

RUANJIAN TIXI JIEGOU DE YANJIU YU YINGYONG

软件体系结构的研究与应用

◎ 武装 程鸿 著 ◎



科学技术文献出版社
SCIENTIFIC AND TECHNICAL DOCUMENTATION PRESS

软件体系结构的研究与应用

武装 程鸿 著



科学技术文献出版社
SCIENTIFIC AND TECHNICAL DOCUMENTATION PRESS

· 北京 ·

图书在版编目 (CIP) 数据

软件体系结构的研究与应用 / 武装, 程鸿著. —北京: 科学技术文献出版社, 2014.10 (2015.3重印)

ISBN 978-7-5023-7648-2

I. ①软… II. ①武… ②程… III. ①软件—计算机体系结构—研究
IV. ①TP31

中国版本图书馆 CIP 数据核字 (2012) 第 275700 号

软件体系结构的研究与应用

策划编辑: 丁坤

责任编辑: 刘绳恩

责任校对: 张燕育

责任出版: 张志平

出版者 科学技术文献出版社

地址 北京市丰台区右安门内大街 57 号 100038

编务部 (010) 58882938, 58882087 (传真)

发行部 (010) 58882868, 58882874 (传真)

邮购部 (010) 58882873

官方网址 www.stdp.com.cn

发行者 科学技术文献出版社发行 全国各地新华书店经销

印刷者 北京厚诚则铭印刷科技有限公司

版次 2014 年 10 月第 1 版 2015 年 3 月第 2 次印刷

开本 787×1092 1/16

字数 308 千

印张 14

书号 ISBN 978-7-5023-7648-2

定价 48.00 元



版权所有 违法必究

购买本社图书, 凡字迹不清、缺页、倒页、脱页者, 本社发行部负责调换

PREFACE

前 言

随着软件技术的发展，软件规模的扩大，软件开发周期的缩短，软件行业分工的细致，市场竞争的激烈，软件开发商必须要快速分析并实现软件产品的生产。当今，软件的淘汰速度是非常快的，软件设计问题也已经超越了数据结构和算法问题的范围，好的软件就应该拥有较好的扩展性、伸缩性、适应性、稳定性和重用性。为了满足用户日新月异、千变万化的需求，好的软件就必须为变化而设计。不断变化的需求、复杂的业务流程、领域知识的缺乏、许多不可避免的因素都会导致软件变化的发生，所以要确认软件中变化和不变的因素，进行分层处理。软件体系结构的出现，极大地满足了多个应用领域的要求，使得各种技术形成的软件体系结构可以最大程度地进行重用。同时引出了大规模软

件开发所面临的一系列问题：如何建造面向对象的软件体系，并有效地组织和管理；如何分析、提取可复用的体系结构；如何设计适合体系结构的环境；等等。软件结构设计方法能够使软件拥有很好的重用性、扩展性和简洁性。软件体系结构虽脱胎于软件工程，但其形成的同时借鉴了计算机体系结构和网络体系结构中很多宝贵的思想和方法。最近几年软件体系结构研究已完全独立于软件工程的研究，成为计算机科学的一个最新的研究方向和独立学科分支。研究软件体系结构根本目的就是解决好软件的重用、质量和维护问题。软件体系结构作为从软件设计抽象出来的一门新兴学科，目前已经成为软件工程一个重要研究领域。

本书主要包括：引言、软件体系结构形式描述、统一建模语言 UML、软件体系结构建模、企业级应用体系结构技术、软件自适应相关研究、软件体系结构与软件重用方法的研究、面向 Web 服务的领域软件体系结构模型等。相关内容都是在参考了大量相关研究的基础上，结合作者的亲身实践编写而成。

在编写本书的过程中，得到了首都经济贸易大学各级领导和同事的鼓励和支持，特别是信息学院杨一平、马慧和姚翠友三位教授为本书编写提供了重要帮助。借本书出版之际，向他们表示真诚的感谢。本书得到首都经济贸易大学校级科研项目(2013XJG022)《虚拟环境中自然现象模拟关键技术的研究》的资助。

由于编写时间仓促，作者水平有限，书中不当甚至错误之处在所难免，诚恳期待广大读者批评指正。

武 装
首都经济贸易大学
2012 年 10 月

目 录

第一章 引言.....	1
1.1 软件体系结构的起源和发展.....	1
1.2 软件体系结构的定义.....	2
1.3 软件体系结构的重要性.....	2
1.4 软件体系结构应用现状及发展方向.....	3
1.4.1 应用现状.....	3
1.4.2 研究热点.....	5
1.4.3 发展方向.....	5
1.5 软件体系结构研究范畴.....	6
1.5.1 软件体系结构描述语言.....	6
1.5.2 软件体系结构风格.....	7
1.5.3 软件体系结构的形式化研究.....	8
1.6 软件体系结构研究的两种途径.....	9
1.7 软件体系结构研究不足之处及发展趋势.....	10
第二章 软件体系结构形式描述.....	11
2.1 软件体系结构作用.....	11
2.2.1 模型和视图.....	12
2.2.2 描述语言.....	14
2.2.3 体系结构风格.....	15
2.2.4 软件体系结构的形式化研究.....	18
2.3 体系结构行为描述概述.....	21
2.4 XYZ 活动图.....	23

第三章 统一建模语言UML	26
3.1 标准建模语言UML的概述	26
3.1.1 标准建模语言UML的出现	26
3.1.2 标准建模语言UML的内容	27
3.1.3 标准建模语言UML的主要特点	30
3.1.4 标准建模语言UML的应用领域	30
3.2 标准建模语言UML的静态建模机制	31
3.2.1 用例图	31
3.2.2 类图、对象图和包	34
3.3 标准建模语言UML的动态建模机制	40
3.3.1 消息	40
3.3.2 四种图的运用	42
3.4 UML的扩展机制	43
3.5 UML简介	45
3.6 UML描述软件体系结构的几种方案	47
3.7 基于UML的软件体系结构六视图描述	48
3.7.1 六视图的体系结构模型	49
3.7.2 六视图及其相互关系	50
3.7.3 六视图体系结构构造过程	51
第四章 软件体系结构建模	60
4.1 软件体系结构建模概述	60
4.2 软件体系结构的描述	61
4.3 软件体系结构风格	67
4.4 软件质量属性	69

4.5	多维软件体系结构模型.....	71
4.6	质量属性驱动的体系结构建模.....	72
4.7	对软件体系结构模型的评估.....	75
4.8	面向方面软件体系结构的UML描述.....	77
4.8.1	构件.....	77
4.8.2	连接件.....	78
4.8.3	方面.....	79
4.8.4	约束.....	79
4.8.5	软件体系结构元素到UML元素的映射规则.....	80
4.8.6	基于UML面向方面软件体系结构元模型.....	80
4.9	应用实例.....	81
4.9.1	传统软件体系结构模型.....	81
4.9.2	面向方面软件体系结构模型.....	82
4.9.3	两种模型对比.....	83
4.10	面向方面软件体系结构的ADL描述.....	84
4.10.1	构件.....	84
4.10.2	系统连接件.....	85
4.10.3	方面连接件.....	85
4.10.4	方面连接件的编织过程.....	86
4.11	实例.....	86
4.11.1	体系结构的ADL描述.....	87
4.11.2	优势.....	89
4.12	体系结构失配研究.....	89

第五章 企业级应用体系结构技术	92
5.1 企业级应用体系结构.....	92
5.1.1 企业级层次的演化.....	93
5.1.2 现代应用软件架构及其优势.....	95
5.2 J2EE系统架构研究.....	97
5.2.1 J2EE体系结构.....	97
5.2.2 分布式RMI体系结构.....	99
5.3 基于组件开发的软件架构.....	100
5.3.1 CBD软件架构问题研究.....	101
5.3.2 CBD组件类型.....	102
5.4 J2EE技术研究.....	102
5.4.1 组件技术.....	103
5.4.2 服务技术.....	104
5.4.3 通信技术.....	105
5.5 设计模式及其相关问题.....	105
5.6 当前流行的J2EE WEB应用框架.....	107
5.6.1 Apache Struts框架.....	107
5.6.2 JATO框架.....	108
5.6.3 WAF框架.....	109
5.7 基于企业参考模型的软件体系结构.....	109
5.7.1 企业参考模型.....	109
5.7.2 企业参考模型的意义.....	110
5.8 企业参考模型的技术理论.....	111
5.8.1 业务组件理论.....	111

5.8.2	大批量定制理论.....	112
5.8.3	层级理论.....	113
5.9	支持体系结构开发的企业参考模型.....	114
5.10	基于企业参考模型的软件体系结构的基本路线.....	118
5.10.1	参考模型元模型.....	120
5.10.2	基于企业参考模型的软件体系结构元模型.....	122
5.11	基于企业参考模型的软件体系结构.....	124
第六章	软件自适应相关研究.....	126
6.1	软件自适应特征分类.....	126
6.1.1	感知环节的特征分类.....	127
6.1.2	决策环节的特征分类.....	128
6.1.3	执行环节的特征分类.....	129
6.2	软件自适应使能技术.....	130
6.2.1	使能技术.....	131
6.2.2	感知使能技术.....	135
6.2.3	决策使能技术.....	136
6.2.4	执行使能技术.....	139
6.3	相关软件工程项目.....	141
6.3.1	以软件体系结构为中心.....	141
6.3.2	以构件模型设计为中心.....	144
6.3.3	以中间件/软件框架设计为中心.....	146
6.4	群体自适应的初步探索.....	149
6.5	相关研究的总结分析.....	150
6.5.1	相关项目总结.....	150

6.5.2	现有工作所面临的挑战.....	151
6.6	软件体系结构自适应三层框架.....	153
6.6.1	用户层.....	153
6.6.2	适应层.....	154
6.6.3	系统层.....	154
6.7	规划阶段的过程模型.....	154
6.8	基于强化学习的规划决策模型.....	156
6.8.1	五元决策模型.....	156
6.8.2	元组内关系.....	157
6.8.3	Client/Server风格实例.....	158
6.9	强化学习决策算法.....	160
6.9.1	应用强化学习的可行性分析.....	160
6.9.2	SARSA算法.....	162
6.9.3	基于SARSA算法的规划模型.....	162
第七章	软件体系结构与软件重用方法的研究.....	164
7.1	软件重用和软件体系结构的研究背景.....	164
7.1.1	软件重用的意义.....	165
7.1.2	软件重用的研究现状.....	165
7.1.3	软件体系结构与软件重用的联系.....	166
7.2	软件重用的技术途径.....	167
7.2.1	领域分析.....	167
7.2.2	重用再工程(Reuse re-engineering).....	169
7.2.3	特定域软件体系结构(DSSA).....	170
7.2.4	设计样本(Design Pattern).....	171

7.3 软件开发模式演进.....	172
7.3.1 生命周期模式.....	172
7.3.2 快速原型模式.....	173
7.3.3 面向对象的软件开发模式.....	174
7.4 工控领域的组态软件.....	174
7.5 构件化可重构软件系统组态模型.....	176
7.5.1 组态概念的引入.....	176
7.5.2 构件化可重构软件系统组态模型建立.....	178
7.5.3 软件体系结构描述.....	180
7.5.4 构件描述.....	182
7.5.5 系统逻辑组态描述.....	185
7.5.6 CBRSCM可重构特性.....	187
7.6 基于构件化可重构软件系统组态模型的软件开发过程.....	187
7.7 构件化可重构软件系统组态模型特性对比分析.....	189
7.7.1 与当前自动化领域的组态软件对比.....	189
7.7.2 与传统软件开发方法对比.....	190
7.7.3 与当前构件组装技术对比.....	191
第八章 面向Web服务的领域软件体系结构模型.....	193
8.1 面向服务软件体系结构.....	193
8.2 Web服务的领域系统构造过程.....	194
8.3 角色和操作.....	196
8.3.1 角色和角色类.....	196
8.3.2 角色模型.....	197
8.3.3 操作.....	197

8.4	Web服务构件和构件类.....	199
8.4.1	Web 服务构件	199
8.4.2	Web 服务构件类	200
8.5	组合 Web 服务构件.....	201
8.5.1	Web 服务构件组合的相关定义	201
8.5.2	Web 服务构件的交互语义	202
8.6	基于 BPEL4WS 的网格服务组合体系结构.....	203
8.6.1	网格服务.....	203
8.6.2	网格服务组合的参考体系结构.....	204
参考文献.....		206

第一章 引言

“如何将一些较小的子系统合并为一个完整的软件系统”一直是计算机科学领域的一个重要问题。当前有许多种不同粒度水平的处理角度或解决方法,比如,早期的一些软件工程师从机器语言或汇编语言的粒度水平解决软件合成的问题,而成功的软件工程师则采用一个相对较粗的粒度水平——高级语言。软件体系结构和体系结构风格的出现则带来了一个更粗粒度及更抽象的方法来帮助我们开发和集成软件系统。

软件体系结构虽脱胎于软件工程,但它的形成也借鉴了计算机体系结构和网络体系结构中很多成熟的思想和方法。近年来,软件体系结构已经成为计算机科学的一个新的研究方向和独立的学科分支。软件体系结构研究的主要内容涉及软件体系结构描述、软件体系结构风格、软件体系结构评价和软件体系结构的形式化方法等方面。解决好软件的重用、质量和维护等问题,是研究软件体系结构的根本目的。

1.1 软件体系结构的起源和发展

伴随着软件工程方法和技术的逐渐成熟和完善,软件体系结构也由最初模糊的概念发展成一个日渐成熟的技术。

20 世纪 70 年代以前,软件开发基本上都是汇编程序设计。此阶段系统规模较小,很少明确考虑系统结构,一般不存在系统建模工作。70 年代中后期,由于结构化开发方法的出现与广泛应用,软件开发中出现了概要设计与详细设计,而且主要任务是数据流设计与控制流设计。因此,此时软件体系结构已作为一个明确的概念出现在系统的开发中。

20 世纪 80 年代初到 90 年代中期,是面向对象开发方法的兴起与成熟阶段。由于对象是数据与基于数据之上操作的封装,因而数据流设计与控制流设计统一成为对象建模。1997 年出现的统一建模语言 UML 则综合了若干种面向对象方法的精髓,从功能模型、静态模型、动态模型、配置模型来描述应用系统的结构。

90 年代以后则是基于组件的软件开发阶段。该阶段以过程为中心,强调软件开发采用组件化技术和体系结构技术,要求开发出的软件具备很强的自适应性、互操作性、可扩展性和可重用性。在此阶段中,软件体系结构已经作为一个明确的文档和中间产品存在于软件开发过程中,同时,软件体系结构作为一门学科逐渐得到人们的重视,并成为软件工程领域的研究热点。

综上,软件体系结构的发展大致经历了 4 个阶段:无体系结构阶段,以汇编语言进行小规模应用程序开发为特征;萌芽阶段,出现了结构程序设计,以控制流图和数据流图构成软件结构为特征;初期阶段,出现了从不同侧面描述系统的结构模型,以 UML 为优秀代表;高级阶段,以描述系统的高级抽象结构为中心,不关心具体的建模细节,划分了体系结构模型与传统的软件结构的界限,该阶段以“4+1”模型为代表。

1.2 软件体系结构的定义

虽然软件体系结构已经在软件工程领域中有着广泛的应用,但迄今为止还没有一个被大家所公认的定义。许多专家学者从不同角度和不同侧面对软件体系结构进行了刻画,较为典型的定义如下。

(1) D. E. Perry 和 A. L. Wolf 的定义:软件体系结构由具有特定形式的体系结构元素构成,这些元素分为处理元素、数据元素和连接元素 3 类。

(2) D. Garlan 和 M. Shaw 的定义:软件体系结构是软件设计的一个层次,它处理那些超越算法和数据结构的设计,研究总体系统结构设计和描述方法。包括总体组织与全局控制结构、通信协议、同步、数据存取、设计元素的功能分配、物理分布、设计元素的组合、设计方案的选择、评估和实现等。

(3) L. Bass 等人在关于软件品质属性研究中提出:体系结构设计至少可包括应用领域的功能分割、系统结构、结构的领域功能分配 3 个方面。

(4) D. Garlan 和 D. E. Perry 把他们原先的定义修正为:软件体系结构包括系统组件的结构、组件的相互关系、控制组件设计演化的原则和指导三个方面。

尽管各种定义都从不同角度关注软件体系结构,但它们对于描述什么是软件体系结构不存在根本的冲突。相反,它们为软件体系结构划定了一个范围,这个范围涉及了软件体系结构研究的重点问题,如软件体系结构的组成、软件体系结构的构建等。这些定义的核心内容都是软件系统的结构,并且都涵盖了如下一些实体:组件、组件之间的交互关系、连接件、限制或约束、组件和连接件构成的拓扑结构、设计原则与指导方针。

总结各种关于软件体系结构定义的共同特征,可以将软件体系结构综合定义如下:在软件密集的大规模系统或具有类似需求结构的软件产品线的开发中,必须从一个较高的抽象层次来考虑组成系统的组件、组件之间的联系或交互,以及由组件与组件交互形成的拓扑结构。这些要素应该满足一定的限制和约束,遵循一定的设计规则,并能够在一定的环境下进行演化,以支持软件体系结构层次的软件复用。而且,软件体系结构应能反映系统开发中具有重要影响的设计决策,便于各种人员的交流,反映多种关注,据此开发的系统能完成系统既定的功能和性能需求。

1.3 软件体系结构的重要性

体系结构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,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。体系结构技术的研究,使软件复用从代码复用发展到设计复用和过程复用,它的作用体现在软件生命周期的各个阶段。

(1) 在规划阶段,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、风险预测等的重要依据。

(2) 在需求分析阶段,需要从需求出发建立更深入的体系结构描述。这时的软件体系结构,是软件开发者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

软件体系结构，可以准确地表达用户的需求，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) 在评估阶段，体系结构是软件质量、性能测试和评价的依据。

(6) 在维护升级阶段，对软件的任何扩充和修改都要在体系结构的指导下进行，以维持整体设计的合理性、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。

Bass 给出了为什么软件体系结构如此重要的三个关键理由：

(1) 软件体系结构的表示可以实现对系统感兴趣的所有各方(风险承担者)间的通信和交流。

(2) 体系结构突出了早期设计抉择，这些抉择将对随后的所有软件工程工作产生深远的影响，也对系统作为一个可运行实体的成功有着深远影响。

(3) 体系结构建立了一个相对小的、易于理解掌握的模型，该模型描述了系统如何构成以及其构件如何协同工作。

1.4 软件体系结构应用现状及发展方向

1.4.1 应用现状

(1) 软件体系结构已成为研究热点，但仍处于非形式化的水平

从研究现状来看，当前的研究和对软件体系结构的描述，在很大程度上还停留在非形式化的基础上。软件构架师仍然缺乏必要的工具，这种工具应该是显式描述的、有独立性的形式化工具。

在目前通用的软件开发方法中，其描述通常是用非形式化的图和文本，不能描述系统期望的存在于构件之间的接口，不能描述不同的组成系统组合关系的意义，难以被开发人员理解，更不能用来分析其一致性、完整性等特性。

当一个软件系统中的构件之间几乎以一种非形式化的方法描述时，系统的重用性也会受到影响，在设计一个系统结构过程中的努力很难移植到另一个系统中去。对系统构件和连接关系的结构化假设，没有得到显式的、形式化的描述时，把这样的系统构件移植到另一个系统中去将是有风险的，甚至是不可能的。

(2) 软件体系结构的形式化方法研究

软件体系结构研究如果仅仅停留在非形式化的框图阶段，很难适应进一步发展的需要。为了支持基于体系结构的开发，需要有形式化建模符号、体系结构说明的分析与开发工具。从软件体系结构研究的现状来看，在这一领域近来已经有不少成果，其中比较有代表性的是美国卡耐基梅隆大学的 Robert J. Allen 于 1997 年提出的 Wright 系统。Wright