


计算机应用技术教材

软件体系结构 的原理、组成与应用

万建成 卢雷 编著



 科学出版社

计算机应用技术教材

软件体系结构的 原理、组成与应用

万建成 卢雷 编著

科学出版社

2002

内 容 简 介

本书全面讨论了软件体系结构研究的由来和发展,介绍了有关体系结构概念的构成和关于体系结构的不同观点,并给出了描述体系结构的有关理论和方法的研究成果。在此基础上指出了体系结构的多层次概念,并就各个层次的构成通过举例进行了全面的说明。体系结构的举例都是来自现实常用和通用的软件技术和应用。通过本书的阅读,读者将对体系结构概念和知识有一个全面而实际的认识。

本书面向软件专业的大学高年级学生、研究生、从事软件工作的工程技术人员,以及对软件体系结构研究和应用感兴趣的读者。

图书在版编目(CIP)数据

软件体系结构的原理、组成与应用/万建成,卢雷编著. —北京:科学出版社, 2002

(计算机应用技术教材)

ISBN 7-03-010673-3

I.软... II.①万...②卢... III.软件工程—系统结构 IV.TP311.5

中国版本图书馆CIP数据核字(2002)第051932号

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>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2002年8月第一版 开本:787×1092 1/16

2002年8月第一次印刷 印张:24 3/4

印数:1—4 000 字数:578 000

定价:33.00元

(如有印装质量问题,我社负责调换(环伟))

前 言

软件体系结构是在识别可重用部件和连接器的基础上,研究软件结构的表达和分析的理论和技术的。它在过去近十年间形成并获得了广泛的关注。作为软件重用技术的高级阶段,体系结构已经形成软件工程研究的一个重要分支和热点。

在经历了“软件危机”的痛苦、反省和研究发展阶段后,当今的软件设计技术已经有了翻天覆地的变化。经过面向对象技术的发展和促进,新型软件的出现,特别是可视化程序技术和 Internet 技术的应用,软件设计方法和开发环境有了巨大的变化。人们对软件的特性和基础结构有了从未有过的深刻认识。反映在设计理论和方法方面,体系结构概念的提出,标志着软件技术将在新的起点上走向更加成熟。软件设计已经从关心基本的结构和算法转移到了对宏观结构的认识上;从关注功能的实现转移到了对综合性能的要求上。

软件体系结构研究的内容和目标都已经显示在系统地、全面地归纳软件的基础和通用模块上,并使其标准化和规范化。未来的软件工程设计会逐步发展成为如同机械和建筑设计的机械组装过程一样,从而摆脱艰难的脑力劳动和无可靠质量保证的手工劳动。这其中,软件体系结构将起到主导作用。也就是说,新一代的软件工程将是体系结构的工程。

作为软件工程的一个分支和一个新的软件设计视角,体系结构在软件设计中的重要作用已经得到普遍的认可。建立在 UML 上的“统一软件开发过程”中,也已经提出了以体系结构为中心的软件设计开发过程。

但是,目前的体系结构研究仍主要集中在理论和实验阶段,尽管已经开展了大量卓有成效的工作,但除了在部件和连接器的一般概念外,人们对于体系结构的认识尚不统一,体系结构的问题分析和结构描述方法差异很大,有关的工程工具和环境还没有建立。然而,在认识到体系结构的重要性后,体系结构的分析设计对于软件工程实施已经是不可缺少的了。

编著本书的目的,正是从软件工程的现实技术和实现环境出发,为读者树立一个多层次结合的软件体系结构概念,从而为正确地分析和建立实际系统的体系结构奠定坚实的基础。本书的内容是:在全面介绍了软件体系结构概念的由来和发展的背景,以及主要的体系结构描述方法后,依据作者十多年积累的软件知识,叙述了软件体系结构的层次概念,并为每个层次列举了主要的和常用的结构模式。通过全书的阅读,读者对于软件体系结构的背景将有一个全面的了解,对于体系结构的组成和表达将有一个实际和深入的认识。

相信,在计算机技术飞速发展的今天,对于任何立志从事软件研究和工程设计的人员来说,体系结构的概念以及实际经验都是不可缺少的,而且是越来越重要的。

参加本书编著的有万建成、卢雷。万建成策划了全书的构成,并编写了 1~5 章、7~11 章。卢雷编写了第 6 章、第 12 章及第 13 章。在全书的校阅中,多名研究生也参加了部分工作。

在全书的形成过程中,山东大学计算机科学技术学院的 2000 和 2001 级软件专业的许

多研究生给予了帮助，参加了资料的收集和整理，为本书的形成作出了积极的贡献；软件工程研究室各位同仁也热情地关心和帮助本书的编写工作。在此，向他/她们表示衷心的感谢。

作者

目 录

第 1 章 概述.....	1
1.1 软件体系结构.....	1
1.2 当前的软件设计.....	3
1.3 软件设计的层次.....	4
1.4 体系结构与软件的工程设计.....	5
1.5 软件体系结构的知识体系.....	6
习题一.....	7
第 2 章 软件体系结构的研究与发展.....	8
2.1 软件工程设计和软件体系结构.....	8
2.1.1 软件设计的目标.....	9
2.1.2 软件设计中出现的问题.....	11
2.1.3 针对问题提出的软件设计思想.....	12
2.2 什么是软件体系结构.....	14
2.2.1 关于软件体系结构的认识发展.....	14
2.2.2 当前对软件体系结构的认识.....	15
2.3 软件体系结构的意义和目标.....	16
2.3.1 体系结构在软件开发中的意义.....	16
2.3.2 软件体系结构的目标.....	17
2.3.3 与其他软件设计活动的关系.....	17
2.4 软件体系结构的发展.....	18
2.4.1 来自程序抽象方面的发展.....	18
2.4.2 来自软件工程和概念模型方面的发展.....	19
2.4.3 来自体系结构方面的发展.....	20
2.5 软件体系结构的研究范畴.....	21
2.5.1 与体系结构有关的研究：风格、设计模式、框架.....	22
2.5.2 体系结构的观点.....	25
2.5.3 领域结构风格.....	29
2.5.4 体系结构的描述语言.....	29
2.5.5 体系结构的形式化.....	32
2.5.6 体系结构的工具和环境.....	33
2.6 软件体系结构与其他软件技术.....	34
2.7 软件体系结构的现状.....	35
习题二.....	37

第 3 章 软件体系结构的层次性	38
3.1 从建筑学看软件的构成.....	38
3.2 软件的物质基础.....	39
3.3 软件的结构基础.....	42
3.3.1 结构化控制流.....	42
3.3.2 部件连接方式.....	44
3.3.3 结构化连接模式.....	47
3.3.4 基本数据类型.....	48
3.3.5 抽象数据类型.....	49
3.3.6 进程及其运行环境.....	51
3.3.7 分时并发计算.....	54
3.3.8 资源共享/并行同步.....	56
3.3.9 实时系统.....	56
3.3.10 软件结构的四视图观.....	58
3.4 软件的层次结构模型.....	61
3.4.1 计算机网络的体系结构.....	61
3.4.2 计算机操作系统的体系结构.....	63
3.4.3 层次体系结构的对比.....	65
3.5 软件体系结构的层次模型.....	66
3.6 软件体系结构的体系.....	70
习题三.....	74
第 4 章 体系结构的设计原理	75
4.1 体系结构的设计原理.....	75
4.2 软件的非功能特性.....	78
习题四.....	81
第 5 章 部件与连接器	82
5.1 部件.....	82
5.1.1 部件及其作用.....	82
5.1.2 部件的实现形式.....	82
5.1.3 部件的表达形式.....	83
5.1.4 部件的类别.....	84
5.1.5 部件的特性.....	84
5.2 连接.....	86
5.2.1 连接的种类.....	87
5.2.2 连接的实现机制.....	87
5.2.3 连接的协议.....	88
5.2.4 连接的特性.....	88
5.2.5 连接的不匹配及其解决方法.....	90

5.3 连接器.....	91
5.3.1 连接器及其作用.....	91
5.3.2 连接器的层次性.....	91
5.3.3 连接器的表达形式.....	92
5.3.4 连接器的类别.....	93
5.3.5 连接器的特性.....	94
习题五.....	96
第6章 体系结构的一般描述.....	97
6.1 主程序与子过程.....	97
6.2 数据抽象和面向对象设计.....	98
6.2.1 基本要点和特征.....	99
6.2.2 对象的实现类型.....	99
6.2.3 设计时的考虑.....	100
6.2.4 对象的运行机制.....	100
6.2.5 面向对象的优缺点.....	101
6.3 层次结构.....	101
6.3.1 适应的设计问题.....	102
6.3.2 设计结构.....	102
6.3.3 层次结构的特性.....	103
6.3.4 层次结构的实现.....	104
6.3.5 应用实例.....	106
6.3.6 层次结构的变种.....	107
6.3.7 层次结构的应用实例.....	107
6.3.8 层次结构的优缺点.....	108
6.4 形式化描述: 类属理论(Category Theory).....	109
6.4.1 基本概念.....	109
6.4.2 代数规范(Algebraic Specification).....	111
6.4.3 类属规范 Spec.....	111
6.4.4 类属规范 P-Spec.....	113
6.4.5 函数型体系结构理论.....	116
6.4.6 进程代数体系结构理论.....	118
6.5 形式化描述: Z 标记语言(Z Notation).....	119
6.5.1 数学基础.....	120
6.5.2 基本概念.....	122
6.5.3 类型构造器: 架构(schema).....	123
6.5.4 应用例.....	126
6.6 化学抽象机模型.....	128
6.6.1 基本概念.....	128

6.6.2 应用例	130
6.7 UniCon	131
6.8 ACME	137
6.8.1 ACME 的核心概念	138
6.8.2 ACME 中体系结构的风格	144
6.8.3 定义 ACME 族(Family)	146
6.9 软件工程设计方法与体系结构描述	148
习题六	149
第 7 章 体系结构的基础结构模式	150
7.1 独享/共享对象	150
7.2 对象的导出成员(Derived Item)	152
7.3 动态约束	153
7.4 惟一对象(Singleton)	155
7.5 引用计数/句柄(Reference Counter/Handle)	156
7.6 汇集(Collection)	159
7.7 循环器(Iterator)	162
7.8 对象关联(Object Association)	166
7.9 注册/观察器(Observer)	168
7.10 责任链(Chain of Responsibility)	170
7.11 聚集(Aggregation)和特性(Property)	173
7.12 策略(Policy/Strategy)	176
7.13 表达式计算和堆栈	181
7.14 对象工厂(Factory)	183
7.15 转换器(Convertor/Builder)	186
7.16 对象的状态/生存周期(State / Life Cycle)	187
7.17 虚拟设备(Virtual Devices)	191
7.18 代理(Proxy)	192
7.19 小结	194
习题七	196
第 8 章 体系结构的设计模式	197
8.1 模型-视图-控制(MVC:Model-View-Controller)	197
8.1.1 界面设计的可变性需求	197
8.1.2 MVC 的结构	198
8.1.3 MVC 的实现	200
8.1.4 MVC 的变化及其他	203
8.2 网络通信套接(Windows Sockets)	205
8.2.1 Windows Sockets 规范和结构关系	205
8.2.2 Windows Sockets 的结构特性	210

8.2.3 WinSock 控件.....	213
8.3 串行通信.....	216
习题八.....	220
第 9 章 运行调度层体系结构.....	221
9.1 进程通信.....	221
9.1.1 信号.....	221
9.1.2 管道.....	224
9.1.3 FIFO 管道.....	226
9.1.4 IPC 进程间消息通信.....	228
9.2 Windows 2000 进程管理和控制的内核对象.....	230
9.2.1 内核对象.....	230
9.2.2 进程间共享内核对象.....	231
9.3 Windows 2000 的作业.....	232
9.4 Windows 2000 的进程.....	234
9.4.1 进程的工作属性.....	235
9.4.2 创建进程.....	236
9.4.3 子进程.....	237
9.4.4 终止进程.....	238
9.5 Windows 2000 的线程.....	238
9.5.1 创建线程.....	239
9.5.2 线程的运行和调度.....	240
9.5.3 多线程运行的问题.....	242
9.5.4 线程同步机制.....	243
9.5.5 使用内核对象的线程同步机制.....	245
9.6 线程池.....	251
习题九.....	253
第 10 章 资源及管理层次体系结构.....	254
10.1 消息机制与隐式调用.....	254
10.1.1 事件与消息.....	254
10.1.2 消息的处理.....	256
10.1.3 消息过滤器.....	257
10.1.4 消息机制的特点.....	258
10.2 动态链接库.....	258
10.2.1 动态链接库.....	259
10.2.2 动态链接库的作用.....	260
10.2.3 何时使用动态链接库.....	261
10.2.4 动态链接库的建立和使用.....	262
10.3 注册表.....	262

10.3.1	注册表	262
10.3.2	注册表信息的访问	264
10.4	剪贴板	266
10.5	过程调用/本地过程调用 LPC/远程过程调用 RPC	268
10.5.1	LPC 和 RPC 及其实现	268
10.5.2	RPC	269
10.5.3	LPC 和 RPC 实现的考虑	270
10.5.4	使用 WinSock 实现的 RPC 和消息传递	270
10.6	开放数据库互连: ODBC	271
10.6.1	ODBC 及其原理	271
10.6.2	ODBC 的体系结构	272
10.6.3	ODBC 的驱动程序类型及结构	274
10.6.4	ODBC 一致性级别	274
	习题十	275
第 11 章	系统结构模式层体系结构	276
11.1	管道-过滤器	276
11.1.1	管道-过滤器结构	276
11.1.2	管道-过滤器的特性	277
11.1.3	管道-过滤器的类型	278
11.1.4	管道-过滤器的优缺点	278
11.2	实时与连续计算	279
11.2.1	实时与连续计算的系统结构	280
11.2.2	控制系统概念	280
11.2.3	系统类型	281
11.2.4	系统的设计和实现	281
11.3	客户/服务器结构	283
11.3.1	客户/服务器概述	283
11.3.2	客户/服务器的结构分析	284
11.3.3	客户/服务器的连接	286
11.3.4	客户/服务器的运行机制	288
11.4	仓库系统: Microsoft SQL Server 的 C/S 体系结构	290
11.4.1	SQL Server 的结构	290
11.4.2	服务接口	295
11.4.3	线程池和线程的多任务处理	296
11.5	黑板系统	297
11.5.1	黑板系统的结构	297
11.5.2	HEARSAY-II 和其他黑板结构应用	298
11.5.3	黑板结构的变体	299

11.6	解释器/虚拟机.....	300
11.6.1	解释器的组成.....	300
11.6.2	解释器的工作机制.....	300
11.7	COM/DCOM 组件.....	301
11.7.1	COM 由来.....	301
11.7.2	COM 结构.....	302
11.7.3	COM 接口.....	304
11.7.4	COM 类型.....	306
11.7.5	COM 运行.....	308
11.7.6	COM 扩展.....	311
11.8	独立组件体系结构及其通信机制.....	313
11.8.1	独立组件体系结构.....	313
11.8.2	组件的消息机制.....	314
11.8.3	从 COM 机制看组件与客户的通信.....	317
11.9	WWW 万维网.....	317
11.9.1	libWWW 的构成.....	318
11.9.2	应用 libWWW 的万维网的 C/S 体系结构.....	319
11.9.3	公网关接口 CGI.....	320
11.9.4	应用 Jigsaw 的新体系结构.....	321
11.10	WWW 浏览器.....	322
11.11	微核(MicroKernal).....	326
11.11.1	需求和解决方案.....	326
11.11.2	系统结构.....	327
11.11.3	动态特性.....	328
11.11.4	实现过程.....	329
11.11.5	微内核的变异及其他.....	331
	习题十一.....	333
第 12 章	领域应用层体系结构.....	334
12.1	IBM San Francisco 框架.....	334
12.2	IBM San Francisco 的体系结构.....	336
12.2.1	层次构成和抽象.....	336
12.2.2	平台无关性.....	338
12.3	IBM San Francisco 基础层.....	339
12.3.1	Entity 实体.....	340
12.3.2	Dependent 对象及其他.....	341
12.4	IBM San Francisco 公共业务对象层.....	343
12.5	IBM San Francisco 核心业务处理层.....	345
12.6	IBM San Francisco 的设计模式.....	347

12.7 IBM San Francisco 系统支持.....	349
12.7.1 配置和服务器管理	349
12.7.2 安全配置及其他	352
12.8 IBM San Francisco 应用的开发方法.....	353
12.9 MetaH 嵌入式软件体系结构描述语言和工具.....	355
习题十二.....	357
第 13 章 体系结构应用例.....	358
13.1 汽车车速控制问题的软件体系结构.....	358
13.2 DOS 操作系统中的汉字处理.....	362
13.3 WINDOWS 操作系统中的汉字处理.....	364
13.4 电力调度网损数据管理系统.....	365
13.4.1 总体结构	365
13.4.2 系统的部件和连接描述	367
13.4.3 系统运行结构图	371
13.5 远程数据传输 RDT(Remote Data Transmission).....	372
13.6 体系结构的混合.....	374
习题十三.....	376
结束语：开放的体系结构研究.....	377
参考文献.....	379

第1章 概述

在诸如建筑和计算机硬件设计中，体系结构是指整个系统构成的基本和主体形态。在一个发展成熟的领域中，这种结构成为建立和考察系统的总体指导或基本出发点。例如，桥梁的形态具有拱形、板式支撑、吊索、斜拉式等。计算机硬件的主体结构形态具有单处理器、多处理器、并行计算、网络计算等。

类似地，软件体系结构是软件在设计构成上的基本、可供设计选择的形态和总体结构。具体地说，就人们现今已经熟悉的软件概念看，软件设计中可供选择的结构形态有：过程(procedure)、包(package)、对象(object)、客户/服务器(client/sever)、分布式(distributed)、可视控件(visual controls)、部件(component)、解释器(interpreter)、浏览器(browser)等。

每一个结构概念都传达了领域的特定设计问题、解决问题的特定方法、方法的实现构成和它们之间的关系、结构的性能和优缺点、考察结构性能的方法等。

有关领域的发展越是成熟，人们对以上结构概念各个方面的认识就越是全面和深刻。表现在工程实践中的是工程的设计和 implement 以及质量控制更加系统、规范，获得的产品结构更加优良合理，质量和服务水平更高，而成本更低。

虽然在经历了“软件危机”的痛苦发展之后，来自不同应用领域的软件研究和工程应用总结出大量的有价值的知识，建立了功能强大的各类设计开发思想和工具，然而，当今软件的设计理论特别是技术，还远没有达到像建筑领域一样成熟发展的阶段。

随着计算机硬件技术的飞速发展，对计算机软件的功能、结构和复杂性提出了更高的要求。随着软件复杂性的不断增加，在软件的设计中，软件的局部和整体系统结构方面，已经越来越显出其重要性，甚至超过了软件算法和数据结构这些常规软件设计的概念。软件体系结构概念的提出和应用，说明了软件设计技术在高层次上的发展并走向成熟。

当今的软件设计开始摆脱传统的“手工作坊式”、单纯经验的个体开发方式，开始走向系统和规范化的工业生产过程。

1.1 软件体系结构

1. 软件体系结构

软件体系结构定义了软件的局部和总体计算部件的构成，以及这些部件之间的相互作用关系。部件包括诸如客户、服务器、数据库、过滤器、程序包、过程、子程序等一切软件的组成成分。相互作用关系包括诸如过程调用、共享变量访问、消息传递等。相互作用也包括具有十分复杂的语义和构成的关系，诸如客户/服务器的访问协议、数据库的访问协

议、网络的传输协议、异步事件的映射等。

除了描述系统的构成和结构关系外，在系统的功能需求方面，体系结构还表达了系统需求和构成之间的对应关系，这为系统的设计提供了分析和评价的依据。在系统宏观层面上，人们所关心的是系统的非功能需求方面的内容，诸如容量、数据吞吐量、一致性、兼容性、安全性、可靠性等，这些在体系结构中也都表达了出来。

一般地说，体系结构清楚地表达了系统的构成部件以及它们之间的作用关系和语义。这些部件又可以用来构成更大、更复杂的部件或系统。在理想的情况下，体系结构描述的各个组成成分都是被独立地定义的，因此，可以在不同的场合中得到重用。体系结构将为它们建立结构和功能规范，经过进一步完善，可以将它们组成更大、更复杂的系统，并且是可以运行的软件系统。由于系统每一层次和每一部分的组成结构是明确规范了的，因此为整个系统的功能和非功能特性的分析提供了全面的和具体的根据。

总的来看，软件体系结构是由结构和功能各异、相互作用的部件集合，按照层次构成的。它包含了系统基础构成单元、它们之间的作用关系、在构成系统时它们的合成方法以及对合成约束的描述。

2. 体系结构的类别

服务于软件开发的阶段，依据层次和细节程度，体系结构可分为概略型、需求型和设计型。

概略型是上层宏观结构的描述，反映系统最上层的部件和连接关系。

需求型是对概略体系结构的深入表达，以满足用户功能和非功能需求的表达为主，通常需要对概略层的部件和连接进行深层的描述。

设计型体系结构是从设计实现的角度对需求结构的更深层的描述表达。在此类型的结构中，需要从不同的侧面/视图，采用各种表达图示和说明，设计系统各个层面的各个部件和连接的结构。该层面的体系结构将直接服务于系统的实现和性能分析。

3. 体系结构的重要性

体系结构的重要性在于它决定了一个系统的主体结构、宏观特性和具有的基本功能及其特性。正如战争的胜负首先在于战略的策划；大型建筑物设计成功的关键首先在于主体结构。同样，复杂软件设计的成功与否在于软件系统的宏观上层结构设计正确和合理性。

因此，软件体系结构是整个软件设计成功的基础和关键所在。它的作用可以表现在软件设计开发的各个阶段。

(1) 在项目规划阶段，粗略的体系结构是进行项目可行性、工程复杂性、工程进展、投资规模、风险预测等重要根据。

(2) 在项目需求分析阶段，需要从需求出发建立更深入的体系结构描述。这时的体系结构，是开发商和客户之间进行需求交互的表达形式，也是交互所产生的结果。通过它，可以准确地表达用户的需求，以及设计对应需求的解决方法，并考察终结系统的各项性能。

(3) 在项目设计阶段，需要从实现的角度对体系结构进行更深入的分解和描述。

(4) 在项目实施阶段，体系结构的层次和部件是建立开发人员的组织和分工、协调开

发人员关系的依据。

(5) 在项目的评估阶段，体系结构是性能测试和评价的依据。

(6) 在项目维护升级阶段，对软件的任何扩充和修改都需要在体系结构的指导下进行，以维持整体设计的合理性和正确性以及性能的可分析性，并为维护升级的复杂性和代价分析提供依据。

在任何一个有规模的从事现代软件产品开发的单位或部门中，都需要在系统工程师的层次之上，再配备体系结构分析师。他的作用是从战略和宏观上把握软件结构的组成、发展和实现技术。对他的素质要求较高，应该能够全面掌握软件设计的理论、分析技术和实现技术以及发展方向。

1.2 当前的软件设计

尽管人们已经普遍认识到，体系结构的设计选择对于软件的长远成功是至关重要的，但在现实实践中，软件体系结构的理论和工具还没有形成系统、规范的概念和描述。人们普遍使用自己定义的符号、图形、线条、说明等来描述自己的系统设计。甚至在表达那些人们普遍接受的设计部件时，不同设计者也常采用各自的表述。其实，即使使用发展成熟的结构分析和设计方法，不同设计者在图形符号和说明上也经常采用各自的习惯做法。

初看起来，当前体系结构描述中存在的这些不规范性，似乎说明了体系结构描述对于软件设计来说是无足轻重的。然而，这种现象的存在是由两方面原因造成的。

首先，由于体系结构所处背景、使用环境、开发目的、历史发展的不同，在长期的工程实践中，软件设计人员和公司已经形成自有的概念、用语和软件系统的设计表述。这种习惯势力的影响在众多领域中都可以见到。当前体系结构的研究已经说明，体系结构的表达是多层次、多方面的，虽然已经提出了多种体系结构的表达视图，但目前还没有形成统一的结构表达方法。所以，目前采用的各种系统结构表达，对于体系结构的总体目标而言都是不全面的。特别是传统的系统分块结构和功能分析，都不能全面和深入把握系统的构成和实现。

其次，对于实际系统的构成而言，高层次的体系结构反映的是它们之间较粗的、抽象的关系。然而，系统代码的构成细节，对于理解诸如吞吐量、通信模式、控制结构、可扩展性等系统层面的概念，已经提供了直接而自然的分析框架。体系结构的描述不过是把这特性更加表面化。虽然这并不能说明不需要体系结构的形式化描述和更加严格的分析方法，但是在已经完成了代码的设计后再建立体系结构描述，除了必要的结构抽取和分析总结作用，的确有点画蛇添足的感觉。

造成这种现象的原因，在于传统的软件设计是开始于总体概念和代码的。说到底，这是由于体系结构方法和工具还没有成熟到产生巨大吸引力的程度所造成的。设想一下，如果系统的总体设计开始于可供分析、实施、运行、维护的，具有足够深入的描述层次的体系结构，情况又会怎么样呢？

关于软件体系结构的现实是，我们还远未形成统一可接受的认识，更不用说有关的系统全面的理论和工具了。但体系结构对软件设计开发全生命周期具有的宏观指导意义和作用，决定了它将是解决软件复杂性和工程设计困难的下一个必由之路。所以，有必要对它进行研究，并规范化。

人们设想有一天，软件的设计将从宏观需求开始，落实在不同层次和细节的体系结构描述，经过全面的实现选择，最后根据体系结构的描述就能够直接生成可考察和实用的运行系统。随着体系结构理论和应用技术研究的发展成熟，这一天一定会到来的。

1.3 软件设计的层次

系统设计是由许多层次构成的。由于每一个层次处理的是不同的设计问题，因此，有必要对这些层次进行严格的区分。在每一个层次上，都存在以下三个方面：构成系统的部件，原始的或合成的部件；允许由部件合成系统(非原始部件)的合成规则；为系统提供语义的行为规则。由于它们在不同的层次上是不相同的，因而在不同的层次上，就需要有不同的概念、不同的设计问题、不同的分析技术。因此，每一层次的设计本质上可以独立地进行，但高一级的设计是建立在低一级设计之上的。

计算机硬件系统设计的层次概念是人们所熟悉的。首先，每一层次处理不同的内容，不同的结构规则指导着使用不同部件的设计；随之而来的是采用不同的概念、分析技术和设计问题。其次，每一层次都有其子结构，在每一层次上都发生在其层次上所使用的部件及结构的抽象和合成。最后，在相邻的两个层次之间都存在着部件之间确认的转换关系。

软件设计也有其层次性。如下是其三个层次的划分：

(1) 结构级：包括与部件相关联的系统总体性能。在此，部件指模块，模块的相互关联通过多种方法处理，操作算子指导着从子系统到系统的合成。

(2) 代码级：包括算法和数据结构的选择。在此，部件指诸如程序语言原语的数值、字符、指针以及控制的进程等。原始操作算子是程序语言提供的算术、逻辑和数据操作。合成机制包括记录、数组、过程等。

(3) 执行级：包括存储器的映射、数据格式配置、堆栈和寄存器的分配。在此，部件指硬件所提供的位(bit)模式。操作算子和合成是由代码描述的。

从 20 世纪 60 年代到 80 年代，软件开发者注意力放在了程序语言的层次上。现今，软件的代码和执行层次的问题已经彻底解决了，然而，对结构级的理解一直都还停留在直觉和经验上。尽管一个软件系统通常都有文字和图表的说明，但所使用的句法和所表达语义的解释从来都没有得到统一。

我们希望软件体系结构的层次能在理解和精确表达上达到统一。在这个层次上，部件是指程序、模块或系统，连接部件的是丰富的相互关系表达和协议，系统模式指导着系统的合成。