21世纪 信息通信系列教材

通信软件设计基础

TONGXIN RUANJIAN SHEJI JICHU

宋茂强 等编著

北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书针对通信软件的特点，介绍了几种适用于通信软件需求分析阶段和设计阶段的形式化语言，包括通用建模语言 UML、消息跟踪语言 MSC 和规格说明与描述语言 SDL，重点介绍了在电信领域得到广泛应用的 SDL 语言。并以设计一个微型交换机软件为例，说明如何运用这些形式化语言进行通信软件的需求分析、概要设计和详细设计。

本书可作为计算机通信专业高年级本科生和研究生的教材，也可供通信软件开发人员参考。

图书在版编目(CIP)数据

通信软件设计基础/宋茂强等编著 .—北京：北京邮电大学出版社，2001.10
ISBN 7-5635-0526-1

Ⅰ.通 … Ⅱ.宋 … Ⅲ.通信软件—高等学校—教材 Ⅳ.TN91

中国版本图书馆 CIP 数据核字（2001）第 057865 号

书　　名：通信软件设计基础
编　　著：宋茂强等
责任编辑：郑　捷
出 版 者：北京邮电大学出版社(北京市海淀区西土城路 10 号)
　　邮编：100876　电话：62282185　62283578
　　网址：http://www.buptpress.com
经　　销：各地新华书店
印　　刷：北京源海印刷厂
印　　数：5 000 册
开　　本：787 mm × 1 092 mm　1/16　印张：14.25 字数：332 千字
版　　次：2001 年 10 月第 1 版　2001 年 10 月第 1 次印刷
书　　号：ISBN 7-5635-0526-1/TN·237
定　　价：26.00 元

前　言

近年来，通信技术发展迅猛，通信系统越来越复杂，通信软件越来越庞大，从事通信软件开发与维护工作的队伍越来越壮大，需要了解和掌握通信软件基础的人也越来越多。为了适应形势需要，北京邮电大学计算机科学与技术学院先后为本科生和硕士研究生开设了“通信软件设计”、“协议工程与通信软件”等课程，但这两门课一直没有合适的教材，本教材试图弥补这一空白。

通信软件的最大特点是广泛使用通信协议和标准。为了便于人们理解和交流，ITU-T在制定各种通信协议和标准时，采用了SDL(Specification and Description Language)语言作为协议和标准的描述语言。根据编者多年来从事通信软件设计的经验，SDL也是一种很好的通信软件设计语言。因此，本教材将重点介绍SDL语言。同时还将介绍通用建模语言UML和消息跟踪语言MSC(Message Sequence Chart)。

本教材共分九章，其中第二章由陈莉萍编写，第六章由张冬梅编写，其余七章由宋茂强教授编写，全书由宋茂强主审。

在编写本教材的过程中，编者得到了杨放春教授、Telelogic公司北京办事处、博士生王颖等的支持和帮助，在此表示衷心的感谢。

李蕾女士为本教材的录入和画图做了大量的工作，在此深表谢意。

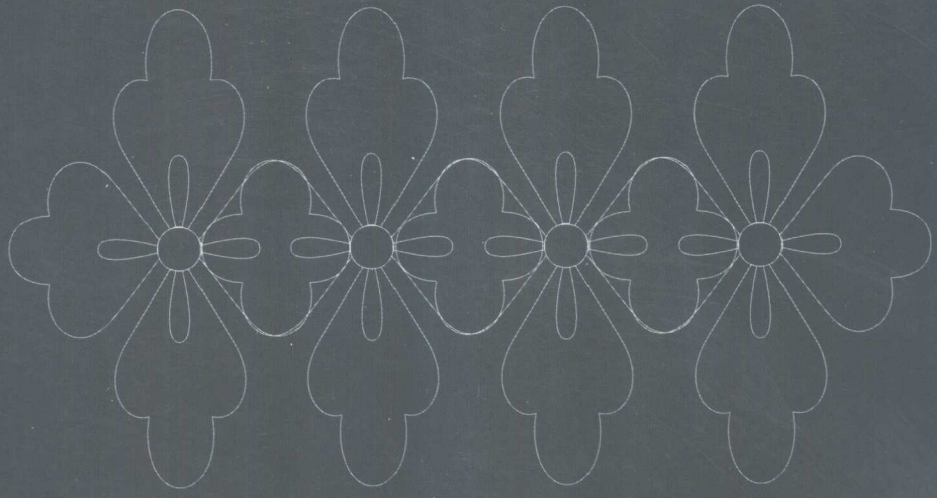
本书可作为计算机通信专业高年级本科生和研究生的教材，也可供通信软件开发人员参考。

由于编者水平有限，书中难免有不完善和谬误之处，敬请读者批评指正。

作　者

2001年8月

责任编辑：郑　捷
封面设计：上官冰冰



ISBN 7-5635-0526-1
9 787563 505265 >

ISBN 7-5635-0526-1/TN·237
定价：26.00元

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第五章 SDL 基本概念

第六章 SDL 中的数据

第一章 概 述

1.1 通信软件的特点

什么是通信软件？凡是用来实现两个或多个实体(计算机,电信终端,交换设备等)之间相互通信的软件都可称为通信软件。通信软件主要有两大类:电信软件和计算机网络软件。电信软件主要包括:电话交换软件、移动通信软件、智能网软件等;网络软件主要包括:各种网络协议和网络应用软件。电信软件和计算机网络软件的特点不尽相同,电信软件具有以下特点:

(1) 实时性强

电信软件系统属于实时系统,系统对外部事件能做出及时响应,如电话交换机用户在任何时刻拿起话机都能听到拨号音(300 ms 以内),用户拨完号码后,系统就会马上为其找到被叫,如果被叫空闲,就可接通被叫(向被叫振铃,向主叫用户送回铃音)。被叫摘机后,双方即可通话。

(2) 运行时间长

电信软件系统一旦投入运行,就要一直运行下去,除非发生故障。因而要求系统具有很高的可靠性。

(3) 并发性强

电信软件系统能够"同时"为成千上万的用户服务,必须具有很强的并发处理能力。

(4) 结果可预期

用户在接受电信软件系统提供的服务时,对其结果是可预期的,如电话用户拿起话机应该听到拨话音(作为主叫时)或可以与对方通话(作为被叫时),如果不是这样,说明系统出故障了,用户只好挂机。用户拨完号码后,可以听到回铃音(被叫空闲)或忙音(被叫忙)或空号音(号码错)或相应的语音提示。否则可认定系统发生故障了。

(5) 跨系统通信

电信软件系统支持跨系统通信,如电话交换机支持电话终端与交换机之间和交换机与交换机之间的通信。交换机之间的通信采用标准信令,如公共信道信令(7 号信令),中国 1 号信令等。

(6) 离散性

电信软件系统是离散系统，支持离散事件，系统行为可以用有限状态机来描述。

网络软件具有以下特点：

(1) 网络软件采用分层结构，如 TCP/IP 协议栈把计算机网络分成四层，即应用层、传输层、网络层和数据链路层，上下层之间通信采用原语形式。

(2) 网络软件主要由网络协议组成，每层使用不同的协议。

(3) 网络协议包含六种元素：服务原语和服务原语时序、协议数据单元(PDU)格式及 PDU 交换时序、协议状态、协议事件、协议变量和协议动作及谓词。

(4) 网络软件实时性要求较低而可靠性要求高。

1.2 通信软件的开发过程

与开发其他软件一样，开发通信软件也需要经历以下阶段：

- 需求分析；
- 概要设计；
- 详细设计；
- 编码；
- 单元测试；
- 集成测试；
- 系统测试；
- 运行维护。

1.2.1 需求分析

需求分析的主要任务是把用户对软件产品的需求和软件产品的运行环境搞清楚，经过对要求和环境的分析，采用某种形式化语言对需求进行描述，最后形成需求规格说明书。用户的需求分为功能性需求和非功能性需求。对于通信软件而言，功能性需求主要包括：软件系统应具有的功能、采用协议或信令系统的情况、与硬件环境的接口关系、与其他系统交互信息的情况等。非功能性需求主要包括：软件的可移植性、可靠性、实时性、可用性、安全保密性和可重用性等。

在需求分析阶段，可以采用对象建模技术(OMT)把分析结果抽象成不同的对象，并描述对象之间的关系。统一建模语言(UML: Unified Modeling Language)是一种比较成熟的标准建模语言。非常适合于实时系统、特别是通信软件系统的分析建模。UML 具有完备的语法和语义定义，具有两种语法表示：图形语法和文本语法。图形语法直观易懂，便于交流，设计人员一般采用图形语法来建模。UML 的主要内容可归纳为静态建模机制和动态建模机制。静态建模机制主要用来构造系统的结构，而动态建模机制主要用来描述系统的行为。关于 UML 更多的内容见第二章。

在需求分析阶段，除了要描述系统的结构和系统行为以外，还需要对系统与应用环境之间交互信息的情况进行描述。ITU-T Z.120 建议中给出的 MSC(Message Sequence Chart，消息顺序图)是一种形式化语言，特别适合于描述通信软件系统与外部环境之间以及系统

此为试读，需要完整PDF请访问：www.ertongbook.com

内各功能模块之间的消息交互情况。关于 MSC 更多的内容见第三章。

需求分析阶段的文档是需求规格说明书。需求规格说明书主要包括以下内容：

(1) 引言

说明本项目的背景，如项目的委托单位和开发单位、本软件系统与其他系统的关系等；给出缩写词的原文和专用术语的定义；列出参考文献。

(2) 任务概述

说明本软件系统的设计目标、运行环境、条件与限制等。

(3) 功能需求

说明本软件系统的功能划分情况、类的定义、系统类图、系统与外部环境之间的消息(信号)交互图等。

(4) 性能需求

说明本软件系统的可移植性、可靠性、实时性、可用性、安全保密性和可重用性等。

(5) 数据描述

给出本软件系统用到的数据项(包括静态数据和动态数据)的描述。如果用到数据库，还要给出数据库的名称和类型等。

1.2.2 概要设计

需求分析阶段解决了系统应该“做什么”的问题，而概要设计和详细设计阶段就是要解决系统“如何做”的问题。概要设计的任务就是根据需求规格说明书，采用合适的形式化语言给出系统的结构设计，划分功能模块，给出模块间接口定义，定义主要的数据结构，设计主要算法等。

ITU-T Z.100 建议中给出的 SDL(Specification and Description Language，规范说明和描述语言)也是一种形式化语言，该语言基于扩展的有限状态自动机(EFSM)模型，特别适合于描述软件系统的离散过程。SDL 既适用于概要设计阶段，也适用于详细设计阶段。由于已有工具支持 SDL 到 C 语言或 C++ 语言的翻译，即代码自动生成，使得 SDL 语言在通信软件设计中得到越来越多的应用。如果在设计阶段采用 SDL 语言，则在概要设计阶段需设计出 SDL 系统图(含功能块划分)，给出必要的信号(消息)定义，设计必要的进程类和功能块类，明确各功能块的功能等。

本书将花较大篇幅来介绍 SDL 语言。

本阶段的文档是概要设计说明书，概要设计说明书的主要内容包括：

(1) 总体设计

说明软件系统的总体结构和功能模块划分情况，以及各模块的功能分配。

(2) 接口设计

设计系统与环境的接口(包括软件接口和硬件接口)和各功能模块之间的接口。

(3) 数据结构设计

设计数据的逻辑结构和物理结构，定义必要的新的数据类型，说明静态数据的加载方法等。

(4) 操作维护管理设计

定义故障信息,设计系统故障监测和故障处理策略,设计系统维护操作等。

1.2.3 详细设计

概要设计完成后,系统功能已划分清楚,系统结构也已确定。详细设计的任务就是细化各功能模块的功能,详细设计系统行为,并用形式化语言来描述设计结果。详细设计分功能模块进行,一般将功能模块分解成多个有限状态自动机(FSM),先画出状态机的状态转移图,然后用 SDL 语言来描述这些状态机。一个状态机对应 SDL 的一个进程,进程间的信号(消息)交互情况可用 MSC 来描述。对于一些复杂的算法,可以定义过程来实现。

编写测试计划也是详细设计阶段的任务,测试计划包括测试方法的设计和测试用例的设计。

本阶段的文档是详细设计说明书。详细设计说明书的主要内容包括:

(1) 软件结构

说明进程划分情况和进程间的消息交互情况,定义本模块内使用的消息。

(2) 数据定义

定义本模块内使用和各进程内使用的数据类型和数据结构,声明变量等。

(3) 行为描述

用 SDL 进程图详细描述系统行为,定义必要的过程,提供变量和过程的共享机制。

(4) 测试计划

设计各进程单元测试的测试方法和测试用例,设计本模块集成测试的测试方法和测试用例。

1.3 通信软件开发工具简介

随着通信技术的不断发展,通信软件的复杂度越来越高,软件系统越来越庞大,对设计者的要求也越来越高。对于如此庞大复杂的软件系统,仅仅依靠人工编码来实现系统是非常危险的。这是因为,由于设计人员的水平不同,人工编码的风格不能保持一致,潜在错误多,调试困难,最终将导致开发周期变长,系统运行不稳定。采用软件开发工具,对通信软件的整个开发过程用系统工程的方法来管理,采用形式化语言来设计,利用模拟验证工具对软件系统进行分析验证,排除软件设计中的逻辑错误,提高系统的可靠性;利用代码自动生成工具来生成代码,保证代码质量,避免人为错误,缩短开发周期。

SDL 语言是 ITU-T 制定标准和协议时使用的形式化语言,也是比较适合于通信软件设计的语言,所以本节介绍一种基于 SDL 的通信软件开发工具:Telelogic Tau。Telelogic Tau 已在我国多家通信设备制造商中得到应用。

1.3.1 概 述

Telelogic Tau 是一套完整的实时软件开发工具,支持开发者使用不同的形式化语言来分析用户需求、设计软件和构造测试集,当系统开发完成后,测试集还能被输出到外部设备,用来测试最终产品。

Telelogic Tau 由以下几部分组成：

- Telelogic DOORS 文档管理系统，支持整个开发过程的文档管理；
- UML Suite 需求分析建模套件，支持用图形化的 UML 语言进行分析建模；
- SDL Suite 设计套件，支持用 SDL 语言进行设计、模拟、验证，支持 SDL 到 C 或 C++ 的自动翻译；
- TTCN Suite 测试套件，支持用测试语言 TTCN 设计测试用例。

1.3.2 需求分析工具

Telelogic Tau UML Suite 是功能强大的分析、建模工具，提供全面的 UML 支持。UML Suite 为基于 UML 表示法全部性能的面向对象的建模提供了完善的工作环境，支持所有 UML 图，包括类图、顺序图、用例图和状态图。UML Suite 是一套完整的 UML 工具，支持需求规格说明、逆向工程、代码生成和重用、文档自动生成等，可帮助开发者在软件开发的早期检测到错误，从而降低开发和维护成本。Telelogic Tau UML Suite 主要包括：

- Class Diagram Editor 类图编辑器，支持图形化的 UML 类图编辑和静态语义检查；
- Sequence Diagram Editor 顺序图编辑器；
- Use Case Diagram Editor 用例图编辑器；
- Statechart Diagram Editor 状态图编辑器。

Telelogic Tau 包含一个通用的 UML 到 SDL 转换工具，可将 UML 类图、顺序图和状态图转换成对应的 SDL 图和 MSC 图。该转换工具使开发者很容易地从需求分析过渡到设计阶段，它保留 UML 分析模型中的信息，直接跳到 SDL 设计，利用 SDL Suite 提供的仿真和验证功能，可以仿真 UML 模型。

1.3.3 设计和实现工具

Telelogic Tau SDL Suite 主要用于设计阶段，支持图形化的 SDL 编辑、在线仿真和验证、代码自动生成、MSC 编辑及测试用例自动生成等。SDL Suite 由以下几部分组成：

- SDL Editor：SDL 编辑器，支持图形化的 SDL 编辑，同时提供一个在线检查的分析器，可以立即标出编辑中的静态语法和语义错误。
- SDL Simulator：SDL 仿真器，支持在开发环境下进行 SDL 系统的仿真运行，仿真结果可自动生成 MSC。仿真器还可以自动穷举搜索 SDL 系统的所有路径，保证系统的每一个状态都是可达的。
- SDL Validator：SDL 验证器，结合 MSC 验证系统行为是否与需求描述一致。
- SDL Translator：SDL 代码自动生成器，支持三种代码生成：第一种是无限制的 C 代码生成器，支持一般的 C 和 C++ 代码生成；第二种是优化的 C 代码生成器，支持小型嵌入式系统的优化代码生成，能满足这种系统对代码效率、最小内存和高性能的要求；第三种是 CHILL 代码生成器。由于在仿真和验证阶段使用的代码与最后生成的代码是相同的，这就保证了测试系统与实际运行系统的一致。
- MSC Editor：MSC 编辑器，支持图形化的 MSC 编辑。

SDL Suite 还支持与实际目标环境的集成。它包含一个运行库，提供 SDL 系统与环境

打交道的源代码，支持多种操作系统，开发者可以在单任务、多任务和分布式系统之间进行选择。自动生成的代码加上指定操作系统的运行库源代码，经过合适的 C 编译器编译连接，即可在实际环境中运行了。

1.3.4 测试集设计与实现

Telelogic Tau TTCN Suite 是事实上的通信系统的标准测试环境。它可以被用来测试从内置芯片到大型交换机和智能网业务的电信和数据通信设备。TTCN 是世界上应用最广泛的专用编程测试语言，并被 ISO 标准化用于测试复杂的通信系统。Telelogic Tau TTCN Suite 提供给测试工程师强大的 TTCN 编辑器、句法分析器和编译器，可将测试方案转化为 C 代码，在测试设备上运行。

Telelogic Tau 还提供从 SDL 系统设计和 MSC 直接生成 TTCN 测试集的功能。

思 考 题

1. 什么是通信软件？
2. 通信软件的主要特点是什么？
3. 开发通信软件与开发非通信软件（如数据库软件）有何区别？

第二章　UML与建模技术

本章将介绍UML的基本知识及建模技术，并着重介绍UML在软件研发过程中需求分析阶段的应用。

2.1　UML概述

UML(Unifined Modeling Language)，称为统一建模语言。1997年11月被OMG(Object Management Group)采纳为基于面向对象技术的工业标准。UML融入了软件工程领域的新思想、新方法和新技术，其作用域不仅限于支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程。

UML是Booch，Objectory和OMT方法的结合，是为简化和强化现有的大量面向对象开发方法而设计的。它是一种定义良好、易于表达、功能强大且通用的可视化建模语言；是对软件系统进行描述、可视化处理、构造和建立软件系统产品文档的建模语言。其主要适用于分析和设计阶段的系统建模。UML不是一门程序设计语言，但可以使用代码生成工具将UML模型转换为多种程序设计语言代码，或使用反向生成工具将程序源代码转换为UML模型。

2.1.1　UML的定义及特点

UML是一种标准的图形化建模语言，是面向对象分析和设计的一种标准表示。其主要内容包括静态建模机制和动态建模机制两大类。

UML的定义包括两个组成部分：语义和语法(表示法)。

UML语义通过元模型来定义。元模型是描述模型的语言，通过自然语言文本和UML图进行组合描述。元模型为UML的所有元素在语法和语义上提供了简单、一致、通用的定义性说明，使开发者能在语义上取得一致，并有可能通过统一UML元素大大简化模型。

UML表示法定义了UML表示符号，为开发者或开发工具提供了标准的图形符号和正文语法。这些图形符号和文字所表达的是应用级的模型，在语义上它是UML元模型的实例。使用这些图形符号和正文语法可构造标准的系统模型。

UML统一了Booch，OMT和Objectory等方法中的基本概念，并具有更强、更清晰和一致性等优点，同时它吸取了面向对象技术领域中其他流派的长处，简化了建模方法，并在

演变过程中提出了一些新的概念。UML:

- 不是一种可视化的程序设计语言,而是一种可视化的建模语言;
- 不是工具或知识库的规格说明,而是一种建模语言规格说明,是一种表示的标准;
- 不是过程,也不是方法,但允许任何一种过程和方法使用它;
- 与具体实现无关,可应用于多种语言平台和工具平台;
- 与具体过程无关,可应用于不限于面向对象的任何软件开发过程;
- 易于使用,表达力强,简单且易扩展;
- 可升级,具有广阔的适用性和可用性;
- 有利于面向对象工具的市场成长。

2.1.2 UML 应用领域

UML用于系统建模,具有广泛的应用范围。UML既可以描述许多类型的系统,也可以用于系统开发的不同阶段,从需求规格说明到系统完成后的测试和维护。

1. 不同类型系统的应用

UML可直接为面向对象软件系统创建模型,也可用来描述其他类型的软件系统、商业机构或过程。常见应用包括:信息系统、具有实时要求的工业系统或工业过程、嵌入式实时系统、分布式系统、系统软件、商业系统等。

其中,商业工程是面向对象建模应用的一个新领域,UML非常适合为公司的商业过程建模。运用商业过程再工程(Business Process Reengineering, BPR)或全面质量管理(Total Quality Management, TQM)等技术,可以对公司的商业过程进行分析、改进和实现。使用UML建模语言为过程建模和编制文档,使系统易于使用。

多数情况下的实际系统是多种系统的结合,如现在许多信息系统都有分布式和实时性的需要。总之,UML是一个通用的标准建模语言,可以对任何具有静态结构和动态行为的系统进行建模。

2. 系统开发过程的应用

UML的应用贯穿于软件开发过程的不同阶段。

首先,需求分析是项目开发的源头,具有非常重要的地位。在需求分析中,可采用用例来捕获用户需求。通过用例建模,描述对系统感兴趣的外部角色及其对系统(用例)的功能要求。由于用例不描述系统全貌,因此需求分析要建立应用领域的概念模型,或称域模型。这时主要关心问题域中的主要概念(如抽象、类和对象等)和机制,需要识别这些类以及它们相互间的关系,并用UML类图来描述。为实现用例,类之间需要协作,这可以用UML动态模型来描述。最初的域模型只是一个框架,而非真正的高层模型。在此阶段,只对问题域的对象(现实世界的概念)建模,而不考虑定义软件系统中技术细节的类,如处理用户接口、数据库、通信和并行性等问题的类,这些技术细节将在设计阶段引入。

在设计阶段提出的技术解决方案将利用分析阶段的结果,加入新的类来提供技术基础结构——用户接口,数据库操作等。分析阶段的领域问题类被嵌入在这个技术基础结构中,而设计阶段的结果是构造阶段的详细规格说明。

构造(实现)是一个独立的阶段,其任务是用面向对象编程语言将来自设计阶段的类

此为试读,需要完整PDF请访问: www.ertongbook.com