

利用 Notes 和 ActiveX

实现集成化企业信息系统

徐锡荣 黄立平 (同济大学经济与管理学院 200092)

摘要: Notes 是一种群组软件, 近年来在办公自动化(OA)和工作流控制等方面显示了强大的功能, 但是这些 Notes 开发的应用系统往往与企业管理信息系统(MIS)相脱离, 不具备或者很少具备数据处理能力, 这是由 Notes 自身的功能决定的, 本文将探讨如何通过 Notes 中使用 ActiveX 组件技术实现企业 OA 与 MIS 的衔接。

关键词: 组件 信息集成 Notes ActiveX DECS

上海市港务局为了实现东方第一大港的目标, 近几年投入大量资金进行码头建设。一流的项目需要一流的管理, 因此从 1997 年二期项目开始, 就开始了管理信息系统的开发和建设, 但由于当时 Java 的体系尚不成熟, 我们采用 Java 进行企业级开发的尝试没有获得完全成功。从三期项目开始, 我们选用成熟的 Notes 作为基础平台, 集成其他不同开发工具和开发技术, 特别是 ActiveX 技术的充分利用, 基本满足了大型工程项目管理的需要。

1 Notes 广泛应用及其在事务和事务处理方面的不足

Notes 作为一种群组工作软件, 它的群件模式包含着三个技术领域: 通信、协同操作和协调运行。它在电子邮件(e-mail)应用、群组信息共享、以及业务过程自动化方面扮演了重要的角色。因此, Notes 作为一种开发工具, 在办公自动化和工作流控制等方面得到了广泛的应用。

Notes 的核心业务能力采用的是客户/服务器工作模式, 数据库是它的信息存放中心。Notes 数据库是一种非结构型数据库, 这种非结构型的数据库以文本信息为基础, 它所存放的信息是对客户问题或请求的叙述性文字, 因此特别适合于电子邮件、文本文档等非结构化信息的存储, 具有强大的文档处理能力。而另一方面, 这种非结构化的数据库对结构型数据, 即关系型数据的处理能力非常弱, 这就大大的限制了 Notes 在事务处理方面的能力, 限制了 Notes 在企业应用开发中的使用范围, 造成了当前企业应用中 OA 与 MIS 相互独立的状况, 形成企业信息孤岛。

在实际的企业应用当中, 结构化数据与非结构化数

据通常都是伴随着实际业务同时发生的, 但是互相独立的各类企业应用系统往往造成这些数据的人为分割。以工程项目建设中的一份合同为例, 假设该合同中包含以下内容: 合同编号、名称、甲方、乙方、合同日期、具体的合同条款、以及一份甲方供料定额清单。在 OA 与 MIS 分离的情况下, 这份合同的处理往往需要经过两套手续: 一是合同文本作为一个整体通过 OA 在企业内部实现流转、审批过程, 最后以文档格式保存在 OA 中, 成为备查资料; 与此同时, 还要人工将其中的一部分信息, 如编号、名称、乙方、日期、甲方供料定额清单等拆离出来, 重新再输入 MIS 中, 实现数据的查询和统计汇总, 以达到实时控制和辅助管理、辅助决策的目的。这就造成了大量的人力物力的浪费, 没有在最大范围内实现信息系统的价值。

2 ActiveX 技术简介

ActiveX 技术基于构件对象模型(COM), 是 OLE 和 OCX 技术结合和改进的产物。一方面, ActiveX 继承了 OLE 所具有的基于对象的、可定制、可扩展的服务能力, 可以用于解决不同系统之间的互操作问题; 另一方面, ActiveX 与 OCX 一样, 提供了桌面环境中的 Windows 构件软件开发的标准解决方案, 可以解决软件重用的问题。

ActiveX 技术实现了 windows 操作系统级的兼容, 支持 windows 平台下的各类开发工具, 如 Visual C++、Visual Basic、以及 PowerBuilder 等。一方面, ActiveX 组件继承了其开发工具所具有的几乎全部的事件、属性、方法, 可以实现其开发工具所具备的强大功能; 另一方面, 它也可以通过开发新的事件、属性和方法, 提供其宿主嵌入环境(如 Notes)使用, 实现对 ActiveX 的引用和控制。这

就使得系统开发过程中,可以使用ActiveX组件来扩展主开发工具所不具备的功能。

3 OA 与 MIS 集成环境下的业务数据分解

OA 与 MIS 的集成首先须要对两者在功能上进行适当划分,而功能的划分首先就是各自所能处理的数据的确定和划分。在 OA 与 MIS 应用集成的情况下,业务数据按照其应用范围可以划分为三类,下面还是以建设合同为例来说明:

(1) 非结构化的数据,这类数据主要是一些文档、图片、图纸等不便于用关系型数据处理的大集合数据,建设合同中的文字性条款就可以归为这一类。这种完全非结构化的数据一般只是在企业内部进行共享和流转,因此,只需要存放在 OA 的文档数据库中就可以了;

(2) 结构化数据,例如建设合同中的甲方供料定额清单,此类数据以数值性数据居多,是业务过程中的关键,用非结构化数据库很难进行处理,因此需要进入关系型数据库,由 MIS 进行处理,达到辅助管理、辅助控制的目的;

(3) 另外还有一些数据,介于前两类数据之间,本文暂且套用一个概念,称之为半结构化数据,这些半结构化数据无论在 OA 中还是在 MIS 中,都往往以关键字的形式出现,它既是 OA 中文档归类及以后检索的依据,又往往是 MIS 中数据统计查询的关键字,是 OA 和 MIS 两类系统模块都不可缺少的一部分。建设合同中的合同编号、甲方、乙方、签订时间就属于此类,因此这些数据不仅在要在 OA 数据库中存储,也必须在 MIS 中保存相应的拷贝。

4 使用 ActiveX 技术实现 OA 与 MIS 集成的解决方案

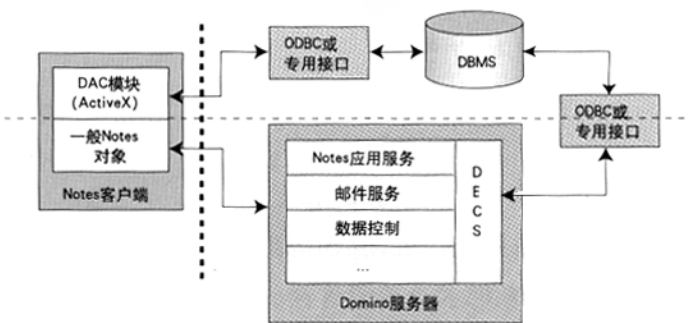


图 1 以 Notes 为基础的 OA 与 MIS 集成系统体系结构

图 1 是以 Notes 为基础的 OA 与 MIS 集成系统体系结构

构图,这种体系结构是以客户/服务器模式(C/S)为基础的。

在该体系结构中,粗虚线左侧是系统的客户端,客户端以 Notes 表单为基础,表单包含两类对象:一类是普通的 Notes 对象,另一类是利用其他开发工具开发的完全符合 ActiveX 标准的数据存取组件(Data Access Controls, DAC)。粗虚线右侧,是服务器部分,服务器部分是整个系统的数据和服务中心,包括两个部分:一个是 Domino 服务器,它处理非结构化的数据,如邮件、文档、以及二次开发的各类 Notes 表单等,提供与 OA 有关的各类应用服务;另一个是关系型数据库,它处理结构化的数据,是企业 MIS 中心。两个数据中心通过 Notes 的 DECS 接口技术连接,保证两者之间的部分数据实现动态一致。

这种体系结构是完全针对前面提到的三种类型的数据处理要求而设计的:

非结构化数据:通过表单中的一般 Notes 对象实现输入输出,数据作为 Notes 文档的一部分,直接进入 Domino 服务器进行保存;

结构化数据:通过表单中的 ActiveX 组件,与关系数据库进行直接连接,实现数据的输入、展示、以及处理功能;

半结构化数据:通过表单中的一般 Notes 对象实现输入输出,数据进入 Domino 服务器后,再利用 DECS 接口,实现关系型数据库中的备份。

在这种体系结构中,由于普通 Notes 对象与 ActiveX 组件分别以独立的方式与各自的数据库服务器交互,因此,在用户界面中如何保持两部分数据的一致性是需要考虑的问题。由于 ActiveX 组件都有公开的属性、方法、事件来供宿主嵌入环境调用,这个问题可以在 Notes 编程环境中通过调用 ActiveX 组件自定义的方法来实现。

5 开发实例

PowerBuilder (PB) 在数据库开发方面的强大功能是当前任何其他开发工具所无法比拟的,下面以 PB 提供的 Window ActiveX 技术为例,来介绍 ActiveX 在 Notes 中的应用。

还是以前面所举的项目建设一份简化的合同为例

(1) 按照前边的数据拆分,关系数据库中至少要包含两个数据表,一个是合同概况(t_contract),另外一个是在合同中的甲方供料清单(t_contract_material),表结构如下(按 Sybase 数据库标准):

```
create table t_contract (
```

```
S_code char(20) not null, // 合同编号
S_name char(50) not null, // 合同名称
S_cdate smalldatetime null, // 签订时间
S_contractor char(50) null // 承包商
)create table t_contract_material (
S_contract_code char(20) not null, //合同编号
S_material_code char(20) not null, //物资种类
S_amount decimal(15,2) null, // 定额数量
... // 其他内容
)
```

(2) 使用 PB 进行 ActiveX 组件设计。PB Windows ActiveX把每个子窗口及子窗口中的所有对象封装成一个 ActiveX 组件，在这个子窗口中可以包含所有常用的 PB 控件，也可以继续打开其他窗口。本例中，在 Project.pbl 中建立一个子窗口 w_contract，该窗口中主要功能就是对一份合同的甲方供料清单进行处理，包括增、改、插、删以及数据检索，除此之外，为了保持组件中的数据与所嵌入的 Notes 表单的其他部分一致，提供一个自定义的窗口函数，格式是：

```
wf_set(String para) return none
```

其中,para是一个字符串类型的参数,要求提供以“+”分隔的两个参数,格式是“code+right”,其中code是合同编号,用来控制w_contract中的数据与Notes表单中的数据的一致性,“right”是操作控制,用来控制操作权限。

把Project.pbl编译成Project.pbd,存放到“c:\mydir”目录下。

(3) 表单设计,非结构数据和半结构化数据全部须要通过普通 Notes 对象进行处理,设计一个表单 Frm_contract,其中包括以下几个域:合同编号(D_code)、合同名称(D_name)、甲方(D_owner)、乙方(D_contractor)、签订时间(D_cdate)、合同文本(D_ctext),另外,还要添加一个 PowerBuilder Windows ActiveX 对象。然后添加代码:

```
Dim ocx_contract '定义全局变量
'----Notes 表单中对 PB ActiveX 对象进行初始化
Sub Postopen(Source As Notesuidocument)
...
Set ocx_contract=source.getobject("PowerBuilder
Window Control")
Ocx_contract.liblist="c:\mydir\project.pbd;"
Ocx_contract.pbwindow="w_sel_prjinvl"
End Sub
'---- Notes 表单中对 PB ActiveX 对象进行控制:
```

```
Sub Click(Source As Button)
Dim ws As New notesuiworkspace
Dim uidoc As notesuidocument
Dim item As Variant
Set uidoc=ws.currentdocument
item=uidoc.document.getitemvalue("D_code")
If Cstr(item(0))="" Then
Exit Sub
End If
Dim args(0)
args(0)=item(0) + "+1"
ocx_contract.invokepbfuction "wf_set",1,args
End Sub
```

(4) 利用 DECS,把 Notes 表单中处理半结构化数据的域同关系数据库中的关系表进行绑定,本例中的绑定关系如表 1 所示。绑定之后,这些半结构化的数据就在 Domino 数据库和 Sybase 数据库中产生了两个内容完全相同的备份。而合同文本(D_ctext)等一些非结构化的数据则只在 Domino 数据库中出现。

Domino: Frm_contract 表单域	Sybase: t_contract 字段
D_code	S_code
D_name	S_name
D_contractor	S_contractor
D_cdate	S_cdate

(5) 要想让已经嵌入了 PB Window ActiveX 组件的表单正常运行,必须注意运行环境的设置,这些设置包括:

- ① 在 Windows 系统中注册 pbrx70.ocx,这是 PB 提供的 Window ActiveX 组件,是宿主环境识别 ActiveX 组件的关键;
- ② ActiveX 组件进行数据库连接所需要的数据接口,数据接口的设置与普通应用程序的设置完全一样。
- ③ 提供 PB 编译程序运行所必须的所有动态连接库 (.DLL),等等。

参考文献

- 1 Randall A. Tamura, Lotus Notes和Domino Server 4.6技术大全,机械工业出版社,1998.10.
- 2 Joe Pescatello, Integrating Domino with ERP Systems, <http://www.lotus.com/developers/devbase.nsf/articles/doc2001011200>.
- 3 Sybase, How to Display a Window ActiveX Control in a Browser, <http://my.sybase.com/detail?id=47765>.