数据都适合用模型描述，有两类最适合用模型描述：①软件高层需求；②软件低层需求和软件架构。这两类数据可能全部用模型描述，也可以部分用模型描述。

DO-331 中引入了规范模型（Specification Models）和设计模型（Design Models）的概念。规范模型描述的是高层需求，是对软件组件的功能、性能或安全特性的抽象表达。规范模型不定义软件设计的细节。设计模型规定了软件组件的内部数据结构、内部数据流和控制流。设计模型描述了低层需求和软件架构。设计模型可用于产生代码。

并非所有模型都适合描述数据，通常下述情况不认为是模型：

- 没有严格的语法（图符、字母、文字）。
- 没有严格的语义。
- 没有用来描述软件的需求和架构。
- 没有对软件的开发和验证带来帮助的。

因此，需要引入专门的建模技术和模型标准来衡量模型的好坏。通常规定：

- 模型必须通过某种明确定义的建模符号来完整描述。
- 这种建模符号具备精准严格的语法。

业内通常选用基于 SysML、AADL 等语言的模型来描述规范模型，而本书要介绍的基于 SCADE 语言的工具则广泛应用于设计模型。值得一提的是，规范模型和设计模型是互斥的，一个模型不可能同时为规范模型和设计模型。用规范模型直接生成代码是需要商榷的。

2. MBDV 的优势

除了具有认证级的代码生成器外，SCADE 是 MBDV 工具也是其另一个重要的特点。MBDV 有许多潜在好处，收获这些好处的能力依赖于实现细节[1]。

（1）V 生命周期变为 Y 生命周期。使用 MBDV 方式的主要动因就是应用模型自动生成代码的功能，省去了工程师的手工编码。这使传统的 V 生命周期变为 Y 生命周期，带来降低开发时间、成本，甚至降低人为错误的潜力。其中，使用一个没有鉴定的代码生成器可以使流程有 20%的缩短，使用一个得到鉴定的代码生成器可以得到 50%[1]。

（2）更聚焦于需求。需求错误发现得越晚，修复的成本就越高。当用一个模型表示需求时，它比文字需求更有能力表达出系统或软件功能的清晰视图。当使用自动化来实现模型时，

使得开发团队能够聚焦于需求的图形化表示，而不是实现细节[1]。

（3）更早更有效的验证。当开发流程中确定使用经过鉴定的代码生成器时，开发焦点从代码转移到了模型。模型仿真的使用促进了及早验证，使得需求以及错误检测能更早成熟[1]。更多的基于模型验证方面的内容可以参考第 5 章的内容。

（4）减少不必要的冗余和不一致。传统的软件开发使用 3 层需求：分配给软件的系统需求、软件高层需求（HLR）、软件低层需求（LLR）。传统方法中，这些层次之间可能存在大量冗余，在进行变更时带来不一致性。使用经过鉴定的建模工具有能力降低这些冗余和不一致性[1]。

（5）纳入形式化方法的支持。建模技术、工具支持和形式化方法之间有一个汇聚，这可以增强 MBDV 的好处。当一个包含代码生成器的建模工具的正确性得到形式化验证时，就可以依赖该工具去执行其预期的功能。形式化方法可以通过验证整个输入和输出空间来提升使用“模型到代码”工具包的信心[1]。

与其他补充文件一样，DO-331 使用 DO-178C 提纲，并根据需要对 DO-178C 修改、替换或增加目标、活动和指南[1]。附录 2 列出了使用 SCADE 工具后完成满足 DO-178C/DO-331 目标的矩阵表。

3. MBDV 与传统方式的区别

表 1-3 和表 1-4 列出了在开发和验证两个层面 MBDV 方式与传统方式的区别。

表 1-3 MBDV 的开发方式与传统开发方式的区别

序号	传统的软件开发	典型的 MBDV 开发
1	高层需求（文本或图片）	规范模型
2	低层需求（文本或图片）	设计模型
3	软件架构（文本或图片）	设计模型
4	编写源代码（手工编码）	自动生成代码
5	软件需求/设计标准	软件模型标准
6	追踪关系（条目化）	追踪关系（模型单元）
7	代码库	模型库

表 1-4 MBDV 的验证方式与传统验证方式的区别

序号	传统的软件验证	典型的 MBDV 验证
1	计划和标准的评审与分析	计划和标准的评审与分析
2	高层需求的评审与分析	高层需求的评审与分析、模型仿真
3	低层需求的评审与分析	低层需求的评审与分析、模型仿真
4	软件架构的评审与分析	软件架构的评审与分析、模型仿真
5	源代码的评审与分析	源代码的评审与分析
6	集成过程输出的评审与分析	集成过程输出的评审与分析
7	测试	测试、模型仿真

续表

序号	传统的软件验证	典型的 MBDV 的验证
8	测试用例、规程及结果的评审与分析	仿真用例、规程及结果的评审与分析
9	覆盖分析	覆盖分析
10	需求覆盖分析	需求覆盖分析
11	结构覆盖分析	结构覆盖分析、模型覆盖分析

1.2 SCADE 介绍

本节介绍 SCADE 产品的起源、发展和现状。

1.2.1 同步语言介绍

SCADE 模型背后使用的是 SCADE 语言，它起源于同步语言。而同步语言（Synchronous Language）诞生于 20 世纪 80 年代早期。当时以计算机为基础的系统可分为以下 3 类：

- 转换系统（Transformational Systems）：根据输入计算输出，然后结束，如绝大部分的数值计算软件、工资管理软件、编译系统。
- 交互系统（Interactive Systems）：与外界环境进行不停的交互处理，如操作系统、数据库、互联网。
- 反应系统（Reactive Systems）：不停地响应外界环境的激励，它与交互系统的不同之处是前者纯粹是由输入驱动的，其典型例子是过程控制系统、信号处理系统。

绝大部分工业领域的实时系统（Realtime Systems）都是反应系统，另外一些则涉及通信协议或者人机界面等因素。同步语言就是以反应系统为应用领域的程序设计语言[4][5][6]。

通常反应系统有 3 个要素：输入、计算、输出，如图 1-5 所示。

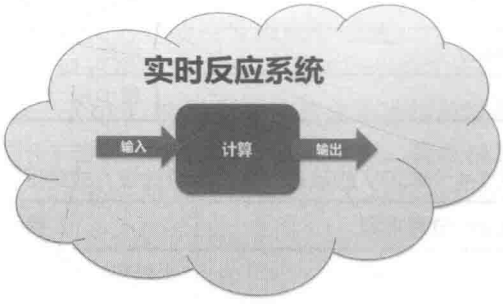

图 1-5 实时反应系统

1. 同步假设

考虑这样一种情况，如果系统处理速度无限快，即系统在一个可以忽略不计的瞬间响应输入并产生输出。这样，系统在一瞬间响应输入，然后等待下一个输入，……，那么，任何连续的两次响应之间有间隔，不会前后重叠，如图 1-6 所示。

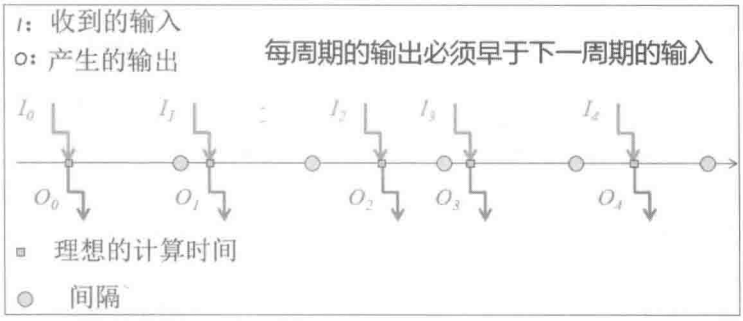

图 1-6　理想计算时间下的反应系统

当然，同步假设中理想的计算时间是不现实的。而在真实系统运行中，只要在每一次新输入发生之前完成了对上次输入的响应，也就满足了同步假设，如图 1-7 所示。

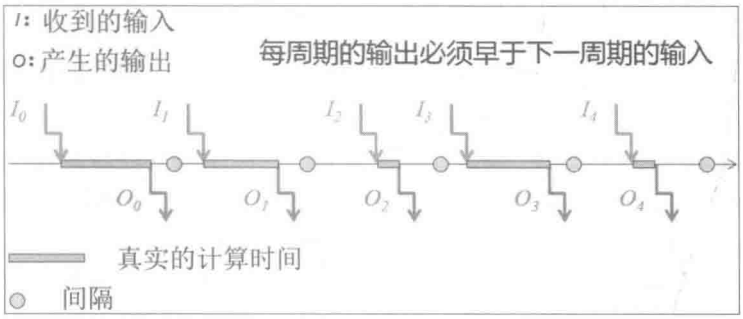

图 1-7　真实计算时间下的反应系统

定义同步假设的内容为：

- “输入-计算-输出”这个过程在规定的时间内完成。
- “输入-计算-输出”的过程中，输入必须不变。
- “输入-计算-输出”的过程中，新到达的输入到下一周期才发生作用。
- “输入-计算-输出”在瞬间完成，可把连续执行的时间离散化：T=0，1，2，3，…
- 当 T=0 周期，系统做初始化工作。
- 当 T>0 的每一个周期，系统做一次“输入-计算-输出”。

再定义“输入-计算-输出”过程的 4 个变量：

- 输入变量：由外部环境给出，记作 I。
- 输出变量：由系统计算给出，提供给外部环境，记作 O。
- 局部变量：由系统计算给出，保存了系统的状态，记作 L。
- 并集变量：是输出变量和局部变量的并集，记作 X，$X=O\cup L$。

则可得到“输入-计算-输出”过程的 3 个解释：

- 输入解释：对输入变量 I 的解释，记作 P。
- 前置介绍：在计算前对 X 的解释，记作 S。
- 后置解释：在计算后对 X 的解释，记作 S^+。

显然，根据同步假设：

T=0 周期

试读结束：需要全本请在线购买：　www.ertongbook.com