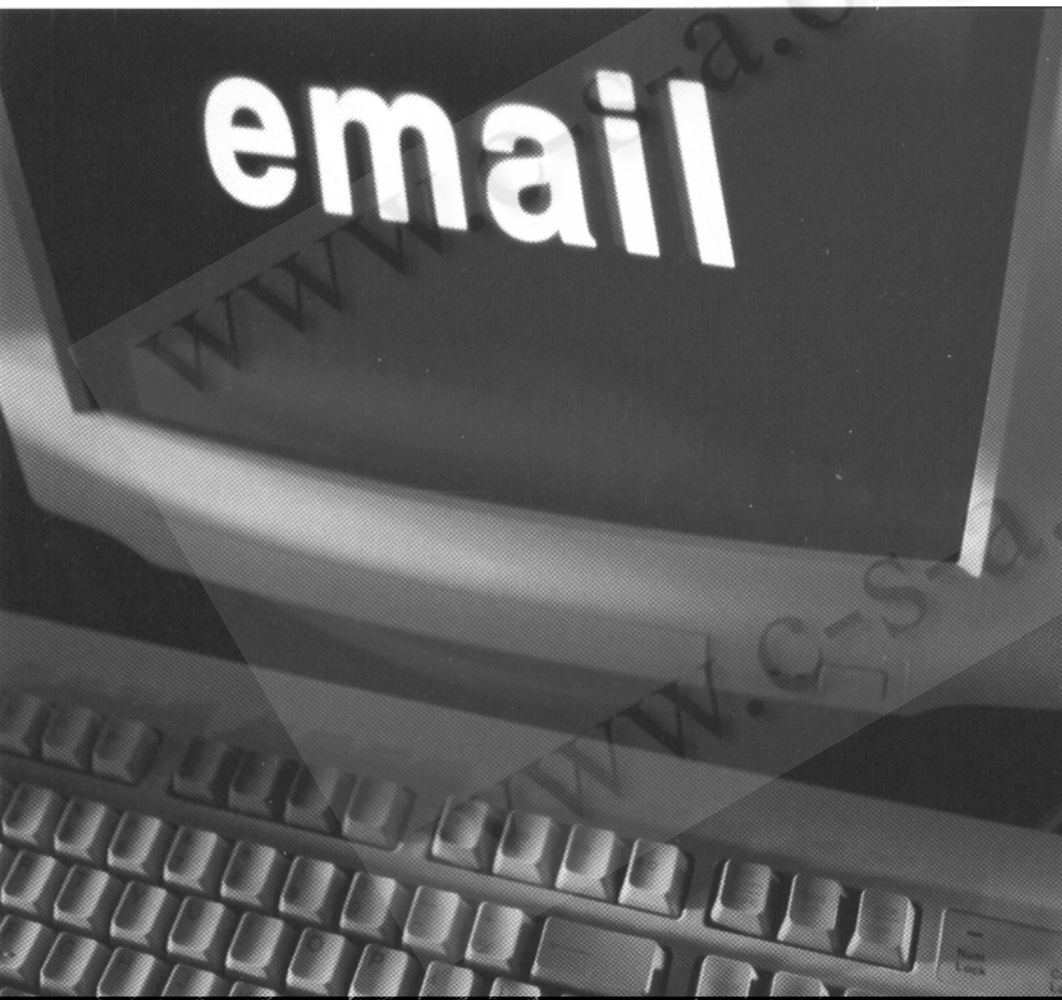


# Producing Object Oriented Components with Rational Rose

## 利用 Rational Rose 创建面向对象组件



**摘要:** 由于组件具有很多优点, 组件的开发引起了越来越多的关注, 其中尤以 COM+ 为特出典型, Rational 公司的 Rose 作为组件的开发工具, 非常高效实用, Rose 能够自动产生源代码, 支援多种程序语言和多种平台, 本文简要阐述了利用 Rational Rose 和 VC++ 开发 COM+ 的技术。

**关键词:** 组件 COM+ Rational Rose

### 1 引言

面向对象技术的优势是使设计者将软件中的棘手问题 - 主要是程序细节与实现, 利用其封装特性隐蔽起来, 从而提高软件可重用性、可靠性、健壮性、可扩展性、分布性和可存储性。面向对象模型化技术给出了软件系统的体系结构, 并引入了软件构件与复用的思维方法。组件 (Component) 是可复用的软件组成成分, 可被用来构造其他组件和程序, 它可以是被封装的对象类、接口定义、类树、一些功能模块、软件框架 (Framework)、软件构架 (或体系结构 Architectural)、文档、分析件、设计模式 (Pattern) 等。

组件构架从本质上讲是可定制的, 用户可以根据他们的需求, 用增加、卸掉与替换等方式来定制应用软件系统。

经过十几年的发展, 构件本身的构件模型及其规范已经比较成熟, 如较有影响的 Windows 环境下的 COM 与 DCOM, UNIX 并跨平台的 CORBA, 以及 OS/2 下的 SOM。目前较有影响的构件规范与技术主要有 CORBA、OLE/ActiveX 和 JavaBeans, 其应用日趋广泛。

目前, 由于 Microsoft 的 Window 平台的巨大市场份额, 基于 Window 平台的组件模型与技术 COM+, 近年来得到了越来越广泛的应用。

## 2 Rational Rose 与 COM+

### 2.1 Rational Rose 简介

Rational 公司的旗舰产品 Rational Rose 进入中国一年来,已受到诸多开发商青睐。它面向 Unix 和 Windows NT/95 平台,提供了一套满足 Client/Server、企业分布和实时系统环境需求的解决方案。它提供可视化建模和双向工程,充分支持 UML (Unified Modeling Language, 统一建模语言)。通过使用通用语言 UML 实现各种应用程序的设计与开发。

应用该开发工具可以将各种模式定义图形化,并由此自动生成 C++、Java、Visual Basic (VB) 等语言的源代码。新版 Rational Rose 在此基础之上又新增加了数据建模功能,因此可以设计出 Oracle 和 SQL Server 及 DB2 乃至 Access 等 RDB(Relational Data Base, 关系数据库)的结构与定义。设计信息也可以作为数据定义语言 (DDL, Data Definition Language) 可输出至定义各种不同类别的数据库上,也可以从各类数据库上生成 DDL (即所谓逆向工程, Reverse Engineering)。

### 2.2 COM+ 简介

COM (Components Object Model) 是软件组件互相通信与操作的一种协议与方式,它是一种二进制级互操作和网络通信标准,允许任意两个遵守此协议的组件互相通信,而不管它们是在什么地方的计算机上运行(只要计算机是相连的),不管各计算机运行的是什么操作系统(只要该操作系统支持 COM),也不管该组件是用什么语言编写的。COM 还提供了位置透明性:当您编写组件时,其他组件是进程内 DLL、本地 EXE 还是位于其他计算机上的组件,对您而言都无需关心,(当然会有性能区别,但是,即使改变了其他组件的位置,您也不必重新编写什么,这是关键所在。)COM+ 的特点:

(1) 真正的异步通信。COM+ 底层提供了队列组件服务,这使客户和组件有可能在不同的时间点上协同工作。

(2) 事件服务。新的事件机制使事件源和事件接收方实现事件功能更加灵活,利用系统服务

简化了事件模型,避免了 COM 可连接对象机制的琐碎细节。

(3) 可伸缩性。COM+ 的可伸缩性来源于多个方面,动态负载均衡以及内存数据库、对象池等系统服务都为 COM+ 的可伸缩性提供了技术基础,COM+ 的可伸缩性原理上与多层结构的可伸缩特性一致。

(4) 继承并发展了 MTS 的特性,从 COM 到 MTS 是一个概念上的飞跃,但实现上还欠成熟,COM+ 则完善并实现了 MTS 的许多概念和特性。

(5) 可管理和可配置性。管理和配置是应用系统开发完成后的行为,在软件维护成本不断增加的今天,COM+ 应用将有助于软件厂商和用户减少这方面的投入。

(6) 易于开发。COM+ 应用开发的复杂性和难易程度将决定 COM+ 的成功与否,虽然 COM+ 开发模型比以前的 COM 组件开发更为简化,但真正提高开发效率仍需要借助于一些优秀的开发工具。COM+ 标志着 Microsoft 的组件技术达到了一个新的高度,它不再局限于一台机器上的桌面系统,它把目标指向了更为广阔的企业应用,乃至 Internet/

Intranet/Extranet 应用系统,因此可以说 COM+ 与多层结构模型以及 Windows 操作系统为企业应用或 Web 应用提供了一套完整的解决方案。

## 3 应用实例

### 3.1 CRM 系统

CRM (Customer Relationship Management, 客户关系管理) 是一种“以客户为中心”的管理策略,通过运用信息技术手段,使企业的销售管理、市场营销、客户服务等效率与有效性,增加企业的竞争力与客户价值。

CRM 是面向电子商务企业 (E-Business Oriented Enterprise) 一个重要的跨职能部门系统,它为现代企业应对所谓的 3C (Competition, Customer, Change) 提供了一个有效的手段。

以下是我们在开发 CRM 系统中,利用 Rational Rose 实现销售管理系统中订单管理的例子。

#### 3.1.1 系统类模型图

为方便起,在此我们省略用例图仅建立类图,如图 1 所示。

#### 3.1.2 各类之间的关系

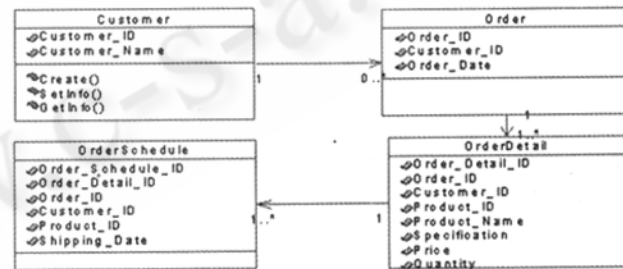


图 1 系统类模型图

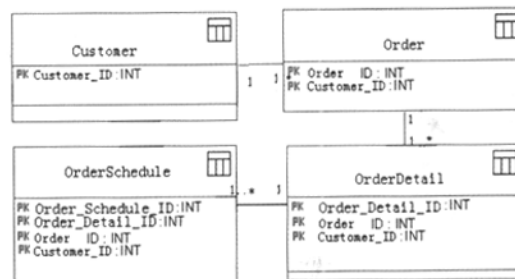


图 2 数据库表图

类 Customer 与类 Order 是: 1:[0,n] 的关联关系  
 类 Order 与类 OrderDetail: 1:[1,n] 的关联关系  
 类 OrderDetail 类 OrderSchedule: 1:[1,n] 的关联关系

### 3.1.3 数据库表图

数据库表图如图 2 所示

### 3.1.4 生成过程及程序

一般的生成步骤:

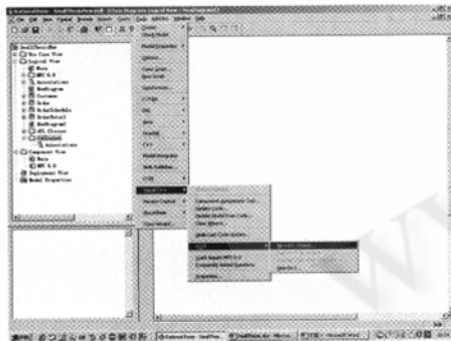
- (1) 生成类图
- (2) 生成相应类的 ATL 对象
- (3) 为类指定 VC++ 组件
- (4) 在 VC++ 中定义 COM+ 智能指针并设定及其 Activate、CanBePooled、Deactivate 方法
- (5) 实现所有方法的功能

### 3.2 股票出售程序生成过程

我再举一关于股票出售的例子, 在本例中共有四个对象, 分别为股票拥有者、帐号、帐号细节、报价单。

以下我们以类 Customer 为例阐述如何建立 COM+。

- (1) 建立新的 ATL 对象 (Tools > Visual C++ > COM > New ATL Object, or right-click on the class) 在弹出窗口中 short Name 栏输入 Customer,



- (2) 指派类为相应组件

——从主菜单中执行指派组件命令

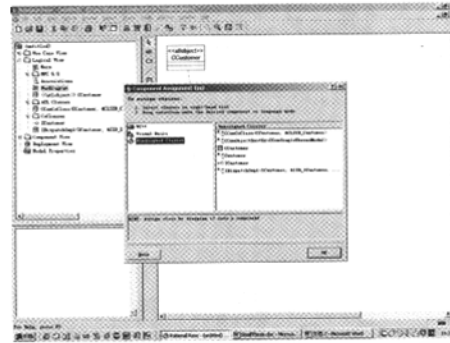
Tools > Visual C++ > Component Assignment Tool,

——在弹出窗口中选中指派类文件夹,

——在右边列表中把 Customer 类指派给 VC++ 文件夹,

——在 IUser 接口中加入 Create (Customer-

ID int, Customer-Name BSTR) 方法. 从主菜单中  
 选更新代码命令。



Create 方法主要实现增加一个客户的功能

(3) 在 VC++ 综合编辑窗口中,

定义 COM+ 智能指针 m-spObjectContext;

CComPtr<IObjectContext> -spObjectContext;

增加 Activate、CanBePooled、Deactivate 方法,

在相应数据库表读写完后加上:

如果成功: m-spObjectContext->SetComplete();

如果失败: m-spObjectContext->SetAbort();

以下是程序清单:

```
#include "stdafx.h"
#include "MyCustomer.h"
#include "CCustomer.h"
HRESULT CCustomer::Activate()
{
    HRESULT hr = GetObjectContext (&m-
    spObjectContext);
    if (SUCCEEDED(hr))
        return S-OK;
    return hr;
}
BOOL CCustomer::CanBePooled()
{
    return TRUE;
}
void CCustomer::Deactivate()
{
    m-spObjectContext.Release();
}
```

STDMETHODIMP CCustomer::Create(long Customer-ID, BSTR Customer-Name)

```
{
    AFX-MANAGE-STATE(AfxGetStaticModuleState())
    return S-OK;
}
```

## 4 结束语

Rose 支持所谓的双向工程(round-trip engineering)

(1) 支持正向工程: 提供使用者建立分析与设计模式, 并可由分析与设计模式自动产生程式码,

(2) 支持反向工程: 提供使用者将原始程式码, 转换成分析设计模式。

Rose 能够自动产生源代码, 支援多种程序语言, 包括有: Visual Basic、PowerBuilder、C++、Forte、Java、Smalltalk、SQL Windows 等 7 种语言, Rose 支持多种平台, 包括有: Windows 95、Windows NT、Alpha NT、SGI、Solaris、AIX、Digital UNIX、HP-UX。使用者可以藉由 Rose 开发基于不同平台的可用软件系统。

利用 Rose 开发 COM+ 非常高效、简便, 既可以用 Rose 直接生成源代码, 同时在代码生成后又可利用逆向工程修改模型, 还可选用 VB、VC 等多种语言生成 COM+, 所以 Rose 是一个开发 COM+ 的不可多得的快速开发工具。■

## 参考文献

- 1 UML 设计核心技术, 蒋慧、吴礼发, 北京希望电子出版社, 2001 年。
- 2 Visual C++ 6 DCOM 开发指南, Jin Maloney、田雨、刘云层、郑轶峰, 清华大学出版社, 2000 年。
- 3 COM+ 开发指南, John Paul, Muller, 清华大学出版社, 2000 年。
- 4 COM+ 技术解决方案设计, Ray Brown, Wade Barnon, 梁王柱译, 机械工业出版社, 2001 年。
- 5 Mastering UML with Rational Rose Wendy Boggs, Michael Boggs, SYBEX, 2000 年。
- 6 Mastering COM and COM+ Ash Rofail, Yaiser Shohoud, SYBEX, 2000 年。