



面向21世纪高等院校计算机系列教材

软件体系结构 及实例分析

余雪丽 主编



科学出版社

www.sciencep.com

面向 21 世纪高等院校计算机系列教材

软件体系结构及实例分析

余雪丽 主编

须毓孝 李林峰 王 莉 副主编

科学出版社

北 京

内 容 简 介

本书共三篇12章,第1~4章构成基础篇,重点介绍软件体系结构的基本概念和基本原理,明确提出软件体系结构的动态性和适应性的理念。第5~9章构成应用篇,从身边的实例出发,以理论为指导,分析软件体系结构原理的具体应用,以求缩短软件体系结构理论与实际软件开发之间的距离。第10~12章构成发展篇,选择以网络为中心的前沿应用实例 CORBA、Web Services、P2P、MAS 及 Grid 来分析软件体系结构的作用,以及软件体系结构的动态性、适应性的体现,通过对新应用、新服务中体系结构及其动态性、适应性的初步分析,起到抛砖引玉的作用。

本书的读者群包括计算机及相近专业本科高年级学生、计算机应用和软件专业的硕士研究生、软件学院的学生。同时,本书对于在企业和公司从事软件开发的技术人员、软件密集型项目及开发团体的软件构架师、项目负责人等都有一定的指导作用。

图书在版编目(CIP)数据

软件体系结构及实例分析/余雪丽主编. —北京:科学出版社, 2004

(面向 21 世纪高等院校计算机系列教材)

ISBN 7-03-014033-8

I. 软… II. 余… III. 软件-系统结构-高等学校-教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2004)第 077094 号

责任编辑:陈晓萍/责任校对:柏连海

责任印制:吕春珉/封面设计:三函设计

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004年8月第 一 版 开本: B5 (720×1000)

2004年8月第一次印刷 印张: 21 1/2

印数: 1-4 000 字数: 398 000

定价: 30.00 元

(如有印装质量问题,我社负责调换<环伟>)

面向 21 世纪高等院校计算机系列教材
编委会

顾问委员：刘开瑛 刘 璟 李东福 施伯乐 谢克昌

主任委员：左孝凌

副主任委员：

陈立潮 陈俊杰 余雪丽 李焕珍

梁吉业 曾建潮

委 员：

马尚才 亢临生 左孝凌 刘晓融 陈立潮

陈俊杰 李东生 李济洪 李焕珍 余雪丽

张荣国 张继福 杨 威 贺利坚 段 富

陶世群 梁吉业 曾建潮 谢康林 韩 燮

缪淮扣

序

在高度信息化的 21 世纪,人们越来越认识到信息教育的重要性。人们都迫切希望信息教育能有较大发展。教育信息化也是摆在我们面前的重要任务。教育部明确要求高等教育实行信息化,要求在未来 5 年内实现信息化教育课程的数量达到 15%~30%。信息社会离不开计算机技术,知识经济需要大量的计算机高级人才。我国正在加强计算机的高等教育,正着眼于为新世纪培养高素质的计算机人才,以适应信息社会高速发展的需要。当前,全国各类高等院校都在各专业基础课程计划中增加计算机的课程内容,而作为与计算机科学密切相关的计算机、通信、信息等专业,更是在酝酿着教学的全面改革,以期规划出一整套面向 21 世纪的、具有中国高校计算机教育特色的课程计划和教材体系。

教育部《关于加强高等学校本科教育工作提高教育质量的若干意见》(教字【2001】4 号)文件也强调指出:“要大力提倡编写、引进和使用先进教材。教材的质量直接体现着高等教育和科学研究的发展水平,也直接影响本科教学的质量。高等学校要结合学科、专业的调整,加快教材的更新换代。”

为推动高校教学改革,提高教学质量,我们重点抓了 21 世纪高等教育教学改革项目,组织并支持了“面向 21 世纪计算机系列教材规划”研究课题。该课题组成员均由高校计算机系的专家教授组成。他们有多年的丰富的教学经验,也具有很强的科研能力。该课题的主要目标是密切结合国民经济的需要,优化计算机教材体系结构,力求将国际、国内计算机领域的新概念、新理论、新技术吸收到本系列教材中,编写出具有科学性、先进性、系统性、实用性、实践性很强的教材,经过推广使用,反复修改,不断提高。

“面向 21 世纪计算机系列教材规划”课题以编写非计算机专业的计算机课程、计算机专业的计算机网络课程、计算机软件课程三个系列教材为主要内容,计划在三年内出版 13~16 种书,服务于本科生、专科生、研究生,以及网络学院和软件学院的学生。本课题把研究系列教材的重点放在影响和带动计算机学科发展的网络与软件,以及直接推动计算机普及和应用的非计算机专业三个方向上,目的是通过集中优势兵力,加强团队协作,能够在教材建设方面按系列有所突破。

相信本套教材的出版必将对教学改革和教材建设起到很大的推动和示范作用。


2002 年 6 月 10 日

前 言

自 20 世纪 80 年代以来, 计算机网络和硬件的发展突飞猛进, 日新月异。相对而言, 软件的发展却远不如人意。为彻底扭转软件开发手工作坊的局面, 从根本上解决软件危机, 保证软件质量, 提高开发效率, 现代软件工程提出了许多新理念, 产生了一系列新的研究分支和新方法。软件体系结构作为现代软件工程的重要组成部分, 已逐渐被学术界和企业界的计算机专家、软件开发人员所接受, 被自觉应用, 成为科学技术人员考虑问题、进行软件设计和工程实现的基本出发点。但是同时, 人们又普遍认为软件体系结构太抽象, 与一般的软件开发活动相距甚远。而且, 软件体系结构不是一个简单的静态设计问题, 是要随着用户的需求和软件生存的环境的变化而变化。

因此, 本书的宗旨是针对上述问题, 力图处理好两对对立的统一体, 既要高屋建瓴, 又要脚踏实地; 既要夯实基础, 又要与时俱进。具体来讲, 软件体系结构是现代软件工程的一个重要分支, 研究的是如何运用现代的理念和方法解决软件整体构架的高层次复用问题, 这是软件体系结构高屋建瓴的一面; 同时, 通过分析大型复杂应用系统的体系结构, 用已开发过的实例从体系结构的视角去剖析它, 从软件体系结构的设计、验证、评估和文档化的一系列活动的要求重新分析这些实例, 反映出对软件体系结构的研究脚踏实地、面向应用的另一面。软件体系结构的研究作为一个学科分支, 有它的基础理论, 有抽象的风格描述, 有作为软件自动生成基础的形式化方法, 这是夯实软件体系结构基础的必经之路; 十多年的软件体系结构研究历程表明, 软件体系结构的动态性、适应性是推动该领域科学研究不断深入的动力。目前, 层出不穷的新应用、新服务, 如 MAS、Grid、Web Services 都要求软件密集型复杂应用系统有动态的、能适应外界环境变化的软件体系结构。这是软件体系结构的研究与时俱进, 不断发展的另一个侧面。

为贯彻上述宗旨, 本书的第一篇——基础篇重点介绍软件体系结构的基本概念和基本原理, 明确提出软件体系结构的动态性和适应性的理念。第二篇——应用篇从身边的实例出发, 以理论为指导, 分析软件体系结构原理的具体应用, 以求缩短软件体系结构理论与实际软件开发之间的距离。第三篇——发展篇选择以网络为中心的前沿应用实例 CORBA、Web Services、P2P、MAS 及 Grid 来分析软件体系结构的作用, 以及软件体系结构的动态性、适应性的体现。

余雪丽负责对本书的统一组织和策划。第 1、4、10、11 章由王莉编写, 第 2、3 章由张国有编写, 第 5 章和 6.1、6.2 节由孙静宇编写, 6.3 节和第 7 章由牛彦青负责撰写, 6.4~6.6 节由须毓孝负责撰写, 张英俊、李林峰和王辉分别撰写第 8 章、

第 9 章和第 12 章。全书由刘永泰进行全面审阅，崔冬华统一文字和语言的风格。研究生郑丽伟、乔玲玲、李冰清、程俊丽、李剑峰、李志芳、梁国军、李祝平、张晓波、袁占花、刘旭、郑凯参与了书稿内容和文字的斟酌、修改、校对及绘图工作。在这里要特别感谢北京航空航天大学计算机学院的金茂忠教授、张莉老师对编写大纲提出的宝贵意见和指导，并感谢董志峰的支持。

由于作者水平所限，本书难免会有不妥和疏漏之处，敬请读者批评指正。

余雪丽

2004 年 6 月

目 录

第一篇 基础篇

绪言	1
第 1 章 软件体系结构概述	7
1.1 基本概念	7
1.1.1 软件体系结构的定义	7
1.1.2 对软件体系结构的认识	9
1.1.3 各种术语和相关概念	11
1.2 软件体系结构的重要性	12
1.2.1 在风险承担和早期设计中的作用	13
1.2.2 软件体系结构在软件开发各阶段中的作用	14
1.2.3 软件体系结构是系统分析和设计的高层复用	15
1.3 软件体系结构研究的领域	17
思考题一	20
第 2 章 体系结构的风格与模式	21
2.1 主程序与子过程	22
2.1.1 结构化控制流	22
2.1.2 结构化连接方式	24
2.1.3 主程序与子过程风格的优缺点	26
2.2 面向对象组织和抽象数据类型	26
2.2.1 抽象数据类型	27
2.2.2 面向对象	28
2.2.3 面向对象的优缺点	30
2.3 管道过滤器风格	31
2.3.1 概述	31
2.3.2 管道过滤器的优缺点	31
2.4 分层风格	32
2.4.1 概述	32
2.4.2 层次系统的优缺点	34
2.5 事件驱动和隐式激发风格	36
2.5.1 事件驱动的特征	36

2.5.2 事件驱动和隐式调用的优缺点	37
2.6 其他风格	37
2.6.1 知识库风格	37
2.6.2 解释器系统	39
2.7 面向模式的软件体系结构	39
思考题二	40
第3章 软件体系结构的形式化方法和集成开发环境	41
3.1 形式化方法简介	41
3.1.1 形式化方法的基本概念	41
3.1.2 软件体系结构的研究背景	42
3.2 软件体系结构的 Z 语言描述	46
3.2.1 Z 语言简介	46
3.2.2 软件体系结构的 Z 语言描述	50
3.3 体系结构的 CSP 描述	54
3.3.1 通信顺序进程简介	54
3.3.2 体系结构的 CSP 描述	55
3.4 软件体系结构描述语言	57
3.4.1 体系结构描述语言的定义与分类	57
3.4.2 体系结构描述语言的目标	59
3.4.3 体系结构描述和分析的要求	60
3.4.4 体系结构构件和连接件建模	60
3.4.5 体系结构配置建模	61
3.5 软件体系结构集成环境	62
3.5.1 集成环境的设计目标	62
3.5.2 集成环境的构成元素	63
3.6 Wright 系统介绍	64
3.6.1 实现关系与交互关系	65
3.6.2 体系结构描述的推理	65
3.6.3 Wright 语言的描述	66
3.7 Darwin 系统介绍	69
3.8 ACME 系统介绍	73
3.8.1 构件的描述	73
3.8.2 连接件的描述	74
3.8.3 系统的描述	75
3.8.4 风格的描述	76

思考题三	76
第4章 动态软件体系结构	78
4.1 动态软件体系结构的概念	78
4.1.1 需求变更引起体系结构的变化	79
4.1.2 对运行时体系结构变化的处理	80
4.1.3 大型复杂软件系统体系结构的动态性	82
4.2 动态软件体系结构研究的发展历程	84
4.2.1 动态体系结构的早期研究	84
4.2.2 动态软件体系结构研究的发展	86
4.2.3 MAS 及网格计算中的动态体系结构	87
4.3 基于 Web 的大型复杂系统需要动态体系结构	89
4.3.1 基于 Web 的大型复杂软件系统的特点	89
4.3.2 典型应用系统软件体系结构的分析	92
4.4 软件体系结构与软件开发过程的关系	103
4.4.1 统一过程概述	104
4.4.2 统一开发过程以软件体系结构为中心	108
思考题四	110

第二篇 应用篇

第5章 需求与需求分析	113
5.1 基本概念	113
5.1.1 需求	113
5.1.2 需求分析	114
5.1.3 需求工程	114
5.1.4 具有动态性要求的需求与需求的动态性	115
5.1.5 其他概念	116
5.2 需求分析	117
5.2.1 重要性	117
5.2.2 需求获取与分析	118
5.2.3 UML 方法与需求分析	120
5.3 需求表示	122
5.3.1 需求框架	122
5.3.2 需求规格说明书	122
5.3.3 多视点的需求描述方法	124
5.4 需求验证与管理	130

5.4.1	需求验证	130
5.4.2	需求管理	131
5.5	一个需求获取的实例	132
5.5.1	实例背景	132
5.5.2	具有动态性要求的需求	133
5.5.3	需求的动态变化性	134
5.5.4	需求分析的实例	136
5.6	小结	138
	思考题五	139
第6章	软件体系结构设计	140
6.1	软件体系结构设计概述	140
6.1.1	软件体系结构设计过程	140
6.1.2	软件体系结构设计的进一步认识	142
6.1.3	软件体系结构设计的方法概述	142
6.2	软件体系结构设计	143
6.2.1	软件总体需求了解阶段	143
6.2.2	选择大颗粒度软件体系结构风格阶段	144
6.2.3	需求获取和表示阶段	144
6.2.4	需求到软件体系结构各要素的映射	144
6.2.5	软件体系结构各组成要素设计阶段	145
6.2.6	详细设计软件体系结构阶段	145
6.2.7	验证软件体系结构的正确性	145
6.2.8	软件体系结构的文档化和评估	146
6.2.9	软件体系结构迭代和优化设计	146
6.3	需求到软件体系结构各要素的映射	147
6.3.1	系统模型	147
6.3.2	需求到体系结构的映射机制	149
6.3.3	体系结构模型之间的约束关系	160
6.4	设计软件体系结构各组成要素	164
6.4.1	用户界面设计	165
6.4.2	数据模型设计	167
6.4.3	控制模型设计	170
6.5	详细设计软件体系结构阶段	172
6.5.1	业务过程建模	172
6.5.2	领域建模	177

6.5.3 软件体系结构建模	177
6.6 验证和变更管理	183
6.6.1 针对规格说明的验证	184
6.6.2 用户界面的验证	185
6.6.3 数据库验证	186
6.6.4 管理变更	186
思考题六	188
第7章 文档化体系结构	189
7.1 文档化的必要性	189
7.1.1 系统文档	190
7.1.2 用户文档	192
7.2 文档化软件体系结构	192
7.2.1 文档化体系结构的必要性	192
7.2.2 文档化体系结构的基本规则	193
7.2.3 体系结构文档化的实施	194
思考题七	196
第8章 软件体系结构评估	197
8.1 评估体系结构的原因	197
8.2 软件体系结构评估的方法	198
8.2.1 SAAM 评估方法	198
8.2.2 ATAM 评估方法	202
8.2.3 评估方法的比较	208
8.3 体系结构评估	209
8.3.1 参与者	209
8.3.2 考察的质量属性	210
8.3.3 评估的结果	211
8.3.4 评估的收益和成本	212
8.4 软件体系结构评估实例	214
8.4.1 准备工作	214
8.4.2 第1阶段	215
8.4.3 第2阶段	221
8.4.4 BCS 评估的结果	222
8.5 小结	223
思考题八	224
第9章 案例分析与设计	225

9.1 了解系统总体需求并确定风格.....	225
9.2 获取和表示需求	226
9.2.1 识别参与者	227
9.2.2 识别用例.....	227
9.2.3 用例的事件流描述	229
9.3 需求到构件的映射	231
9.3.1 系统对象定义	231
9.3.2 建立类图	231
9.4 需求的动态行为表示	234
9.5 模型到数据库的映射	243
9.5.1 数据库模式要面向应用系统	243
9.5.2 对象模型向数据库表的映射规则	243
9.5.3 数据库的映射和逐步细化过程的分层描述	244
9.6 细化体系结构	246
9.6.1 细化的步骤	246
9.6.2 构件设计	247
9.6.3 连接件设计	248
9.7 合理布局软件体系结构各组成要素	249
9.8 验证软件体系结构的正确性	251
9.9 体系结构文档化及评估	251
思考题九.....	251

第三篇 发展篇

第 10 章 网络环境下的软件体系结构	255
10.1 分布式软件体系结构	255
10.1.1 局域网文件服务器体系结构	256
10.1.2 客户/服务器体系结构	257
10.1.3 基于 Web 的体系结构	258
10.1.4 分布式软件体系结构风格	258
10.2 公共对象请求代理结构 CORBA	260
10.2.1 引言	260
10.2.2 CORBA 基本组成与框架	261
10.2.3 运行机制	263
10.2.4 CORBA 体系结构的动态性与适应性分析	263

10.3 对等结构	264
10.3.1 对等结构简介	264
10.3.2 对等结构的关键技术和特性	264
10.3.3 对等结构的应用	265
10.3.4 对等结构的问题	267
10.4 Web 服务 (Web Services)	269
10.4.1 Web Services 的产生背景	269
10.4.2 Web Services 的软件体系结构	269
10.4.3 Web Services 的运行机制	273
10.4.4 基于 J2EE 的 Web Services 体系结构	273
10.4.5 Web Services 的开发生命周期	275
10.4.6 Web Services 体系结构分析	276
10.4.7 面向服务的体系结构 SOA	277
思考题十	281
第 11 章 多 Agent 系统的动态软件体系结构	282
11.1 多 Agent 系统	282
11.1.1 多 Agent 系统简介	282
11.1.2 多 Agent 系统的特征	283
11.1.3 多 Agent 系统体系结构的描述方法	285
11.2 典型的多 Agent 系统体系结构	286
11.2.1 FIPA 规范的体系结构	286
11.2.2 OMG 体系结构	289
11.2.3 其他体系结构	290
11.3 多 Agent 系统应用实例分析	291
11.3.1 远程会议预约系统——一种典型的分布式协作	292
11.3.2 基于 JADE 平台的实例系统开发	295
11.3.3 Web Agent——被广泛应用的一种小型 Agent 系统	304
思考题十一	305
第 12 章 网格动态体系结构	306
12.1 基本概念	306
12.2 五层沙漏结构	307
12.2.1 基本思想	307
12.2.2 软件体系结构描述	308
12.2.3 动态性分析	311
12.3 开放式网格服务体系结构 OGSA	311

12.3.1 基本思想	311
12.3.2 软件体系结构描述	312
12.3.3 动态性分析	314
12.4 网络软件体系结构动态性实例分析	315
12.4.1 实例的动机与方法	315
12.4.2 软件体系结构描述	317
12.4.3 基于软件体系结构的适应性	318
思考题十二	321
参考文献	322

第一篇 基础篇

绪 言

“软件体系结构允许或拒绝软件的质量属性。”

如同计算机系统结构在计算机工程中的重要作用一样，对于软件体系结构（Software Architecture, SA）的概念、高层设计和复用的研究是“现代软件工程”的一个重要组成部分，也是近十多年软件工程发展最为迅速的研究领域之一。我们认为本书的编写，应当遵循以下三个基本指导思想：软硬件协同设计，遵循统一标准，深刻认识现代复杂动态软件体系结构对适应性的要求。

1. 软硬件协同设计（Hardware and Software Co-Design）

2003年1月，卡耐基-梅隆大学软件工程研究所 SEI 受委托在美国的华盛顿附近邀请政府、工业界和学术界的有关人士举办了关于国防部体系结构框架 DODAF（Department of Defense Architecture Framework）和软件体系结构 SA 的工作会议。这个会议为与会者提供了一个讨论应用系统的整体框架和软件体系结构之间的相关性、相似性和差异性的论坛。DODAF 是由美国国防部委托为大规模系统的表示建立的基础框架，它从系统观点 SV（System View）、操作观点 OV（Operational View）以及技术观点 TV（Technical View）三个不同的角度描述系统，共包含 20 种产品。DODAF 的前身是 C4ISR，即命令、控制、通信、计算机、智能、监视和侦察的体系结构框架（Command、Control、Communications、Computer、Intelligence、Surveillance and Reconnaissance）。美国国防部注意到，多年来在他们致力于研究大规模现代应用系统的体系结构的同时，卡耐基-梅隆大学的软件工程研究所 SEI 已经开辟了一套途径，对软件体系结构进行分析、构建和文档化，同时 UML 已经成为被普遍接受的描述软件设计的标准规范。SEI 和 UML 都在使用多种视角描述和表达软件体系结构。这次会议的目的是使与会者进一步认识到在构建系统和软件体系结构方面面临着新的挑战。

毫无疑问，在建立较大规模应用系统的工程中，体系结构是一个最基本的组成部分。当前，人们看到有关体系结构的名词铺天盖地而来，例如“信息体系结

构”、“企业体系结构”、“系统体系结构”、“聚合系统体系结构”、“软件体系结构”、“通信体系结构”、“硬件体系结构”、“安全体系结构”、“数据体系结构”等。很难找到有关以上种种体系结构的确切定义，“系统体系结构”和“软件体系结构”是使用最广泛的概念。无论是建构“系统体系结构”或者是“软件体系结构”都与构建这些结构的目的密切相关。“系统体系结构”与“软件体系结构”的一个很重要的不同点在于：“系统体系结构”（或者说系统工程师或 DODAF 框架）关心的是功能性，是领域提出的挑战，而“软件体系结构”（或者说软件工程师）关心的是功能的实现途径。在开发一个大型应用系统的初始阶段，如果该系统属于大规模软件密集型系统时，要从 DODAF 的一组观点演化到一组软件体系结构的观点；当采用快速演化的资源选择机制时，应挑选合适的 DODAF 观点，以及某些能描述重要软件能力的软件体系结构观点；在作投资决定时，某些软件体系结构的观点特别是关系到投资者利益的观点应当被考虑，以便拓展在 DODAF 中表达能力比较差的一些方面。DODAF 在捕捉系统详细设计方面信息的能力比较弱。比如，对于牵涉到分布式数据或分布式计算的软件风格的描述比较欠缺，对于关系到终端用户的演算能力方面，如合理的人机界面的描述也比较差。总之，DODAF 与当前的软件体系结构开发方法是为了不同的目的，由不同的组织分别发展的，它们有一些重叠，但在表达体系结构方面则有显著的不同。为开发一个系统或者一个软件，有必要选择不同的观点。IEEE 标准 1471-2000 和 SEI 的观点及途径都可以为选择观点提供指导。

很久以来，人们认识到“体系结构”对一个应用系统的生命周期有着非常重要的影响。过去，与硬件相关的体系结构起着主导作用，与软件相关的体系结构成分经常在开发过程中成为牺牲品。今天，大规模软件应用系统比比皆是，软件系统开发的费用以及软件系统的复杂性不断增加已经打破了软件与硬件以往的平衡。由于软件开发技术的迅速成熟，整个系统开发实践可以从软件体系结构的基本原理中得到极大的益处。

美国国防部提出的体系结构框架与软件体系结构是有联系的，各自侧面面有所不同。从长期的软件开发经验中，软件开发者和软件开发管理者深刻体会到，好的软件设计师需要有足够的功底，能做到硬件吃透，软件精通，熟悉工艺，才能在把握和运筹大规模软件系统时灵活掌握，运用自如。他们既可以在需要考虑运行速度时，采用专用硬件加速，又可以在需要降低成本时以软件程序实现预定功能，还能借鉴硬件设计的成功思想用以指导软件设计。从本质上讲，软件与硬件的设计和实现是相通的，是可以相互转换的。

2. 遵循统一标准

(1) IEEE 体系结构工作组 AWG

“软件体系结构”是非常重要的。但是，体系结构的概念还没有统一，也没有