

# 软件体系结构设计方法的研究

王一宾<sup>1,2</sup>, 李心科<sup>1</sup>

(1. 合肥工业大学, 安徽 合肥 230009; 2. 安庆师范学院, 安徽 安庆 246011)

**摘 要:** 科学合理的软件体系结构是开发一个成功应用系统的必要条件, 对于提高系统的开发效率和质量具有非常重要的意义。首先分析了软件体系结构与相关概念的关系, 接着剖析了软件体系结构的几种主要设计方法的设计过程, 最后指出了这些设计方法的不足之处及下一步可能的研究方向。

**关键词:** 软件体系结构设计; 工件驱动; 用例驱动; 模式驱动; 领域驱动; 需求驱动

中图法分类号 TP311.5 文献标识码 A 文章编号 1000-7024 (2005) 03-0604-04

## Research on method of software architecture design

WANG Yi-bin<sup>1,2</sup>, LI Xin-ke<sup>1</sup>

(1. Hefei University of Technology, Hefei 230009, China; 2. Anqing Teachers College, Anqing 246011, China)

**Abstract:** A scientific and reasonable software architecture is a necessary condition to develop a successful applied system. And it is very important to improve the development efficiency and quality of the system. Firstly, the relation between software architecture and some concepts are analyzed. Then, some software architecture design methods are deeply analyzed. Finally, some shortcomings of these design methods and some promising research directions are indicated.

**Key words:** software architecture design; artifact-driven; use-case-driven; pattern-driven; domain-driven; requirement-driven

## 1 引言

软件体系结构概念的提出和应用, 表明软件工程走向新的发展阶段。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主导作用。新一代的软件工程将是体系结构的工程<sup>[2]</sup>。

软件体系结构设计包括早期的设计决定, 体现了系统的全局结构, 对于整个系统的质量有着决定性的影响。为了确保各种质量因素, 正确地对体系结构进行抽象很有必要。为了获取对软件体系结构设计的抽象, 人们已经提出了许多方法, 我们把这些设计方法分类为工件驱动的方法、用例驱动的方法、模式驱动的方法、领域驱动的方法、需求驱动的方法。

## 2 工件驱动的软件体系结构设计方法

### 2.1 工件和软件体系结构之间的关系

软件过程每经历一个阶段, 就会发生一次知识转换的情况。这种转换是由人来完成的, 这就像是接力一样, 一个人把大脑中的知识以某种方式传递给另一个人, 再由另一个人传递下去, 直至编码人员把这些知识固化在最终的软件中。在软件成型之前, 知识的主要载体是文档和模型, 我们称它们为工件。工件是实用的软件制品, 工件的概念表示某一方法的

工件描述, 是指诸如工件类、工件操作、工件属性等。一般说来, 每种工件都有一套与之相关的试探法, 用来标识相关的工件实例<sup>[2]</sup>。软件体系结构是软件在设计构成上的基本、可供选择的形态和总体结构。可以从方法的工件描述中提取软件体系结构描述。

### 2.2 工件驱动的软件体系结构设计

工件驱动软件体系结构设计过程如图 1 所示。分析和设

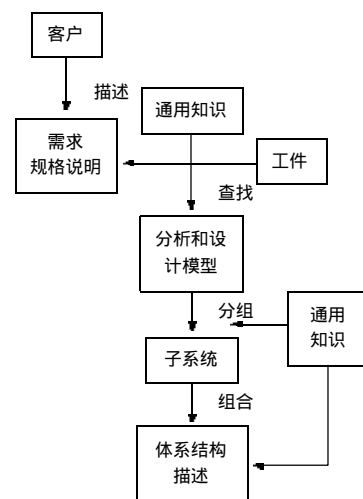


图 1 工件驱动的软件体系结构设计过程

收稿日期: 2004-11-08。 基金项目: 安徽省教育厅自然科学基金项目 (2005KJ364zc); 安庆师范学院科研基金项目 (2003yly06)。

作者简介: 王一宾 (1970-), 男, 安徽安庆人, 硕士生, 讲师, 研究方向为软件工程; 李心科 (1963-), 男, 副教授, 博士, 硕士生导师, 研究方向为软件工程。

计模型及子系统共同表示解决方案抽象,通用知识表示知识领域概念的特殊化。描述表示需求规格说明的描述;查找表示对工件的查找,如系统分析阶段中需求规格说明类,查找过程得到了软件工程师的通用知识的支持,也得到了构成工件的启发式规则的支持,其结果是一组工件实例,用分析和设计模型的概念表示;分组将工件分组为子系统,为单个软件系统的全局结构的开发定义整体体系结构;组合表示软件体系结构由子系统组合而成,这一阶段也用到通用知识概念所提供的支持。

### 2.3 工件驱动的软件体系结构设计方法的不足

工件驱动的软件体系结构设计方法在体系结构开发方面存在以下不足:文本形式的系统含混不清、不够精确、不够完整,因此它作为导出体系结构抽象的来源作用不够;子系统的语义过于简单,难以作为体系结构构件;对子系统的组合支持不足等。

## 3 用例驱动的软件体系结构设计方法

### 3.1 用例和软件体系结构之间的关系

用例是能够向用户提供有价值结果的系统中的一种功能。用例获取的是系统功能需求,所有的用例合在一起构成用例模型,它描述了系统的全部功能。该模型代替了传统的系统功能说明,用例模型所涉及的系统细节最少,而离系统需求最近。用例模型的主要目的是定义系统要完成什么功能,从而使用户和软件工程师达成一致。它用于驱动软件系统的其它开发工作,也是对象建模活动的起点。用例模型由角色类型和用例类型组成。

用例和软件体系结构之间存在着紧密的关系。一方面,软件体系结构受到我们所希望的系统支持的用例的影响,用例驱动软件体系结构的建立。软件体系结构受到以下约束和使能因素的影响,如系统软件、中间件(包括框架)、遗留系统、标准和政策、非功能性要求、分布需求等。另一方面,用例也受到软件体系结构的影响,软件体系结构是在细化阶段的迭代过程中被创建的。

### 3.2 用例驱动的软件体系结构设计

根据用例和软件体系结构的相互关系,我们可以通过迭代的方式来构建软件体系结构,如图2所示<sup>[4]</sup>。首先在很好地了解领域范围的基础上,建立一个临时的软件体系结构,但不考虑具体的用例。接着选取几个重要的用例,并进一步使体系结构能够支持这些用例。然后再选取更多的用例,建立更加完美的体系结构。依次类推,在每一次迭代中,都选取并实现一组用例来确认体系结构。如果必要,再对体系结构进行改进。随着每次迭代的进行,可在所选用例的基础上,进一步实现体系结构的专门应用部分。因此,在通过迭代实现整个系统时,用例有助于我们逐渐完善体系结构。

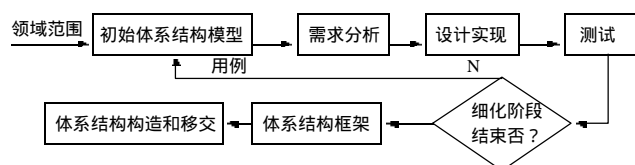


图2 用例驱动的软件体系结构设计过程

一个软件开发项目,可粗略分成初始和细化阶段与构造和移交阶段两大块。初始阶段的主要目标是设定产品应该做什么的范围,并建立初始业务案例,从业务的角度表明项目的可行性。细化阶段的主要目标是建立体系结构基线,捕获大多数需求。构造阶段的主要目标是开发整个系统,并确保产品可以开始移交给客户,即产品达到最初的可操作能力。移交阶段的主要目标是确保得到一个准备向用户社团发布的产品。在这个阶段,需要培训用户如何使用该软件。软件体系结构主要是在细化阶段的迭代过程中发展起来的。每次迭代的进行都要经历5种核心工作流,即需求、分析、设计、实现和测试。虽然每次迭代都要经历需求、分析、设计、实现和测试工作流,但不同阶段的迭代侧重点不同。在初始和细化阶段中,绝大部分工作集中在捕获需求,进行初步的分析与设计上。在构造阶段中,重点则转移到详细设计、实现和测试上。每次迭代在其结束时进行评估。其中的一个目的就是判定是否提出了新的需求,或者现有需求的变化是否影响到后续的迭代。一次迭代产生一个增量结果。一个增量是一次迭代的内部版本与下一次迭代的内部版本之间的差别。

### 3.3 用例驱动的软件体系结构设计的不足

用例驱动的软件体系结构设计方法在体系结构开发方面存在以下不足:难以适度把握领域模型和商业模型的细节;对于如何选择与体系结构相关用例没有提供系统的支持;用例没有为体系结构抽象提供坚实的基础;包的语义过于简单,难以作为体系结构构件等。

## 4 模式驱动的软件体系结构设计

### 4.1 设计模式与软件体系结构之间的关系

软件体系结构描述的是组件及其关系,针对特定的应用领域,设计模式提供了获得高层次的可复用组件的描述方式,是复用成功设计的有效手段。体系结构定义了软件整体的结构、组成、协作及控制参数,基于设计模式设计的软件组件可以是整个体系结构的一部分,可以帮助体系结构框架获得更高层次的设计复用和代码复用,有助于获得无需重新设计就可适用于多种应用的框架结构体系<sup>[5]</sup>。

### 4.2 模式驱动的软件体系结构设计

模式驱动的软件体系结构设计方法从模式导出体系结构抽象。图3表示了该方法的基本过程。需求规格说明表示对问题的规格说明,该问题可以通过模式得以解决,体系结构模式的描述主要由4个概念组成:意图、上下文、问题和解决方案。意图表示使用模式的基本原则;上下文表示问题的产生环境;问题表示上下文环境中经常出现的问题;解决方案是以元素及其关系的抽象描述的形式来表示对问题的解决方案。体系结构模式表示应用的结果。组合表示在导出体系结构描述时,体系结构模式之间的相互协作。在许多体系结构设计方法中,都包括作为一个子过程的模式驱动的体系结构设计方法。

### 4.3 模式驱动的软件体系结构设计的不足

模式驱动的软件体系结构设计方法在体系结构开发方面存在以下不足:在处理范围广泛的体系结构问题时,模式库可能不够充足;对模式的选择仅依靠通用知识和软件工程师的

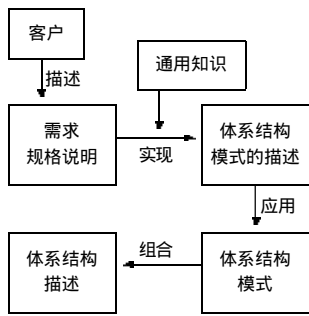


图3 模式驱动的软件体系结构设计过程

经验,模式的应用并不是一个简单直接的过程,它需要对问题进行全面分析;对于模式的组合没有提供很好的支持等。

## 5 领域驱动的软件体系结构设计

### 5.1 领域分析与软件体系结构之间的关系

领域分析与软件体系结构同时作为软件系统开发的早期形式,本身存在着很强的连续性和依赖关系。从软件工程观点来看,虽然软件体系结构开发是整个项目开发过程的起点,而且在软件生命周期中,软件体系结构设计也处于前列,但是软件体系结构的建立依赖于领域分析。领域分析不仅是对领域内一系列相似系统可重用信息的识别、提取、收集和整理,更是对这些可利用信息进行的整理和再加工,使这些可重用信息规范化、标准化,从而更有利于重用,软件体系结构模型的建立参考并使用这些规范化的可重用信息,使基于软件体系结构的新系统的开发结果可以直接加入领域分析,进一步充实领域模型,形成良性循环<sup>[6]</sup>。

### 5.2 领域驱动的软件体系结构设计

可以把领域驱动的软件体系结构(DSSA)看成是多系统范围内的体系结构,即它是从一组系统中导出的,而不是某一单独的系统。图4表示了DSSA方法的基本过程<sup>[2]</sup>。DSSA方法的基本工件是领域模型、参考需求和参考体系结构。DSSA方法从领域分析阶段开始,面向一组有共同问题或功能的程序。这种分析以场景为基础,从中导出功能需求、数据流和控制流等信息。领域模型包括场景、领域字典、上下文图、实体关系图、数据流模型、状态转换图以及对象模型。除了领域模型之外,还定义了参考需求,它包括功能需求、非功能需求、设计需求、实现需求,而且它主要关注解决方案空间。领

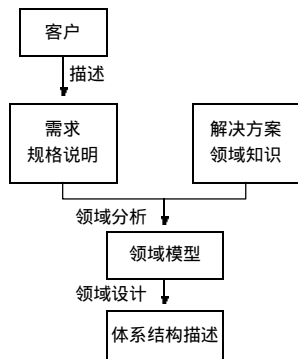


图4 领域驱动的软件体系结构设计过程

域模型和参考需求被用于导出参考体系结构。DSSA过程明确区别参考体系结构和应用体系结构。参考体系结构被定义为用于一个应用系统族的体系结构,应用体系结构被定义为用于一个单一应用系统的体系结构。应用体系结构是从参考体系结构实例化或求精而来的。

### 5.3 领域驱动的软件体系结构设计的不足

领域驱动的软件体系结构设计方法在体系结构开发方面存在以下不足:问题领域分析在导出体系结构抽象方面效果较差,解决方案领域分析不够充分等。

## 6 需求驱动的软件体系结构设计

### 6.1 需求分析与软件体系结构之间的关系

需求分析主要关心问题空间,其主要目的是限定软件系统所解决的问题范围,而软件体系结构是关于所求解问题的解决方案空间,它的主要目标是确定问题解决方案的结构。软件体系结构是软件需求、业务技术流程和社会环境因素的整体高层规划,在体系结构设计过程中,反映用户需求并根据需求指导软件体系结构设计选择是一个相互作用的过程。

### 6.2 需求驱动的软件体系结构设计

传统的软件开发都是由编程范型驱动,所谓需求驱动是指软件体系结构设计基于需求分析的结果,因此需求驱动下软件体系结构设计主要是描述所解决问题和解决方案之间的动态关系,其设计过程如图5所示<sup>[7]</sup>。

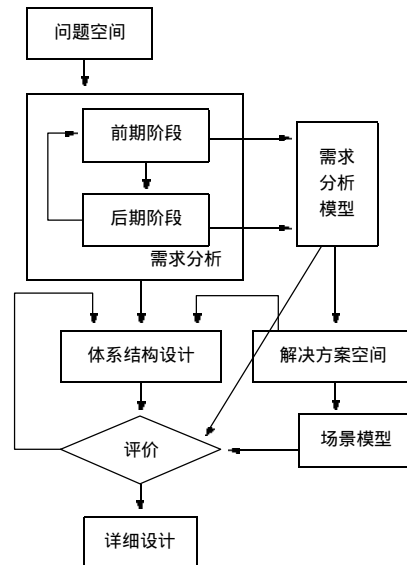


图5 需求驱动的软件体系结构设计过程

需求分析首先在于问题的描述,可并行建立目标模型和场景模型。问题空间定义了所解决的问题,一般是组织的高层业务目标,其特点是具有较高的抽象性。需求分析可分为前期阶段和后期阶段,前期阶段目的是通过研究组织环境来理解问题,主要成果是产生一个组织模型,该模型包括相关行为者及其目标和内部依赖关系,这一阶段的工作基本上是非形式化的,主要考虑组织和非功能需求,强调理解系统需求底层的原理和需求的根本原因。后期阶段主要在操作环境中描述系统相关功能和属性,通常强调需求完整性、一致性及功能

需求的自动验证等。体系结构设计是根据组件或子系统之间的数据、控制及其它依赖关系描述的系统全局结构设计,该结构描述系统如何分解为组件,并且各组件如何相互交互。如前所述,把软件体系结构看作是抽象的组件通过连接交互的整体,基于细化的需求模型,可以确定某个组件要满足的非功能需求(用软目标表示)和功能需求(用场景表示)。通过描述组件和连接的抽象属性,得到软件体系结构的抽象模型,再从需求分析模型确定的解决空间中发现体系结构设计方案,从而得到具体的体系结构实例。然而,此时还不能肯定这个实例是否满足需求分析模型,需要对体系结构进行评价,即选定某解决方案的同时建立场景模型。

### 6.3 需求驱动的软件体系结构设计的不足

需求驱动的软件体系结构设计方法在体系结构开发方面存在以下不足:很难对非功能需求进行建模;很难确定最佳求解问题与求解空间;需要设计分析与设计工具的原型系统等。

## 7 结 语

综上所述,每一种设计方法都有其自身的优缺点。据估计,使用基于体系结构的设计,整个生命周期的代价可以减少约10%,并且可以通过严格的体系结构分析大大减少设计错误和风险,提高产品的可变性和可维护性。由于认识到好的体系结构是决定软件系统成败的重要因素,软件体系结构已经成为国内外软件工程学者研究的热点。随着软件系统规模

的不断扩大,软件体系结构的设计显得越来越重要。由于软件体系结构研究刚刚起步,现有的研究成果尚不能满足工程需要,尤其需要进一步对基于软件体系结构在软件开发中的应用进行研究,使得软件体系结构在软件开发中发挥重要的作用。

### 参考文献:

- [1] Shaw M, Galan D. Software architecture: Perspectives on an emerging discipline[M]. Prentice Hall, 1996.
- [2] 冯冲,江贺,冯静芳. 软件体系结构理论与实践[M]. 北京:人民邮电出版社, 2004.
- [3] 万建成,卢雷. 软件体系结构的原理、组成与应用[M]. 北京:科学出版社, 2002.
- [4] 刘韶涛,余金山. 用例驱动的软件体系结构构建[J]. 华侨大学学报, 2003, 24(2): 208-212.
- [5] 周小健,余冬梅,张聚礼. 基于设计模式的软件体系结构研究[J]. 甘肃工业大学学报, 2003, 29(4): 99-102.
- [6] 彭宏,王锦,张激. 基于域分析的软件体系结构研究[J]. 计算机工程, 2002, 28(2): 120-122.
- [7] 琚川徽,程勇,袁兆山. 需求驱动的软件体系结构设计[J]. 合肥工业大学学报, 2002, 25(3): 350-354.
- [8] 韦群,熊璋,赵芳. 软件体系结构开发方法及其应用[J]. 计算机工程与设计, 2003, 24(3): 77-80.

(上接第 576 页)

数的增加而减少。因为再附着点距离 $z_{av}$ 和 $R_c$ 数几乎为比例关系,式(8)导出的再附着点距离变化所对应的时间常数 $\tau_r$ 几乎为一正值。

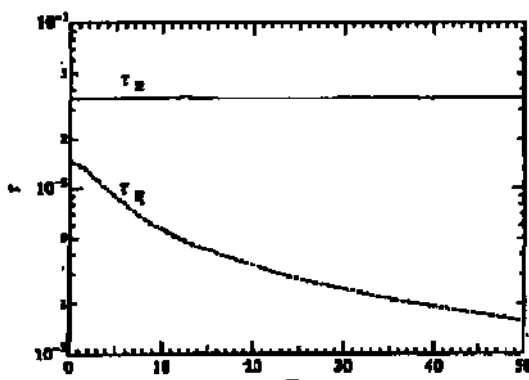


图 7 时间常数 $\tau_r$ 和 $R_c$ 数的关系

## 4 结 论

在压力差 $\Delta p$ 种种阶跃变化的情况下讨论了流动参数与 $R_c$ 数的关系,导出了简单的计算式,画出了直观的关系图。通过数值计算得出以下结论:再附着点距离 $z_{av}$ 和 $R_c$ 数几乎为比

例关系。雷诺数 $R_c < 10$ 时,流量系数 $C_d$ 随着 $R_c$ 数增加而增加较快,雷诺数 $R_c > 10$ 后流量系数 $C_d$ 的变化渐慢。惯性长 $l$ 几乎随 $R_c$ 数成正比地增加。时间常数 $\tau_r$ 随着 $R_c$ 数的增加而减少。 $R_c$ 数增加而时间常数 $\tau_r$ 几乎为一正值。

### 参考文献:

- [1] Gosman A D, Vlachos N S, Whitelaw J H. Low Reynolds-number pipe flow in the vicinity of three-dimensional obstacles[J]. Journal Mechanical Engineering Science, 1979, 21(5): 335-343.
- [2] 金忠青. N-S 方程的数值解和紊流模型[M]. 南京:河海大学出版社, 1989.
- [3] Keith T G, James E A John. Calculated orifice plate discharge coefficients at low Reynolds numbers[J]. Trans ASME, 1977, 99(2): 424-425.
- [4] 程平,李受人,程耕国. 液流通过节流孔的惯性长的数值研究[J]. 水动力学研究与进展, 2003, 18(3): 302-305.
- [5] Sumita T. CFD analysis of flow through a spool valve[J]. Journal of the Japan Hydraulics & Pneumatics Society, 1991, 23(2): 138-142.
- [6] Funk J E, Wood D J, Chao S P. The transient response of orifices and very short lines[J]. Trans ASME, Basic Eng, 1972, 94(2): 483.