

一个基于 MVC 的通用 Client/Server 应用开发框架

高保庆¹, 傅秀芬¹, 孟利², 吕杰¹

(1. 广东工业大学 计算机学院, 广东 广州 510075;

2. 广东工业大学 机电工程学院, 广东 广州 510075)

摘要:在论述了应用开发框架的基本概念及优势的基础上, 给出了一个基于 MVC 模式的 Client/Server 通用应用开发框架, 详述了该框架客户端、代理和服务端的主要组成及其实现技术。该框架主要借鉴了目前流行的 J2EE 和 Struts 框架原理, 利用代理在客户端和服务端转发 XML 格式的业务数据, 在保持传统的 Client/Server 优势的同时, 增强了系统的可扩展性并实现了用户界面和业务逻辑数据的分离, 为大型 Client/Server 应用软件系统开发提供了一个高层次的通用开发平台, 大大提高了软件复用度和缩短了软件开发生命周期。

关键词:应用开发框架; MVC; 代理; 软件复用

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2008)07-0126-03

A Universal Client/Server Framework for Application Development Based on MVC

GAO Bao-qing¹, FU Xiu-fen¹, MENG Li², LÜ Jie¹

(1. School of Computer, Guangdong University of Technology, Guangzhou 510075, China;

2. School of Machinery and Electronic Engineering, Guangdong University of Technology, Guangzhou 510075, China)

Abstract: On the basis of discussing the application developing framework's basic concepts and advantages, gives a universal client/server application developing framework based on MVC and detailed description in the composition and main realization technology of the framework's client, agent and server. Reference to some current important principle of popular J2EE and Struts, making use of an agent to dispatch XML format business data between client and server, the framework strengthens the extensibility of software and keeps presentation logic and business logic separated on the basis of maintaining traditional client/server's advantages. As a high-level and universal developing platform for large-scale client/server application software system, it greatly enhances the degree of software reusing and shortens software development life cycle.

Key words: application developing framework; MVC; agent; software-reusing

0 引言

目前 Browser/Server(B/S)架构的应用软件越来越流行, 支持 B/S 的应用开发框架也层出不穷, 如 J2EE, Struts, Hibernate, Spring 等, 而 Client/Server(C/S)架构的应用软件由于业务变更和扩展不够灵活等原因, 其市场逐渐减少。然而鉴于企业信息系统功能复杂、业务繁多、界面响应速度要求高, 及用户界面体验要求高等特点, 在一定领域内, C/S 架构的应用软件仍然具有不可替代的优势, 因此研究能够模拟 B/S 原理和解

决 C/S 现存问题的新的应用开发框架就成为一项十分有意义的工作。

1 应用开发框架的基本概念及优势

应用开发框架^[1]是可被应用开发者定制的应用骨架。它并不是包含构件应用程序的小片程序, 而是实现了一定应用领域的通用完备功能(除去特殊应用的部分)的底层服务。使用这种框架的编程人员可以在一个通用功能已经实现的基础上开始具体的系统开发, 大大提高构件和核心业务逻辑的重用, 并在一定程度上保证了应用软件的质量和可靠性。应用开发框架提供了所有应用期望的默认行为的类集合。具体的应用通过重写子类(子类属于框架的默认行为)或组装对象来支持应用的行为。

收稿日期: 2007-10-02

基金项目: 广东省自然科学基金项目(06021484)

作者简介: 高保庆(1981-), 男, 河南南阳人, 硕士研究生, 研究方向为对等计算、协同软件; 傅秀芬, 教授, 硕士生导师, 研究方向为网络安全、数据挖掘、协同软件等。

应用开发框架不同于 MFC、VCL 等构件级框架,它指使用这些构件级框架所构造的基于特定模式和软件架构的更高级框架,框架的粒度更大,通过配置工具的简单配置,一个基本的原型系统就可以运行起来,从而大大缩短大型应用软件的开发生命周期,提高开发效率,降低开发成本。

2 框架所使用的主要技术

(1) Agent 代理技术。

代理(Agent)是一个能在特定环境中采取复杂自治行为的计算机系统。从应用的角度看,代理就是能自动执行用户委托任务的计算实体。文中所使用的软件代理^[2],是整个应用开发框架的重要功能部件,包括客户端代理和服务端代理,可能运行在同一台计算机上,也可能运行在不同的计算机上。客户端代理用于转发用户的业务请求,服务端代理接收业务请求并返回相应的业务数据集。

(2) MVC 模式。

MVC 模式^[3]即“Model - View - Controller”(模型 - 视图 - 控制器)。MVC 应用程序中,Event(事件)导致 Controller 改变 Model 或 View,或者同时改变两者。只要 Controller 改变了 Models 的数据或者属性,所有依赖的 View 都会自动更新,如图 1 所示,MVC 模式已经在大部分 JAVA 开发框架中得到充分体现^[4],它有效地分离了用户表现层(用户界面)和业务逻辑实现,最大程度上实现了业务变更和扩展对系统其它部分的影响,增加了系统的可扩展性。

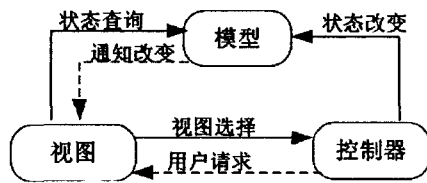


图 1 MVC 模式原理

3 基于 MVC 的 C/S 通用应用开发框架

本框架整个结构按 MVC 模式设计,主要包括以下组成部分:客户 UI 主界面动态生成控制器、系统界面风格统一控制器、客户端代理(负责与服务器进行通讯)、服务器端代理、服务端业务逻辑分发代理、服务器事务控制器、权限控制器等,如图 2 所示。

3.1 实现目标

该框架主要致力于实现以下目标:

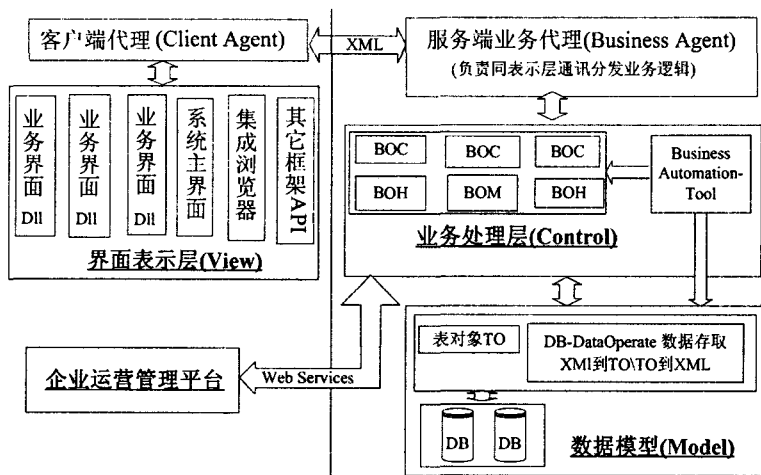


图 2 基于 MVC 的 C/S 通用开发框架体系结构

- * 系统主界面包括菜单、导航条、系统风格等都可使用框架的配置工具(由框架统一开发配置)。
- * 模块化的业务组件可配制。
- * 客户端/服务器通过各自代理自动通讯。
- * 应用开发框架自动管理数据库存取,提供数据库缓冲池、事务处理、并发控制等,无需编程人员干预。
- * 具体软件开发过程中,界面程序员(客户端)和业务程序员(服务端)完全分开,界面程序员只关注系统的客户体验,业务程序员只关注业务逻辑,从而提高各自的工作质量和效率。

3.2 框架各个主要模块功能、运行原理及特点

1) 系统主界面通过客户端代理与服务器端代理自动通讯,框架读取数据库中的配制信息进行各个模块的动态加载。图 3 描述了客户端框架读取框架配制信息完成框架内基本动作的流程。

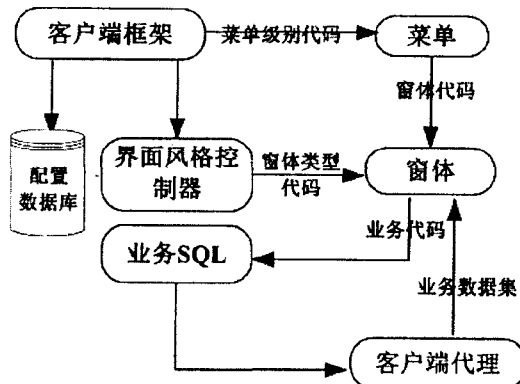


图 3 通用 C/S 应用开发框架客户端原理

2) 在具体应用开发中,对于从框架基类中继承的各种类型窗体,将自动被纳入系统统一风格控制器中,实现了当系统风格改变时,各具体业务界面不需作任何的变动(类似 B/S 中的 CSS 的效果)。

3) 所有客户端界面均以 DLL 模块的形式存在,由客户端代理根据模块配制信息进行动态加载(实现

了界面表示层的可配制)。

4) 所有客户端要展示或处理的数据由客户端代理通过统一接口,以 XML 的格式与服务器端代理进行交互。

5) 服务器根据客户端传递过来的命令代号和业务类代码(业务界面与业务类的对应关系在配制平台中已配制好)查找到相应的业务处理类,并将客户端的请求交给该业务类进行处理(实现了业务逻辑的可配制)。

6) 业务处理类的具体方法由程序员去实现,这些均继承于系统提供的统一基础类库(实现了方法与接口的统一);数据库所有的表映射成 TO(表对象,该对象由系统提供工具根据表结构自动生成,下文将有论述),程序员只需在业务处理中对 TO 进行赋值(调用 TO 的 Set 方法),然后将该 TO 交给框架去处理,平台自动管理事务(模仿 Hibernate 和 Spring)^[5]。

7) 根据目标系统的需求,还可在业务处理类中通过 Web Service 调用外部的 Web 应用服务(实现了 C/S 与 B/S 的有效整合)。

8) 实现了前台界面与服务器业务逻辑的完全脱离:前台界面将更专注于界面的美观与客户体验,不需关心后台业务逻辑的具体实现,后台业务处理类响应客户端请求,可将数据存储在本地的数据库中或取本地应用数据库数据,也可以通过 Web Service,调用 Web 应用服务器提供的远程接口,读取或存储远程数据库中的数据。

9) 前台主应用程序与界面展示 DLL 是完成脱离的,且主界面上的模块菜单为动态生成的:不仅实现了前台界面的可配制,同时前台主应用程序对于所有 C/S 应用都是通用的,从而实现了主程序的高可复用性。

10) 事务控制与权限控制均在框架内部得以实现,对于用该应用框架开发的系统,无需关心该部分的实现。

11) 由于负责业务处理的服务器端各类数目和各类之间的关系(如图 4 所示)都已经确定,因此服务器端各类:BOC(TC, Business of Object), BOM(TM, Business of Manager), BOH(TH, Business of Helper), TO (Table Object)均有框架提供的自动化代码工具根据数据模型及其之间的关系自动生成,服务器端程序员只需要在这个基础上根据系统具体的详细设计做部分修改。

4 框架可用性及与传统 C/S 框架比较分析

1) 某大型软件科技集团已经使用本应用开发框架开发了若干重要项目,包括一款个人理财产品和一

个大型分布式院线系统(服务端实现技术有部分改进),本框架表现出良好的稳定性和可靠性,同时大大缩短了软件开发生命周期。

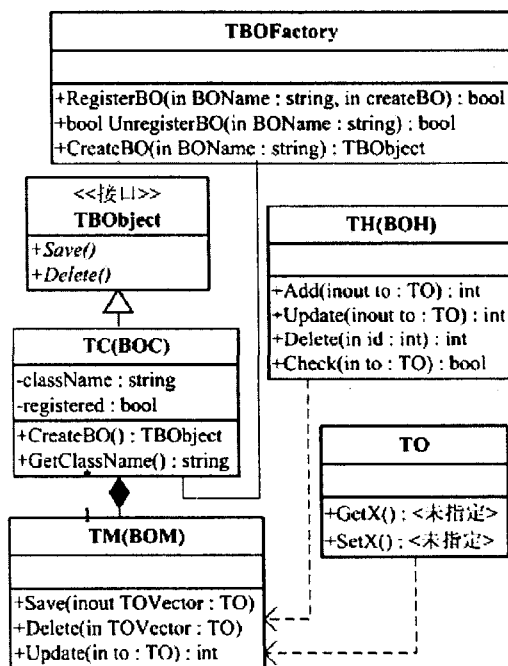


图 4 服务器端业务类之间关系

2) 下面将主要在 6 个方面对本应用开发框架与传统 C/S 框架的主要特点作一个对比,如表 1 所示。

表 1 基于 MVC 的通用 C/S 应用开发框架与传统 C/S 框架的主要特点对比表

比较项目	基于 MVC 的开发框架	传统的 C/S
重用性	可高度重用前台主界面和业务逻辑都是可配制的,界面和业务处理都是以组件的形式存在,很好地发挥了面向对象的思想,因此该平台可适用于任何 C/S 软件产品的开发,界面和业务组件	界面与业务没有分离,几乎不能重用
与其它系统整合	服务器端通过 Web-services 可以与任何语言开发第三方软件进行集成、整合,而前台界面不需作任何变动	使系统整合变得非常困难,且代价很高
程序结构	实现了 MVC 的思想:前台界面与业务处理逻辑完全分离开,使程序结构清晰明了,耦合性较低	前台界面与业务逻辑绞合在一起,程序结构比较混乱,耦合性比较高
效率	实现了分布式处理,使得处理效率更高	没有实现分布式处理,处理效率不是很高
界面美观性	前台程序将更专注于界面的设计,使界面更加美观;同时系统风格由控制器统一控制	美观得不到有力保障
代码规范性	采用平台约定的统一规范来编码;继承统一的基类,实现统一接口,使得程序代码更加规范	由于程序员得不到有力的约束,代码规范将得不到有力的保障

5 结束语

指出了传统 C/S 架构的应用软件所存在的问题

(下转第 132 页)

b. 查看日志服务器上的 Sebek 服务器端页面, 没有发现任何键击信息, 判断是被恶意软件所感染。

c. 查看 HoneyWall 上的 Snort 报警信息, 发现恶意软件通过攻击 RPC 端口感染蜜罐主机, 随后蜜罐主机向攻击主机发起 TFTP 连接获取数据。

d. 分析 HoneyWall 上的记录全部网络流量的 pcap 文件, 根据攻击主机 IP、TFTP 端口找到对应网络连接数据包。通过 ethreal 重组网络连接发现获取文件后存放在蜜罐主机上的

c:\windows\system32\msgdl1.exe, 在蜜罐主机对应目录找到恶意软件样本。

e. 通过 www.virustotal.com 查询, 确认该恶意软件为“高波”蠕虫变种。

f. 继续查看蜜罐感染“高波”蠕虫变种后的行为。在 IPTables 的日志中发现蜜罐对外部主机的 RPC 端口发起大量的网络连接, 但由于 IPTables 的外出流量限制均被丢弃^[4]。

3.2 虚拟蜜网

使用 VMWare 或 User-Mode Linux 将多个虚拟系统设置在一台物理机上, 通过这种方式, 在防火墙构架下, 就构成了虚拟 honeynet (如图 2 所示的网络拓扑)。虚拟系统可以使我们在单一主机系统上运行几

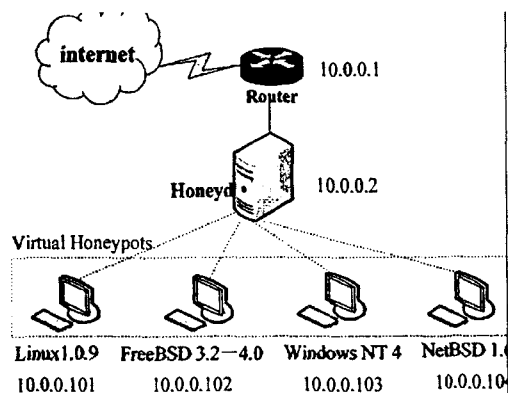


图 2 虚拟蜜网网络拓扑

(上接第 128 页)

及开发支持 RAD 的 C/S 框架的必要性, 给出了一个基于 MVC 的通用 C/S 应用开发框架, 介绍和分析其主要功能及运行原理, 并在各个主要方面与传统 C/S 框架做了比较分析。通过实际项目的使用, 验证了框架的可行性和可靠性, 大大提高了软件复用率和缩短了软件开发周期, 保证了软件质量及按时交付。

参考文献:

[1] Chen Xin. 应用框架的设计与实现——.NET 平台[M]. 温

台虚拟计算机。虚拟蜜网大大降低了成本、占用空间以及管理蜜罐的难度。此外, 虚拟系统通常支持“悬挂”和“恢复”功能, 这样就可以冻结安全受危及的计算机, 分析攻击方法, 然后打开 TCP/IP 连接及系统上的其它服务^[5]。

4 结束语

讨论了基于 Honeypot 理论的网络入侵诱骗技术, 它不同于一般的被动防范, 而是一种主动防御的安全技术。随着越来越多的用户开始在网络中使用 Honeypot, 更多的产品将会被开发。蜜罐技术的最终目标是能够对高智商和反应敏捷的黑客进行欺骗, 而现有的蜜罐技术还远未成熟, 在欺骗的复杂性、部署维护的难度及范围上都还存在着较大的缺陷, 所以要形成成熟的蜜罐技术是非常具有挑战性的。另外, 蜜罐技术作为网络安全中的一个新兴领域, 是对现有安全体系的一个重要补充, 对提高网络安全性起着重要的作用。随着网络入侵类型的多样化发展, 蜜罐也必须进行多样化的演绎, 只有这样才能更好地保护国家集体以及个人的财产安全。

参考文献:

- [1] 徐超汉, 柯宗贵. 计算机网络安全实用技术[M]. 北京: 电子工业出版社, 2005.
- [2] 张 斌. 黑客与反黑客[M]. 北京: 北京邮电大学出版社, 2004.
- [3] 刘彦保. 入侵诱骗技术分析及其模型建立[J]. 河南科学, 2006, 24(4): 532-535.
- [4] 诸葛建伟, 张芳芳, 吴智发. 撒下蜜网研究黑客[J]. 电脑安全专家, 2005(7): 21-23.
- [5] Artail H, Safa H, Sraj M, et al. A hybrid honeypot - framework for improving intrusion detection systems in protecting organizational networks[J]. Computers & Security, 2006, 25(4): 274-288.

显, 靳向阳, 译. 北京: 电子工业出版社, 2005.

- [2] Patrick A S. Building trustworthy software agents[J]. Internet Computing, IEEE, 2002, 6(6): 46-53.
- [3] Gamma E. 设计模式: 可复用面向对象软件的基础[M]. 李英军译. 北京: 机械工业出版社, 2007.
- [4] Johnson R. J2EE Development Frameworks[J]. IEEE Computer, 2005, 38(1): 107-110.
- [5] 杨 涛, 周志波, 凌 力. 基于 Struts 和 Hibernate 的 J2EE 快速开发框架的设计与实现[J]. 计算机工程, 2006(10): 83-85.