

软件体系结构及其风格的现状与发展^①

The Situation and Development of the Software Architecture and Its Style

汪保杰 王如龙 (湖南大学 软件学院 湖南 长沙 410082)

摘要: 软件体系结构是软件工程领域中一个重要的研究课题,是大型软件开发中的关键技术。经过十多年的发展,软件体系结构也超出了传统的对软件设计阶段的支持,逐渐扩展到整个软件生命周期。通过对现有软件体系结构和风格的归纳总结,描述了软件体系结构的主要发展方向及其应用的重要意义。

关键字: 软件体系结构 软件体系结构风格 基于软件体系结构的软件开发 综述

1 引言

随着软件规模和复杂度不断地扩大和增加,软件开发的成败在很大程度上取决于软件体系结构的设计。将那些公认的、被多次成功使用的软件体系结构称为软件体系结构风格,比如 REST(Representational State Transfer)风格、多级正交风格、SOA(Service-Oriented Architecture)风格、RIA(Rich Internet Application)风格等等。软件体系结构可以看成是软件体系结构风格的实例,软件体系结构风格是描述某一特定应用领域中系统组织方式的惯用设计模式,是一组约束的集合。

面向对象,面向组件,面向服务已是当今主流的软件开发思想,同时以演化和增量为基础的迭代开发模式也是主流的开发过程。尽管如此,如果软件体系结构设计不当,就可能引起系统重构,甚至整个体系结构重新设计,大量实践也证明了:一个成功的软件系统往往都有一个好的软件体系结构。

2 软件体系结构定义

软件体系结构概念远在 20 世纪 80 年代就已提出,到 90 年代引起人们的重视,在学术界成为研究的热点并在软件开发中得到一定的应用。许多研究人员从不同角度和不同方面对软件体系结构进行了刻画,较为典型的定义有:

(1) 1992 年, Dewayne Perry 和 Alex Wolf 的定义^[1]: 软件体系结构={元素, 形式, 准则}。

SA= {elements, form, rational}

(2) 1993 年, Mary Shaw 和 David Garlan 的定义^[2]: 软件体系结构={组件, 连接件, 约束}。

SA= {components, connectors, constrains}

(3) 1994 年, Bass 等人提出^[3]: 软件体系结构设计至少应包括应用领域的功能分割、系统结构、结构的领域功能分配三个方面。

(4) 1995 年, David Garlan 和 Mary Shaw 修正定义^[2]: 软件体系结构包括系统组件的结构、组件的相互关系以及控制组件设计演化的原则和指导三个方面。

(5) 1996 年, Kruchten 的“4+1”模型^[4]认为,软件体系结构由逻辑视图、物理视图、过程视图、开发视图以及场景视图组成。

(6) 1997 年, Bass, Clements 和 Kazman 提出^[3]: 软件体系结构包括组件、组件的外部可见性以及相互的关系。

比较上述定义不难发现他们都涵盖了如下实体: 构件、连接件、配置以及设计约束。总之,软件体系结构是在大中型软件开发中,必须从一个较高的层次来考虑组成系统的结构以及设计规则和约束。而且,软件体系结构应能体现开发中具有重要影响的设计决策(关键需求决定体系结构设计),便于各种人员的交流,反映多种关注。

3 软件体系结构风格综述

通过对软件体系结构风格的研究和实践促进了对

^① 收稿时间:2009-02-21

设计和过程的重用,经过实践证实的解决方案可以可靠地用于解决新的问题。基于软件体系结构风格构建的系统可使别的设计者很容易地理解系统的体系结构。例如:如果某人把系统描述为 B/S 模式,则不必给出设计细节,我们立刻就会明白系统是如何组织和工作的。

随着软件研发技术的不断进步,近些年接连总结出了几种新型的软件体系结构风格,如:正交软件体系结构风格、REST 软件体系结构风格、面向服务的 SOA 软件体系结构风格、基于 WEB 的插件体系结构风格、RIA 富客户端风格等等,关于软件体系结构风格的研究还处于起步阶段。

3.1 正交风格

正交软件体系结构^[5]是一种以垂直线索为基础的层次化结构,他由组织层(Layer)和线索(Thread)构成,组织层是由具有相同抽象级别的构件组成,线索是相对子系统的特例,他是由一组分布在不同层并有调用关系的构件组成。

正交软件体系结构其实就是将软件功能正交分解,按功能的正交相关性,垂直分割为若干个线索,线索又分为几个层次。其主要特征可归纳为:

- (1) 由完成不同功能的 $n(n>1)$ 个线索(子系统)组成。
- (2) 系统具有 $m(m>1)$ 个不同抽象级别的层。
- (3) 线索之间是相互独立的(正交的)。
- (4) 系统有一个公共顶层(驱动层)和公共底层(公共构件和数据层)。

对于大型的且比较复杂的软件系统,线索还可以划分为更低一级的线索,形成多级正交软件体系结构。

3.2 REST 风格

REST 风格^[6]是由 Roy Thomas Fielding 博士在 2000 年首次提出的,是针对 WEB 应用而设计的,其目的是为了降低开发的复杂性,提高系统的可伸缩性。REST 提出了如下设计准则:

- (1) 网络上的所有事物都被抽象为资源(resource)
- (2) 每个资源对应一个唯一的资源标识符(resource identifier)
- (3) 通过通用的连接器接口(generic connector interface)对资源进行操作
- (4) 对资源的各种操作不会改变资源标识符
- (5) 所有的操作都是无状态的(stateless)

(6) 强调中间媒介的作用(如代理服务器、缓存服务器和网关)

3.3 SOA 风格

SOA 是一种方法论,一种软件体系结构风格^[5],一种分布式处理的软件体系结构,它要解决的问题领域跟 CORBA、DCOM、EJB 等早期的分布式技术式一样,但 SOA 在松散耦合、语言无关、平台无关等方面要占很大优势,并且 SOA 提出的这种体系结构最大限度的包容了以往的技术,使已有的 IT 投资得到了保护和重用。SOA 应用可以采用 Web Service、CORBA、DCOM、RMI 等中的任何一种来实现。不过,Web Service 技术对 SOA 支持的最好,也是被广泛应用于 SOA 的实现技术。

3.4 RIA 风格

随着应用软件业务逻辑越来越复杂,原先作为主流显示技术的 HTML/JAVASCRIPT/CSS 愈发捉襟见肘。为提升用户体验,出现了一种新类型的 Internet 应用程序,就是 RIA。RIA 结合了桌面应用程序的反应快、交互性强的优点与 Web 应用程序的零部署、易升级、操作简便的特性,简化并改进了 Web 应用程序的用户交互,增强了用户体验。

RIA 风格的特征可归纳为:

- (1) 强交互性: RIA 支持丰富的 UI 组件
- (2) 直接管理: 局部的数据更新,通过客户端计算可直接实现对用户请求的响应
- (3) 多步骤处理: 所有内容在一个界面中,添加转换效果,使应用程序的状态在各步骤中轻松移动
- (4) 文本独立性: RIA 集成 XML 特性,简化异质系统的通信,方便数据的存取
- (5) 平台无关性: 应用层次对所有 RIA 客户端都是一致的

4 软件体系结构的发展历史与现状

4.1 软件体系结构的发展历史

20 世纪 70 年代以前,系统规模和复杂度较小,很少明确考虑系统结构。70 年代中后期,出现了概要设计与详细设计,此时软件体系结构已作为一个明确的概念出现在系统开发中。

20 世纪 80 年代初到 90 年代中期,是面向对象开发方法兴起与成熟阶段。90 年代以后强调软件开发采用构件化技术和体系结构技术,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。

1995 年出版的 IEEE Software 体系结构专刊^[7]和 1996 年出版的专著《Software Architecture: Perspectives on an Emerging Discipline》^[2],可以认为是软件体系结构作为软件工程一个研究方向正式提出的标志。此后十年内,软件体系结构领域得到了蓬勃发展,获得工业界的广泛关注与认同,如 UML2 标准中引入了软件体系结构领域中连接子的概念^[8],在实际软件开发过程(如 RUP 统一软件开发过程)中也引入软件体系结构的概念和原则。2006 年出版的 IEEE Software 软件体系结构专刊^[4]总结了这十年间的软件体系结构研究与实践。

目前体系结构研究可以分为典型的两种派别:学院派与实用派。学院派研究者侧重于软件体系结构形式化理论研究。实用派研究者将软件体系结构设计、描述与表示同传统的软件系统建模视为一体,将软件建模技术直接用来描述软件体系结构。

4.2 软件体系结构的发展现状

4.2.1 软件体系结构的形式化描述语言 ADL

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进展。目前主要的 ADL 包括 Unicon, Rapide, Darwin, Wright, C2 SADL, Acme, axel, XYZ/ADL, ABC/ADL 等。随着软件体系结构研究的不断深入,研究人员还挖掘出了 ADL 需要进一步深入考虑的问题,比如支持动态、分布、移动系统的建模,针对不同应用领域的建模等,从而提出了一些新型的 ADL,比如基于可扩展构件模型 Fractal 的 ADL、从结构和行为的角度来描述体系结构的 π -ADL、基于多 Agent 系统的 Skwyrl-ADL 等。软件体系结构的形式化研究为软件开发和维护提供了强有力的支持。

目前,存在众多的体系结构描述语言,这样势必导致认知和使用上的混乱。学者们认为有必要从元模型层次探讨统一的体系结构描述方式。随着软件系统的不断发展和演化,新的问题也不断涌现,对软件体系结构的描述也将是研究的热点。

4.2.2 软件体系结构的多视图模型表示

研究软件体系结构的首要问题是如何表示软件体系结构,即如何对软件体系结构建模。根据建模的侧重点的不同,可以将软件体系结构的模型分为 5 种:结构模型、框架模型、动态模型、过程模型和功能模型。

这 5 种模型各有所长,也许将 5 种模型有机地统一在一起,形成一个完整的模型来刻画软件体系结构更合适。例如, Kruchten 在 1995 年提出了一个“4+1”的视角模型(逻辑视角、过程视角、物理视角、开发视角和场景视角); Homelie 的 4 视图模型(概念视图、模块视图、执行视图、代码视图); CMU-SEI 的 Views and Beyond 模型(模块视图、构件和连接子视图、分配视图)等。此外,工业界也提出了若干多视图描述软件体系结构模型的标准,如 IEEE 标准 1471-2000(软件密集型系统体系结构描述推荐实践)、开放分布式处理参考模型(RM-ODP)、统一建模语言 UML 以及 IBM 公司推出的 Zachman 框架等。对软件体系结构的模型描述将一直是研究的热点。

4.2.3 软件体系结构的分析和设计研究

软件体系结构是对软件系统的高层抽象,是在软件开发过程之初产生的,因此设计优质的体系结构可以减少和避免软件错误的产生和维护阶段的高昂代价。生成一个满足软件需求的体系结构的过程即为体系结构设计,体系结构设计过程的本质在于:将系统分解成相应的组成成分(如构件、连接件),并将这些成分重新组装成一个系统。体系结构设计有两大类方法:过程驱动方法和检查列表驱动方法。要将体系结构运用到软件开发过程中,分析和设计是第一步,因此,软件体系结构的分析和设计在企业应用开发中得到了相当大的重视。

4.2.4 基于软件体系结构的软件开发过程研究

软件开发过程模型是跨越整个软件生存周期的系统开发、运行、维护所实施的全部工作和任务的结构框架,给出了软件开发活动各阶段之间的关系。

目前,常见的软件开发过程模型大致可分为三种类型:

(1)以软件需求完全确定为前提的瀑布模型。

(2)在软件开发初始阶段只能提供基本需求时采用的渐进式开发模型,如螺旋模型,敏捷开发过程等。

(3)以形式化开发方法为基础的变换模型。

所有开发方法都是要解决需求与实现之间的差距。但是,这三种类型的软件开发模型都存在这样或那样的缺陷,不能很好地支持基于软件体系结构的开发过程。因此,研究人员在发展基于体系结构的软件开发模型方面做了一定的工作,比如 IBM Rational 的 RUP 统一开发过程等。关于软件体系结构对软件开发

过程支持的研究还处于探索阶段。

4.2.5 特定领域的软件体系结构 DSSA

鉴于特定领域的应用具有相似的特征,因而经过严格设计,可以有效地实现复用,并可借鉴领域中已经成熟的体系结构。特定领域的体系结构是将体系结构理论应用到具体领域的过程。常见的有电信软件的体系结构研究、CASE 体系结构、CAD 软件的参考模型、测试环境的体系结构、信息系统的参考体系结构、网络体系结构、机场信息系统的体系结构等。

4.2.6 软件产品线体系结构的研究

软件体系结构的开发是大型软件系统开发的关键环节。体系结构在软件生产线的开发中具有至关重要的作用,在这种开发生产中,基于同一个软件体系结构,可以创建具有不同功能的多个系统。在软件产品族之间共享体系结构和一组可重用的构件,可以增加软件重用、降低开发和维护成本。

一个产品线代表着一组具有公共的系统需求集的软件系统,它们都是根据基本的用户需求对标准的产品线构架进行定制,将可重用构件与系统独有的部分集成而得到的。

4.3 软件体系结构的主要研究方向

归纳现有软件体系结构的研究方向,主要包括如下几个方面:

(1) 软件体系结构理论模型的研究。比如:多视图表示

(2) 软件体系结构描述研究。主要研究体系结构描述语言及其支持环境、体系结构描述规范。

(3) 软件体系结构设计研究。包括体系结构设计方法、体系结构风格与模式、体系结构设计空间等内容,针对软件体系结构设计目前还没有较为详细的综述文献,对风格的研究也处于起步阶段。

(4) 软件体系结构分析与验证方法。研究如何将软件的非功能特性转化为体系结构的需求,如何分析体系结构满足期望的需求;统一的,自动或半自动分析方法的研究;对体系结构的语法、语义、类型失配等进行检查与验证的研究。

(5) 软件体系结构演化与复用研究。研究产品线中软件体系结构演进的理论与方法,从已有文档、系统设计和代码中逆向提取软件体系结构;体系结构复用与构件组装;领域的软件体系结构研究;软件产品线技术等。

(6) 动态软件体系结构研究。研究软件系统由于特殊需要必须在连续运营情况下的体系结构变化与支撑平台。

(7) 基于软件体系结构的软件开发。研究引入体系结构后的软件开发过程(如:项目组织管理、配置管理等);基于体系结构开发与中间技术集成;基于体系结构的程序框架自动生成技术;如何针对实践的需要选择适合的风格;软件体系结构对网构软件的支持等

(8) 基于软件体系结构的部署研究。部署方案评估以及生成最优部署方案等等。

5 结语

软件体系结构的好坏是决定软件系统成败的主要因素,软件体系结构已经成为国内外软件研究的热点,经过十多年的发展,软件体系结构也超出了传统的对软件设计阶段的支持,逐渐扩展到整个软件生命周期。本文通过对现有软件体系结构和风格的归纳总结,描述了软件体系结构的主要发展方向及其应用的重要意义。

参考文献

- 1 Perry D, Wolf AL. Foundations for the study of software architecture. ACM Sigsoft Software Engineering Notes, 1992,17(4):40 - 52.
- 2 Shaw M, Garlan D. Software Architecture-perspectives on an emerging discipline. 北京:清华大学出版社(影印版), 1998.
- 3 Bass L, Clements P, Kazman R. Software Architecture in Practice. New York: Addison Wesley Publishing, 1998.
- 4 Kruchten P, Obbink H, Stafford J. Special issue on software architecture. IEEE Software, 2006,23(2):22 - 30.
- 5 王一宾,张玉州,程一飞.几种新型软件体系结构风格的分析.计算机技术与发展, 2008, 18(8):39 - 42.
- 6 Fielding RT. Architectural Styles and the Design of Network-based Software Architectures, 2000.
- 7 Garlan D, Perry D. Special issue on software architecture. IEEE Software, 1995,21(4):269 - 274.
- 8 Garlan D. A ten-year perspective on "Formalizing Architectural Connection". Most Influential Paper of the ICSE'94 Award Presentation, Given at ICSE 2004.