

# 软件体系结构模型的分析及研究

周 枫

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650051)

**摘要:** 软件体系结构技术的研究, 可以提高软件系统的开放性、集成性和复用性. 本文通过分析几种软件体系结构的系统模型, 从技术及应用二个层面上探讨它们的技术特性, 提出了一种基于复用部件的软件体系结构的模型框架. 该模型综合了几种主要的软件体系结构模型的技术与工程特性, 它在软件项目的工程化方面具有指导性作用.

**关键词:** 软件体系结构; 部件; 中间件; 总线

**中图分类号:** TP311.52    **文献标识码:** A    **文章编号:** 1007-855X(2002)06-88-05

## 0 引言

软件工程是指导软件开发过程、提供认知和表达软件系统思想和方法的一门学科. 在软件工程技术的发展中, 软件体系的体系结构一直是管理者、开发者和用户十分关注的问题. 究其原因主要有三: 一是软件系统的规模越来越大, 无论是开发还是运行、维护, 软件系统都已变得非常困难. 如果能在软件体系结构上有所改进, 就能从技术上大大降低系统的复杂性. 二是软件系统中实现的功能块之间的耦合度、冗余度太高, 难于单独升级和重复使用. 软件体系结构的研究可以从技术上降低功能块之间的耦合度和冗余度, 提高功能块之间的独立性和复用性, 为功能块结构框架的复用创造条件. 三是软件系统的兼容性、集成性太差, 不同应用平台上运行的软件难于相互融合. 如果能从软件体系结构入手, 通过部件、中间件等技术过渡, 就可大大提高软件系统的跨平台运行能力和集成能力.

开放性、集成性和复用性是评价软件系统体系结构性能优良的三个重要指标.

开放性主要是指软件系统中数据的开放性、功能的开放性和系统的可扩充性. 一个系统的开放性是否良好主要取决于该系统的体系结构模型.

集成性指的是通过一致的信息描述手段和处理机制, 能够将若干功能不同的子系统集成在一个执行环境中的能力. 一个系统集成性的好坏也主要取决于体系的体系结构模型.

一个开放性好的应用系统, 应该容易实现集成, 而集成性的高低又取决于系统部件的复用性; 集成性和复用性的改进、提高又能促进系统的开放性.

本文从软件体系结构的概念入手, 通过分析几种软件体系结构的系统模型, 从技术及应用二个层面上探讨它们的技术特性, 提出一种基于可复用部件的分层软件体系结构的模型框架. 该模型综合了几种主要的软件体系结构模型的技术与工程特性. 这项技术的研究对于寻求一种既开放、又通用的软件体系结构以支持即插即用的软件开发方法具有重要的指导意义.

## 1 软件体系结构的概念

一个软件系统是由若干个组成部分, 按照一定的组织结构形式构成的. 这个组织结构形式从概念上就是一个总体性框架, 它表达了软件系统各个组成部分之间的关联关系以及控制系统设计和进化的一组原则.

软件体系结构技术中研究的首要问题是: 部件的规格定义和体系结构的建模.

收稿日期: 2002-04-25.

第一作者简介: 周枫(1958~), 男, 副教授, 主要研究方向: 软件工程.

## 1.1 部件

部件(Components)是软件系统中具有独立功能的部分,也是软件体系结构中重要的组成要素,它在功能和数据上构成了一个软件系统的基础。

从技术上说,软件部件是一种定义良好、功能独立、可以重复应用的二进制代码集,它可以是一个功能服务块、一个经过封装的对象组、甚至可以是一个系统框架或软件应用模型。

在不同的软件开发时期,软件部件概念的内涵是不同的。例如在20世纪60~80年代初期,软件工程方法是以结构化分析与设计方法(SADT)为主导,因此部件是以功能模块的形式构造的。模块的内聚性、耦合性是评价模块划分质量的主要标准。20世纪80年代以后,面向对象的软件开发技术的迅速发展,软件部件则以类或对象的形式构造。类是功能和数据的聚合,它具有封装性、多态性和继承性。类比起模块来说内聚度更高、独立性更强且具备了复用的基础。但类的实现依赖于编程语言环境,且对象仅存在于一个程序中,程序之外无法感知和访问对象。其复用性还难于满足软件工程的实际需要。20世纪90年代以来,由于技术与应用环境的不断进步,基于对象的部件其内涵有了较大的扩充和改进,主要表现在:部件的构造粒度、统一的二进制标准、实现位置透明地访问、外部特性允许不唯一等方面。

进入20世纪90年代以来,计算机应用逐渐步入以网络为应用平台的阶段。基于网络而建立的各类应用系统不仅实现了数据共享,并能支持各类计算资源的共享。由此提出并发展了分布式的客户/服务器的应用模式。

在分布式网络环境下发展的分布对象技术是计算机应用的下一个主流方向,其主要技术目标是:寻找解决异构网络环境下应用的互操作性问题。为了解决大型应用系统的集成性和扩展性之间存在的矛盾,可计算部件的作用日益突出,并对部件的技术规范及应用层次进行了深入的研究。

中间件(Middleware)是一种独立于应用系统的特殊部件。在功能上它是介于体系结构中应用层和操作系统之间的一个通用服务层次,而在技术上中间件的作用是隔离应用系统与运行环境的关联关系,即:屏蔽网络硬件平台的差异性和操作系统与网络通信协议的异构性,而使传统的客户/服务器应用模式扩展为客户/中间件/服务器的三层或多层模式。从技术上支持开发可靠的、可扩展的、复杂的事务密集型应用系统。

目前许多行业组织和国际性大公司都在研究软件部件的规范标准,希望能在今后的市场竞争中获取主动。最有代表性的软件部件的规范标准主要有:Microsoft公司推出的OLE/COM、OMG协会的OMA/CORBA以及Sunsoft公司的JavaBeans。

## 1.2 软件体系结构的建模

软件体系结构研究的是如何表达软件系统的结构连接关系,用更专业的术语来描述的话,软件体系结构要定义构成该系统的所有计算部件和部件之间交互的连接件,以及如何将部件和连接件结合在一起的规程。

软件系统建模技术的提出,主要目的是为了解决随着技术的发展,软件应用系统的规模日益扩大,结构模型变得日益复杂而导致的系统设计方面的问题。从早期提出的以数据为中心的体系结构模型、以功能执行为中心的体系结构模型,逐步发展到今天以系统内核为中心的面向对象的体系结构模型和基于总线结构的体系结构模型。这些体系结构模型的演变进化是为了不断适应计算机应用领域的扩大而引出的新的需求。软件系统建模技术的共同目标就是要提高实际应用系统的开放性、集成性、适应需求变化的扩展性以及开发与执行的高效性。

一个软件应用系统的设计过程中,计算部件及其中间件的规格定义以及这些部件的结构体系是决定系统开发质量的重要因素。研究软件体系结构方面的问题,对于提高应用系统的集成性、复用性以及质量和效率有着重要的指导意义。

## 2 软件体系结构模型的分析

软件体系结构表达的是应用系统中各种功能部件之间的一种结构关系,在目前的实际应用中主要提

出了四种基本的软件体系结构模型, 它们代表了软件工程与产业发展的不同阶段.

图 1 表达的是以数据为中心的应用体系结构的模型. 在这个模型中, 数据集合被置于整个软件系统的核心位置, 并以统一的数据描述形式提供各个功能部件共享. 各个功能部件的开发过程完全独立, 部件之间存在一致的数据交换的接口.

这种结构模型的优点是整体扩充性好, 可灵活添加符合数据交换标准的应用程序或系统程序, 能够提供数据的复用. 技术上的不足是系统结构中功能上的联系比较松散, 无法实现功能复用, 容易造成代码冗余. 另外数据接口标准难于符合所有的应用需求, 这使系统带有局限性. 从系统的开放性来说, 这种结构模型只具有数据的开放性, 不具有功能的开放性.

图 2 所示的是以功能执行为中心的应用体系结构的模型. 该模型使用一个统一的执行中心的形式来实现各类应用或系统部件的数据和界面的共享. 其最大的特点是实现了两个分离: 一是共有的计算和执行功能(如: 对数据库的存取操作和其它系统的接口等)和应用程序的分离; 另一个是用户与系统的界面交互功能和应用程序的分离. 这两个分离优化了代码执行效率, 避免了代码冗余, 便于实现风格一致的用户界面. 系统与数据库的交换功能集中到执行中心统一实现, 有利于数据的管理与维护, 保证了数据的一致性. 这类模型既可实现数据的开放性, 又具备了功能的开放性, 系统的扩充性也较好.

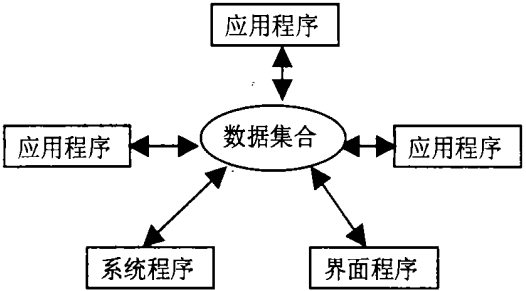


图 1 以数据为中心的应用体系结构模型

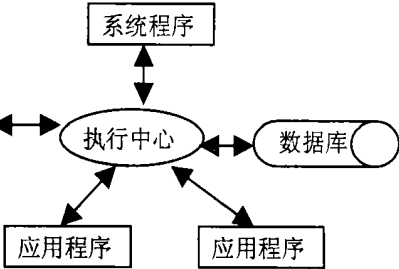


图 2 以执行为中心的应用体系结构模型

但这种结构模型的不足之处在于: 一个能够符合各种应用需求的执行中心(即: 一组公共的功能集合), 在实际中是难于确切定义的. 另外系统的功能和执行效率过分依赖于执行中心, 而执行中心既要与用户界面和所有应用程序通信又要管理和维护数据库, 其功能结构的设计比较复杂, 容易造成系统运行的不稳定, 从而降低效率.

20 世纪 90 年代以来, 面向对象技术的发展提出了图 3 所示的面向对象的应用体系结构模型. 这类模型主要由系统内核对象、界面对象、应用对象按照约定的通信协议, 通过相互之间传递的消息协同配合实现所要求的功能. 系统内核中的对象封装了能为界面对象和所有应用对象所共享的数据和相应的操作, 界面对象中封装了与用户交互界面有关的数据及操作, 应用对象中封装了与应用相关的数据和操作.

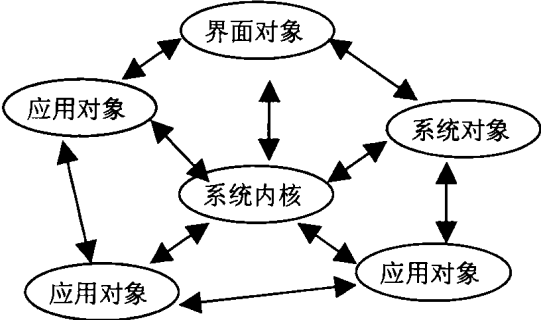


图 3 面向对象的体系结构模型

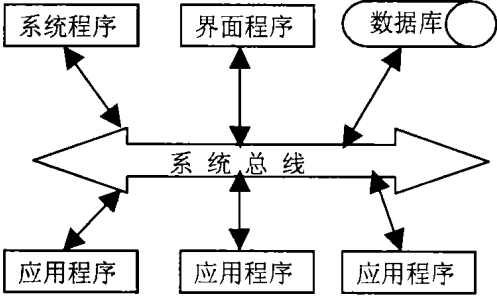


图 4 基于总线的体系结构模型

这类模型的好处是: 系统由各类对象实体组成, 各对象实体地位平等. 模型属于一种无中心的体系结构. 这使得系统的开放性更好, 集成性更高, 扩充更灵活. 另外数据和功能的封装既提高了系统的安全性, 也为对象的复用提供了技术支持, 同时还降低了由于数据和功能的集中管理而带来的通信开销和操作上

的复杂性。

该类模型技术上的不足在于随着对象数目的增加, 对象与对象之间的消息通信链路将以平方级数急剧增加, 结果势必加大系统的通信开销. 另外每个对象实体中还需要构造一个包含若干个逻辑相关对象的服务信息库(对象之间的通信协议), 这类信息不但重复, 而且还要求保证信息的一致性, 这又增加了信息冗余. 再者对象之间的接口缺乏统一标准, 导致对象实体的扩充不规范, 不利于系统的维护和对象的复用。

图 4 表达的是一种基于总线的应用体系结构模型. 在这种模型中, 对象是构成部件的基本元素. 由于面向对象技术的发展以及中间件概念的提出, 部件中的对象采用统一的规格精心定义, 并按照开放式规格标准构造部件. 系统以总线结构的形式连接各个部件. 这个总线实际上是一个实现部件之间连接的公共接口规格, 部件可以即插即用, 实现无缝集成. 这类模型的好处是: 部件之间的通信链路数为线性速率增加, 各个部件之间的接口规范一致, 导致部件之间通信的复杂度大大降低并提高了部件的互操作性. 不足之处在于一个统一的公共接口规格难于满足各类问题应用域的不同需求. 虽然某些公司和组织现已提出若干通用的公共应用接口规范, 如: JavaBeans、OMG/CORBA、OLE/COM, 但其内容也多是原则性和框架类的, 操作性不强。

3 分层体系结构的框架模型

以上分析并论证了几种软件体系结构模型的技术特性. 从目前软件应用技术的发展方向看, 软件系统的设计目标是: 开发开放性更好, 集成性更灵活的应用体系结构. 本文从目前计算机应用技术的发展动态及应用技术的需要, 提出一种基于可复用部件的分层软件体系结构的模型框架, 这种结构模型在软件的工程化方面具有指导意义, 其开放性更好, 系统的集成性更高. 模型如图 5 所示:

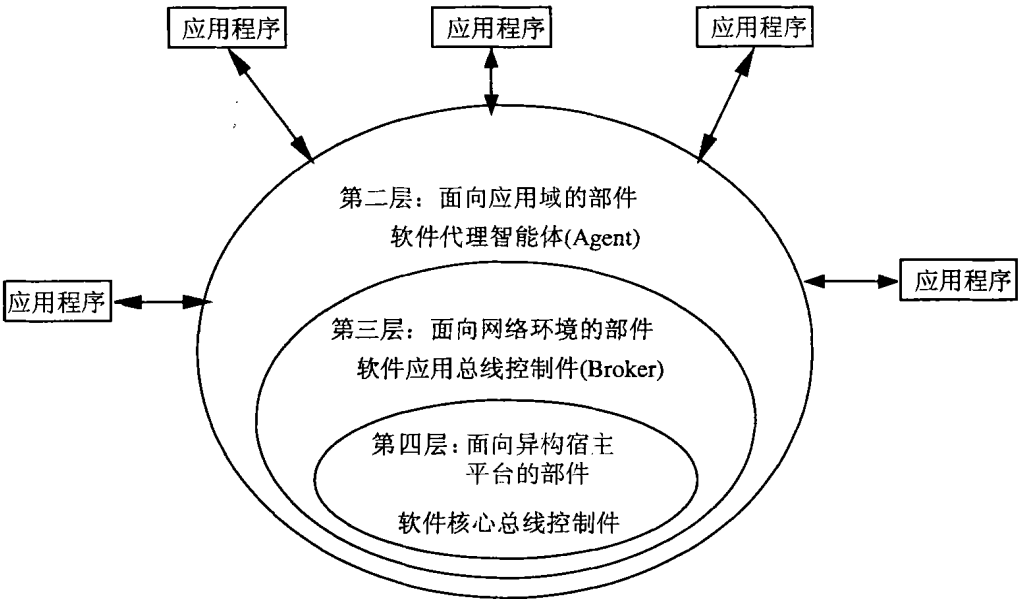


图 5 基于复用部件的分层应用体系结构模型的概念框架

图 5 中第一层为用户应用程序层, 主要由面向用户实际需求的需求而构造的各类应用程序组成. 该层是一个按照统一的标准规范编制的、实现用户所需功能的面向对象应用程序的集合. 第二层为面向不同的问题领域, 适用于各类应用需求而构造的接口部件集合. 这类部件的主要作用是代理实现应用程序的对象之间的通信过程, 即: 将应用系统的接口功能从应用程序中分离出来, 以软件代理智能体( Agent) 的形式提供公共通信接口, Agent 具有较高的自主性, 能够对周围环境有一定的感知性, 并且可以根据需要在网络节点间移动. 设置该层的目的是使应用程序更关注于用户功能实现。

第三层为面向不同网络环境的应用部件的集合, 其作用主要是为上一层面向不同应用领域的接口部件提供一个网络环境下通用的公共接口的标准规范. 用于屏蔽网络中本地对象和远程对象的位置属性, 使

得与之进行通信的其它对象来说, 它们都是等价的.

第四层由面向异构宿主平台的核心应用部件构成. 核心应用部件的作用主要是实现网络运行在不同宿主平台之间的数据交换和功能的互用, 它屏蔽了异构宿主平台之间的差异, 将适用于不同应用需求的部件内部的异构数据转换为同构数据, 并向上一层的应用总线提供统一的面向平台的服务.

基于可复用部件的分层体系结构模型的概念框架, 综合了以上介绍的几种软件体系结构模型的优点, 使得开放性更好, 集成性更高. 更重要的是在这个概念框架中, 面向用户需求的应用程序的结构更简单, 功能更单一, 代码更容易编制实现. 而涉及到通信、接口、环境、平台的那一部分功能已被充分地对象化, 并以符合标准规范的软件部件的形式提供, 最大限度地支持了系统功能的复用. 这不仅提高了应用系统的开发效率和执行效率, 还在技术上保证了应用系统的质量.

该模型从技术上支持即插即用的软件开发方法, 即插即用的设计思想是受到硬件制造过程的启发而用相同的思路来处理大型软件的研发工作. 由专门的机构和人员按统一的标准规范生产软件部件, 而另一部分人则构造整个软件的结构, 并且把软件部件插入该结构以便迅速完成大型软件的开发工作.

#### 参考文献:

- [ 1 ] Mary Show, David Garlan, Software architecture. perspectives on an emerging discipline[ M ] . Prentice Hall, 1996.
- [ 2 ] Edward Yourdon, Object—Oriented System Design. An Integrated Approach[ M ] . Yourdon Press/Prentice Hall, 1994.
- [ 3 ] 周枫, 刘晓燕等. 软件工程[ M ] . 重庆: 重庆大学出版社, 2001.
- [ 4 ] 汪成为, 郑小军等. 面向对象分析设计及应用[ M ] . 北京: 国防工业出版社, 1992.
- [ 5 ] [ 美 ] R. Otte, P. Patrick, M. Roy 著. CORBA 教程: 公共对象请求代理体系结构[ M ] . 北京: 清华大学出版社, 2000.

## Analytical Research on the Software Architecture Model

ZHOU Feng

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650051, China)

**Abstract:** The research on software architecture technology may enhance the opening, integrating and reusing-ability of the software systems. This paper analyses some system models of software architecture, and makes an inquiry into their features from technology and application angles. A more—layer model of software architecture on reusing components is put forward which synthesizes some main technical and engineering features of software architecture, which has a directive function for software engineering.

**Key words:** software architecture; component; middleware; bus