

基于 UML 的计费系统的建模和实现

Development and Modelization of Network Billing System Based on UML

武建华 李 静 (暨南大学珠海学院计算机系 广东珠海 519070)

摘要:本文详细介绍了基于 UML 描述的计费系统的建模过程,包括需求模型、对象结构模型、行为模型和实现模型,并就系统实现中的一些关键技术进行了探讨。

关键词:UML 建模 计费系统 远程控制

1 引言

UML 作为一种面向对象的标准建模语言,统一了 Booch, Rumbaugh 和 Jacobson 提出的 OO 方法的概念、符号表示及模型,并对它们的语义、图形表示和使用规则作了完整而详细的定义,具有丰富的表达能力。UML 有 9 种不同的图,可从不同视角对系统建模。在需求模型中,用例图用来描述系统的功能;在对象结构模型中,类图、对象图和包图用来描述系统的静态结构;在行为模型中,序列图、协作图、状态图和活动图用来描述系统的行为,反映系统对象之间的动态关系;在体系结构模型中,构件图和部署图用来描述软硬件体系结构及通信机制^[1]。使用 UML 为软件系统建模,架起了系统分析人员、用户和程序开发人员之间交流与沟通的桥梁,为系统的设计与开发过程搭建一个广泛时效的可视化平台,大大促进软件开发各阶段表示的一致性,提高了开发效率。

计费系统建立在高校校园网之上,随着教学需要的增加和机房的对外开放,急需一个提供强大的远程控制、高效的用户管理和完善的查询统计功能的计费系统,以减轻管理人员的工作负担,提高工作效率和管理水平。在该系统的开发中,作者选择了 Telelogic 公司的 Poseidon for UML 作为建模工具^[2],构建系统的需求模型、对象结构模型、行为模型和实现模型,并对系统实现中的关键技术给出了解决方法。

需求产生角色,使用用例图进行可视化描述,清晰表达用户的目标,即发现真正的需求并以用户和开发人员理解的方式加以描述。“真正的需求”是指在实现时可以给用户带来预期功能的需求。用例分析提供了一种捕获功能需求的系统且直觉的方法,可驱动整个开发过程,是开发者准确迅速了解用户要求和相关概念的有效方法^[3]。在本系统中,通过仔细分析系统功能,抽象确定了 5 个角色即“学生”、“教工”、“服务器端管理员”、“服务器端操作员”和“客户端操作员”和 12 个用例,如图 1 所示。反映了系统总体功能和与外界的交互情况。

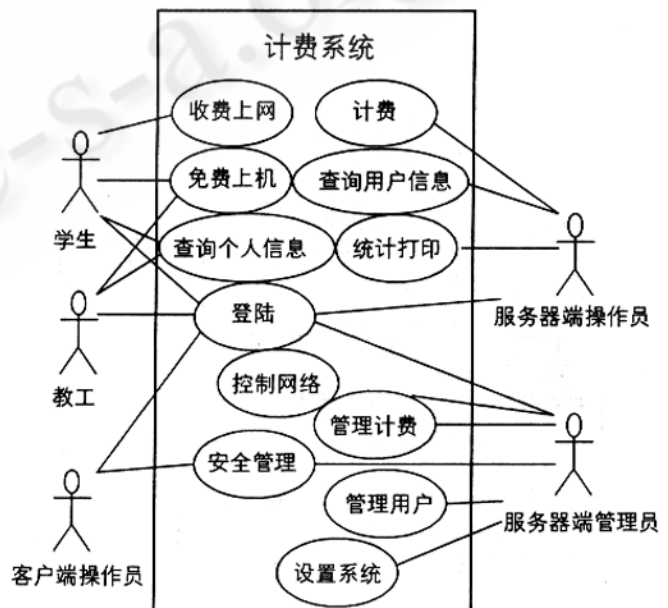


图 1 系统用例视图

2 基于 UML 的系统建模

2.1 需求模型

使用 UML 进行软件开发时,首先根据用户提出的

2.2 对象结构模型

对象结构模型主要有类图、对象图和包图组成,分层次描述系统的静态结构。类图展示了一组类、接口和协作及它们间的关系,系统可有多个类图,单个类图仅表达了系统的一个方面。通过分析用例图,可以发现相关的类并确定类之间的关联和其内部结构。在本系统中抽象出用户类、管理人员类和上机方式类。其中,管理人员类分为服务器端管理员,服务器端操作员和客户端操作员。“管理员”类具有通用的属性:用户名、密码;“服务器端管理员”类享有管理的最高权限,有管理用户、安全管理、设置系统、管理计费、查询、控制网络等操作;“客户端操作员”类的操作有管理用户、安全管理等。如图 2 所示,表示了各类之间的关联关系。对象图使用和类图相同的标识和关系,显示某一时刻类图的实例。在 UML 中,子系统被建模为一个由类组成的包^[4],包图用于表示包或类之间的关系,如图 3 所示,描述系统由子系统构成的分层结构。

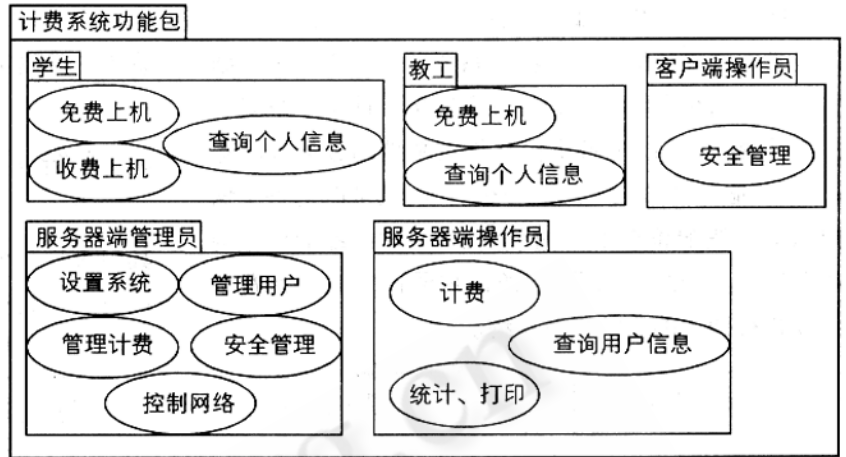


图 3 计费系统包图

2.3 行为模型

行为模型描述了系统的动态行为和组成对象间的交互关系。由顺序图、协作图、状态图、活动图等表达。顺序图展现了一组对象和由这组对象收发的消息,用于按时间顺序对控制流建模。协作图展现了收发消息的对象结构组织,按组织结构对控制流建模。协作图通过图形的几何排布显示交互中的角色,消息显示为附属在连接分类角色关系直线上的箭头,消息的顺序由消息描述前的顺序号来表示。图 4 表示管理员对学生、教工上机计费进行管理的协作图。这个用例是由管理员发出操作转换成固定的请求,发送给其他对象执行而产生的,表现了处理请求用例所涉及的十个相关对象之间相互协作的关系。由于顺序图、协作图和活动图都可以显示交互,因此当需要在 UML 中描述一个交互时,可以依据时间、空间等特征在这三种图中做出选择^[4]。

状态图展示了一个特定对象的所有可能状态以及由于各种事件的发生而引起状态之间的转移,描述其在生命周期中的行为。给出状态图可以分析系统状态的动态变迁。对于管理员管理计费实例所绘制的状态图如图 5 所示。当在“管理员信息核对”状态,管理员输入名称和密码,系统就会读出相应的用户权限,允许执行相关操作。

2.4 实现模型

实现模型包括构件图和部署图。构件图展现了一组构件之间的组织和依赖,用于对元代码、可执行的发布、物理数据库和可调整的系统建模,明确系统和部分

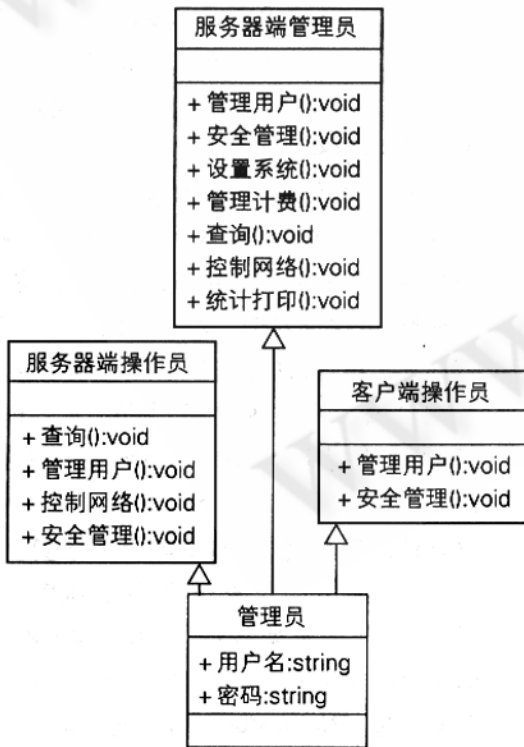


图 2 管理员类图

的功能和软件结构。部署图展现了对运行时处理节点以及其中构件的配置。它描述系统硬件的物理拓扑结构和在此结构上执行的软件分布情况。计费系统置于校园网之中,可以把各个信息管理系统的数据库服务器、应用服务器放在网络中心机房内,也可以把各个信息管理系统配置在不同的节点上,通过内部的局域网相互通信。如图 6 所示,普通用户通过局域网计费系统使用校园内的各类信息系统。

达,保证了系统分析与设计的正确性和准确性,为编码阶段的代码实现奠定了基础。

3 系统实现的关键技术

UML 一个重要特性是能自动生成软件代码。尽管其中大部分代码不能成为最终的源程序,但对编码实现有重要的参考价值。在本计费系统的软件实现中,考虑到系统对远程控制和计费安全等方面的严格要求,应用了一些新技术,现就一些关键技术进行讨论。

3.1 远程控制

虽然 TCP 协议能保证通信中数据的正确性和数据顺序,但 TCP 协议对系统资源的要求较高,当系统快速连接大量客户端的时候性能较差,因此在系统中改用程序结构较简单的 UDP 协议。为了保证数据正确性,定义了 CSRPC (Change System Remote Control Protocol) 应用协议,用来模拟 TCP 的“三次握手”,实现信息的安全。

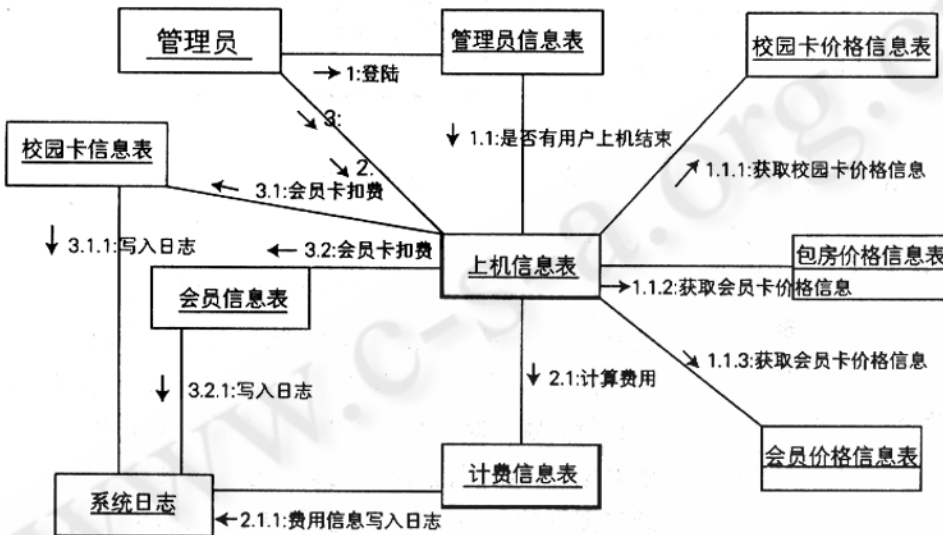


图 4 计费管理协作图

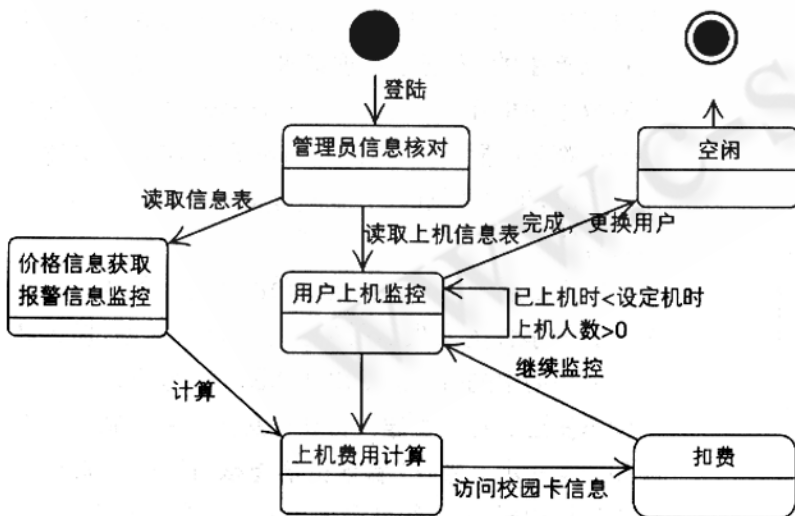


图 5 管理员计费系统状态图

通过以上对计费系统的整体设计描述可以看出,在系统建模过程中,UML 提供了灵活、易读和清晰的表

3.2 加密传输

本计费系统中传输的数据采取了单向加密和双向加密两种技术。对于客户端管理员密码采取单向加密技术,使加密信息不能还原,以保证安全。对于通信加密则采用双向加密技术,使加密的信息可以还原。本系统中的加密技术是通过 STL 来实现的。STL (Standard Template Library) 即标准模板库。从逻辑层次来看,STL 体现了泛型化程序设计的思想 (generic programming),泛型是一种软件复用技术;从实现层次看,整个 STL 是以一种类型参数化 (type parameterized) 的方式实现的。其部分代码如下:

```

AnsiString TNetMessage::EncryptText ( AnsiString text, AnsiString Key)

```

```

{ AnsiString strReturn = text;

```

```

std::string crypt;
if (text != "")
{ // 数据单向加密
    using namespace encr;
    TEncrypt <32> e( Key.c_str() );
    e.encrypt( text.c_str(), crypt );
    strReturn = crypt.c_str();
}
return strReturn;
}

```

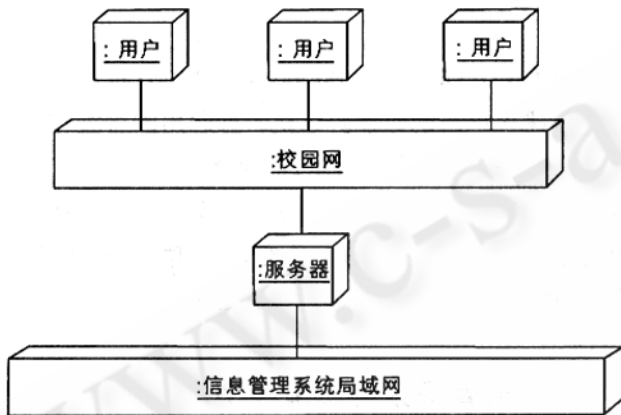


图 6 系统部署图

3.3 实时计费

为了保证计费系统的准确性采用实时计费方式,这是一种多媒体定时器方式,每过一段时间启动计费功能。多媒体定时器与 Windows 定时器函数不同,它不使用容易漏失的消息窗口,而是提供两种触发方式即回调函数或 event。大部分情况下系统使用回调函数触发方式,但 event 触发方式在多线程之间共享定时器非常有用。若它的线程优先级比其它线程的优先级高,就能很好的达到系统的安全要求。具体实现代码如下:

```

TChargeThread::Execute( void )
{ // 建立 event object
    HANDLE hTimerEvent = CreateEvent( NULL, false,
    false, NULL );
    // 设立多媒体定时器
    TimerModule -> TimerID =
    timeSetEvent( 60000, 0, ( LPTIMECALLBACK )
    hTimerEvent,

```

```

0, TIME_PERIODIC | TIME_CALLBACK_EVENT_
SET );
// 在线程被终止前始终执行线程中的内容
while ( ! Terminated )
{ // 等待 hTimerEvent 被设定为 signaled
    WaitForSingleObject( hTimerEvent, INFINITE );
    Synchronize( Charge );
}
CloseHandle( hTimerEvent );
}

```

3.4 多线程技术

为提高进程内的并发性,Windows 中引入了线程的概念,一个进程可以同时拥有多个并发线程。多线程就是指一个木马进程开启了三个线程,一个为主线程,负责远程控制的工作;另外两个为辅助线程。其中一个辅助线程称为监视线程,负责检查木马程序是否被删除或者被停止自启动。将木马技术引入到计费系统中,在客户端除正常的登录管理进程外,还有一个监控进程和 IE 监视进程,即使客户端程序关掉,监控进程也能自动重启客户端,保证计费系统的正常工作。

4 结束语

在软件开发中,好的建模是系统设计开发的基础。将 UML 应用到计费系统的设计与开发中,使开发者能够准确理解系统各部分之间的内在联系,加快软件开发过程,提高软件开发水平和软件质量,充分显示了 UML 建模系统具有较优越的整体性能,在软件开发中具有一定的现实意义。

参考文献

- 1 张玲红等, UML 在运输业务管理系统建模中的应用, 计算机工程与应用, 2004; 14: 207—209.
- 2 Poseidon UML Tool. <http://www.gentleware.com>.
- 3 Robert France, Cris Kobryn. UML for software engineers. In: Proceedings of the 23rd international conference on Software engineering. July 2001: 705—706.
- 4 Hans - Erik Eriksson 等著, 俞俊平等译, UML 工具箱 [M], 电子工业出版社, 2004. 1.