

# 面向构件的门诊部信息管理系统软件架构

丁晨路<sup>1</sup>, 覃勇<sup>2</sup>, 王旭辉<sup>2</sup>

(1. 重庆通信学院 研究生管理大队, 重庆 400035;

2. 重庆通信学院 机动作战通信系指挥自动化教研室, 重庆 400035)

**摘要:** 构件化软件的设计集中于软件体系结构设计, 其最主要的特征在于目标系统由大量可复用构件组成, 这就要求在设计阶段应当设计出满足需求的软件体系结构。以一个门诊部信息管理系统软件架构为例, 介绍了面向构件的软件架构方法, 并对系统架构选型进行分析, 提出了本系统的软件架构, 为基于构件技术的系统实现提供良好的依据。

**关键词:** 构件; 软件架构; 三层体系结构

中图分类号: TP311.1

文献标识码: A

文章编号: 1674-7720(2011)14-0004-03

## Component-oriented software architecture of clinic information management system

Ding Chenlu<sup>1</sup>, Qin Yong<sup>2</sup>, Wang Xuhui<sup>2</sup>

(1. Department of Graduate Management, Chongqing Communication Institute, Chongqing 400035, China;

2. Department of Command Automation, Chongqing Communication Institute, Chongqing 400035, China)

**Abstract:** Component-based software design focus on software architecture design, its main feature is that the target system by the composition of a large number of reusable components, which requires the design phase should be designed to meet the needs of the software architecture. In this paper, use a clinic information management system software architecture for example, firstly introduces the software architecture based on component technology methods, and then analyzes the system architecture selection. At last, provides a good basis for making the component-based system software architecture for system implementation.

**Key words:** component; software architecture; three-tier architecture

自从构件技术提出以来, 如何有效地划分构件、进行最大化的软件复用成为研究的热点。构件化软件的最主要特征<sup>[1]</sup>在于目标系统由大量可复用构件组成, 从系统化构件复用过程来看, 传统软件开发方法中的详细设计被划分到了构件制作阶段, 与基于构件开发目标系统属于相对独立的阶段。构件化软件的设计是传统软件开发方法中的总体设计或高层设计, 要求设计阶段就应设计出满足需求的软件体系结构。

### 1 面向构件的软件架构

软件架构<sup>[2]</sup>是指某个软件或计算机系统的软件架构, 即组成该系统的一个或多个结构, 构成软件的各个部分, 形成该构件的外部可见属性及相互间的关系。基于构件的软件架构, 更加强调构件的概念, 架构时考虑两

个方面: (1)与软件基础设施有关, (2)与构件在基础设施上的操作有关。基于这两方面的考虑, 通常在架构时考虑技术架构和应用架构两个方面。

#### 1.1 技术架构

技术架构<sup>[3]</sup>定义了技术环境和软件基础设施, 是支持组织以实现运作(非功能)需求(尤其是组织的应用程序和信息系统结构)的硬件和软件基础设施的体系结构。它描述了所使用的技术的结构和内部关系, 以及这些技术如何支持组织的运作需求。

技术架构包括技术环境和技术基础设施两个部分。技术环境解决不同软硬件平台之间的差异性。技术基础设施包括在应用中可以被标准化的部分。标准化使得通用代码从面向业务的构件中移出, 放到一个独立的

## 软件天地 Software Technology

可复用的软件基础设施中。软件基础设施可以通过预先构建的构件框架的方式提供,它处理构件行为中已经标准化的方面。

### 1.2 应用架构

应用架构<sup>[3]</sup>包括了构件开发者和组装者面临的架构问题。是自服务的体系结构,用于支持和实现这样的业务需求,包括该业务与其他应用程序之间的接口。它描述了应用程序的结构以及该结构如何实现组织的功能需求。图1所示为面向构件的软件构架模型<sup>[4]</sup>。

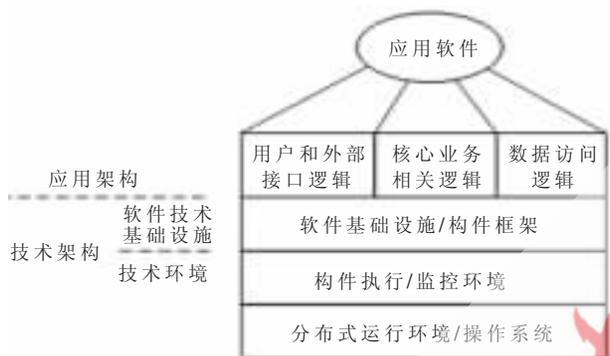


图1 基于构件的软件架构模型

## 2 典型系统架构分析

以一个门诊部信息管理系统为例,对基于构件技术的系统架构进行分析。

### 2.1 架构选型

根据需求分析过程中得到的用例模型,进行了如下选型过程<sup>[5]</sup>:

#### (1) 选择套装软件还是自行开发

先考虑了市场上套装软件的产品,希望借助成熟的产品加快实施进度,确保软件的质量和稳定性。但是通过考察,发现套装软件存在以下问题:

①从功能角度分析:市上现有的系统虽然提供了丰富且强大的功能,但与本系统的需求交集只占了大约60%,此部分之外的需求需要花费很大的代价定制,而软件中约有50%的特性是本系统所不需要的。

②从实施周期角度分析:由于客户化的内容比较多,包括前期的需求映射和套装软件的模型调整,要完成实施,经过评估需要2年左右的时间,超出了用户的预期。

③从成本角度分析:套装软件相对比较昂贵,另外需要购买相应的咨询、实施服务,这样使得成本更大。

④从扩展性角度分析:套装软件集成了一套强大的模型,通过模型的配置,基本上可以满足业务上的变化,但是在条件不能满足的情况下,由于系统实现非常复杂,采用的技术手段包含了很多自由的标准,难于进行二次开发,不利于移植和维护。

基于以上的分析,选择了自行开发软件。

#### (2) 技术平台的选择

所谓“平台”就是能够独立运行并自主存在,为其所支撑的上层系统和应用提供运行所依赖的环境。

技术平台是一套完整的、严密的服务于研制应用的软件产品及相关文件。真正的技术平台应该是选择合适的技术体系(如J2EE、.NET等),技术架构(一组设计模式如MVC的集合),充分发挥技术体系及技术架构的优势,能够大大提高应用软件开发速度,指导并规范应用软件开发、设计、编码、测试、部署各阶段工作,提炼用户真正需求,提高代码正确性、可读性、可维护性、可扩展性、伸缩性等软件工具。优秀的技术平台还包括一套高效的底层通用的代码,甚至还包括代码生成器、代码安全漏洞检查工具等。

### 2.2 三层软件架构

在总结基于构件的软件架构方法和进行架构选型分析后,在本系统中引入三层结构的设计思想,采用构件化的开发方法,将系统业务逻辑封装在应用服务器上,客户端应用程序自动从应用服务器上下载和更新业务逻辑,达到“应用逻辑与程序代码分离,数据与应用分离,应用与流程分离”,最大化地支持业务的灵活开展。图2所示为系统软件架构。

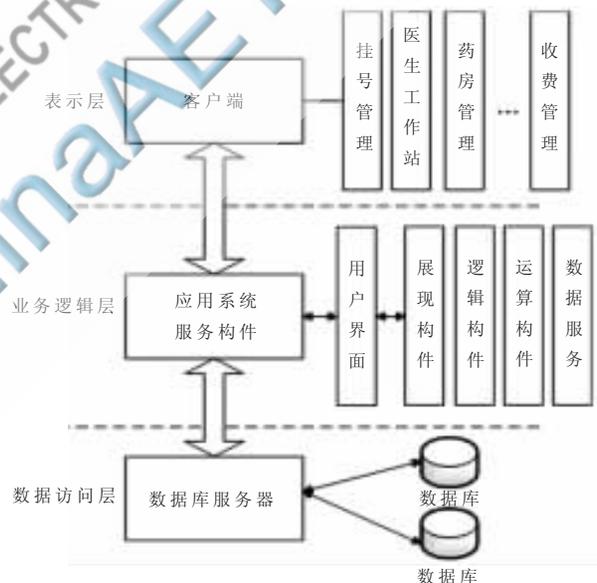


图2 门诊部系统软件架构

系统采用三层体系结构设计。所谓三层体系结构<sup>[6]</sup>,即用户层、应用层和数据库服务器。用户层主要指用户界面,要求尽可能地简单,使最终用户不需要进行任何培训就能方便地访问信息;第二层是应用服务器,也就是常说的中间层,即组件层,所有的应用系统、应用逻辑、控制都在这一层,系统的复杂性也主要体现在应用层;最后的数据存储层中,数据库服务器存储大量的数据信息和数据逻辑,所有与数据有关的安全、完整性控制、数据的一致性、并发操作等都是第三层完成。

《微型机与应用》2011年 第30卷 第14期

在系统中,用户层采用 Delphi7 编制,为最终用户提供操作界面。用户在操作界面上可以输入信息,显示查询结果。具有操作简单,信息显示方式多样等特点。

业务逻辑层也采用 Delphi7 编制。这一层是整个系统的关键,系统每个功能的具体实现算法,各服务构件均在本层实现。

数据库服务器为 Windows2003 Server +SQL Server2000 模式。在数据库服务器上通过大量使用视图、触发器和存储过程实现数据的完整性控制、数据的一致性和并发操作。

采用三层体系结构设计可以有效降低开发和维护成本,简化管理,可适应不断变化的业务需求,有效提高系统安全性。这样的好处显而易见:

(1)由于数据访问是通过中间层进行的,因此客户端不再与数据库直接建立数据连接。即建立在数据库服务器上的连接数量将大大减少;

(2)可维护性得以提高。因为业务规则、合法性校验存在于中间层,因此当业务规则发生改变时,只需更改中间层服务器上的某个组件(如某个 DLL 文件),而客户端应用程序不需做任何处理,有些时候,甚至不必修改中间层组件,只需要修改数据库中的某个存储过程就可以了;

(3)良好的可重用性;

(4)事务处理更加灵活,可以在数据库端、组件层、MTS(或 COM+)管理器中进行事务处理。

本文以一个门诊部信息管理系统软件架构为例,分析研究了基于构件技术的系统软件架构,为下一步系统设计与实现提供了重要依据。与传统软件开发方法按照设计进行编程以实现目标系统不同,构件化软件的基本功能单元已经以可复用构件的方式实现,所以按照设计好的软件架构来复用构件,如何尽可能充分、正确、有效地复用软件框架,成为了下一步构件化软件实现的主要工作之一。

参考文献

- [1] 软件构件技术概述[EB/OL].<http://dev.csdn.net/develop/article/13/13254.html>.2007.
- [2] Len Bass.软件架构实践[M].车立红,译.北京:清华大学出版社,2004.
- [3] BROWN A W 著.大规模基于构件的软件开发[M].赵文耘,张志,译.北京机械工业出版社,2003.
- [4] Katharine Whitehead 著.基于组件的开发[M].王海鹏译.北京:人民邮电出版社,2003.
- [5] 黄柳青,王满红.构件中国[M].北京:清华大学出版社,2006.
- [6] 阎宏.软件的架构与设计模式[EB/OL].<http://soft.yesky.com/lesson/495/2012495.shtml>.2005.

(收稿日期:2011-03-23)

作者简介:

丁晨路,女,1986年生,在读硕士,主要研究方向:计算机网络与数据库。