

客户/服务器图形用户界面 设计与开发指南

*Using PowerBuilder,
SQLWindows, Visual Basic,
PARTS Workbench*

Jonathan S. Sayles, Steve Karlen,
Peter Molchan, and Gary Bilodeau

希望

学苑出版社

网络与通信系列丛书(二)

*GUI-BASED DESIGN AND DEVELOPMENT
FOR CLIENT/SERVER APPLICATIONS*

**客户/服务器图形用户界面
设计与开发指南**

Jonathan S. Sayles 著

Peter Molchan

Steve Karlen

Gary Bilodeau

刘 畅 许 平 译

林 钢 孙 勇 审校

学苑出版社

(京)新登字 151 号

内 容 提 要

本书详细介绍了客户/服务器应用中的 GUI 设计与开发技术,包括客户/服务器结构、面向对象技术、GUI 技术、C/S-OO-GUI 技术及 C/S-OO-GUI 开发平台,以及 Microsoft Windows 下的应用程序开发等技术。书中提供了一个商业应用的样本程序,并着重介绍了四种主要开发工具:PowerBuilder 3.0, Digitalk, Microsoft Visual Basic 3.0 以及 SQLWindows。本书内容翔实,适合软件开发人员及广大计算机工作者使用。

需要本书的读者,请直接与北京海淀 8721 信箱书刊部联系,邮政编码 100080,电话 2562329。

版 权 声 明

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

本书英文版名为《GUI-BASED DESIGN AND DEVELOPMENT FOR CLIENT/SERVER APPLICATIONS》,由 John Wiley & Sons 公司出版,版权归 John Wiley & Sons 公司所有。本书中文版由 John Wiley & Sons 公司授权出版,未经出版者书面许可,本书的任何部分均不得以任何形式或任何手段复制或传播。

网络与通信系列丛书(二)

客户/服务器图形用户界面设计与开发指南

著 者:Jonathan S. Sayles 等

译 者:刘 畅 许 平

审 校:林 钢 孙 勇

责任编辑:甄国宪

出版发行:学苑出版社 邮政编码:100036

社 址:北京市海淀区万寿路西街 11 号

印 刷:兰空印刷厂

开 本:787×1092 1/16

印 张:14.875 字 数:344 千字

印 数:1~5000 册

版 次:1994 年 10 月北京第 1 版第 1 次

ISBN 7-5077-0973-6/TP·32

本册定价:19.80 元

学苑版图书印、装错误可随时退换

目 录

第一章 C/S-OO-GUI 技术及 C/S-OO-GUI 开发平台	1
1.1 过去的方式	1
1.2 C 或者不是 C	2
1.3 C/S-OO-GUI 概要	3
1.4 面向企业客户/服务器结构	5
1.5 GUI: 技术领域中的天鹅	6
1.6 面向对象	8
1.7 四种产品概述	9
第二章 客户/服务器技术概述	13
2.1 客户/服务器	13
2.2 通信——把所有方面连接在一起的纽带	18
2.3 客户/服务器——它为何物	19
2.4 客户跟踪系统	25
2.5 这些意味着什么	28
第三章 GUI 和 Microsoft Windows 应用程序开发简介	31
3.1 什么是 Windows 技术——为什么采用 Windows	31
3.2 窗口的一般特性	35
3.3 图形界面的组成	38
3.4 窗口类型	41
3.5 多重文档界面(MDI)框	43
第四章 面向对象入门	48
4.1 传统系统设计	48
4.2 面向对象程序设计(OOP)	49
4.3 为什么面向对象程序设计是有意义的	49
4.4 在样本应用程序中把所有方面结合在一起	54
4.5 应用程序界面	58
第五章 商业应用例子	62
5.1 应用程序界面	62
第六章 用 PowerBuilder 3.0 构造样本应用程序	66
6.1 历史	66
6.2 安装 PowerBuilder	66
6.3 PowerBuider 的组成	67
6.4 什么是 PowerBuilder 应用程序	69
6.5 利用 PowerBuilder 创建数据库/表	70

6.6	WATCOM SQL DBMS	71
6.7	数据库 Painter	71
6.8	使用数据库 Painter	73
6.9	创建数据库.....	74
6.10	生成一个简单的 PowerBuilder 应用程序	76
6.11	正本和函数	87
6.12	检验点	93
6.13	w_main 窗口控制中的剩余正本	95
6.14	菜单回顾	97
6.15	PowerBuilder 数据窗口控制	102
6.16	什么是数据窗口.....	102
6.17	PowerBuilder 透视	109
第七章	Visual Basic 3.0	110
7.1	数据库存取和网络研究	110
7.2	Microsoft SQLServer——数据库的建立	115
7.3	SQLServer 下的数据库管理	115
7.4	Visual Data 下数据库对象的建立	116
7.5	Visual Basic 开发综述	117
7.6	Visual Basic 工作台	122
7.7	编码标准	128
7.8	从何开始	129
7.9	检查点	135
7.10	Visual Basic 编码	136
7.11	建立屏幕流程.....	137
7.12	数据库存取.....	141
7.13	Visual Basic 的其他功能	149
7.14	这一切意味着什么.....	149
第八章	利用 SQLWindows 开发客户/服务器 GUI	151
8.1	安装	152
8.2	数据库	155
8.3	SQLTalk	160
8.4	SQLWindows 的开发环境	161
8.5	创建 SQLWindows 应用	171
8.6	开发 Employee Update 窗口	175
8.7	创建可执行文件	190
8.8	SQLWindows 概要	191
第九章	利用 PARTS Workbench 开发 C/S-OO-GUI	192
9.1	要求	192
9.2	OS/2 的安装.....	192

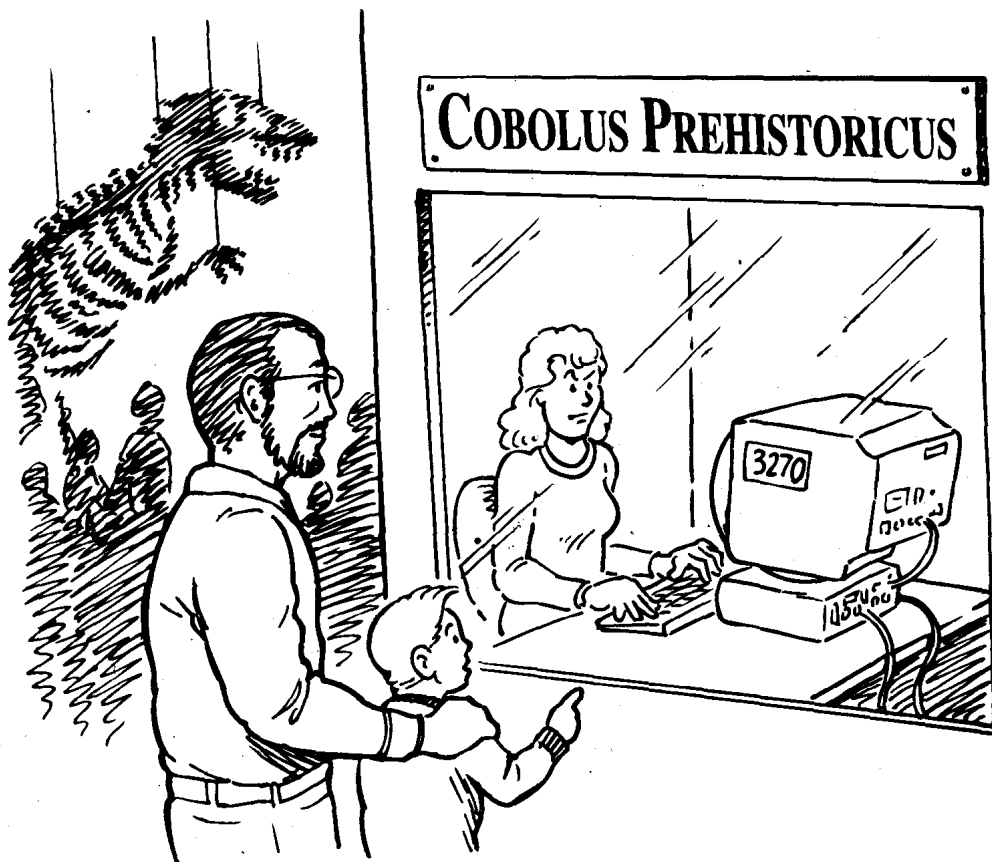
9.3	启动 PARTS Workbench	194
9.4	指导	194
9.5	退出 PARTS Workbench	195
9.6	PARTS Workbench 概述	195
9.7	Parts 基础	200
9.8	样本商业应用	209
9.9	包装应用	230

第一章 C/S-OO-GUI 技术及 C/S-OO-GUI 开发平台

1.1 过去的方式

在 MIS 计算的过去的 30 年中(60 年代、70 年代和 80 年代),应用系统专家大概熟悉两、三种,也可能是四种技术。一般来说,如果你掌握了 COBOL 和 JCL 以及 IMS 或 CICS,你就是公司里的专家,有一个较光明的职业前景,并且你的工作也比较保险。如果你知道 ALC (在 60 年代被称为汇编语言编码,在 80 年代被称为 BAL ——基本汇编语言),你就是一个技术专家。

几十年来,这种有限的技术基础,虽然只包括了两到四种主机系统,却对于在数据处理系统中谋职已足够了。但是,四年来事情变化多大呀!



古代技术博物馆

90 年代对新技术的需求是毋庸置疑的。如果留意一下现在的贸易方式,就会发现,如果

现在不是在使用面向对象或客户/服务器 GUI^① 应用程序(都在 UNIX 上运行),则有可能被放进储藏室(在穿卡机旁边)或在技术博物馆中作为古代计算技术的一部分被展览(“看那里,她是 COBOL 程序员。你看她多紧张。她正在五千条语句中跟踪 GO TO 分支。什么是 GO TO? 不管给你的计算机教师付多少报酬,他都不会讲的”)。

1.1.1 “我的死亡报告已被极大地夸大了”——W. C. Fields

实际上,根据本篇文章,COBOL、JCL、IMS 同 CICS 不是为将来工作预备的,主机系统是丰富的并且依然有足够的工作留给那些记得如何拼写 PERFORM THRU 及追踪寄存器赋值的人们(包括主要的维护工作——记住,这就是使博物馆中的 COBOL 程序员生出皱纹和鱼尾纹的原因)。所有好的工作好在这里被定义为:开发,要求很少或不需要随时响应,好的回报(也许它应放在首位)。我将把所有任务留给你来完成,并且由于需要——它们还需要 SQL 服务器和 PowerBuilder 开发人员)使用面向对象的方法,客户/服务器,GUI 技术来完成。设想把 MIS-Nova 从 MIS-Antiqua 中分离出来将在以后五年变得更明显。

个人计算机和局域网能否取代主机(实际上为各公司节约了资金)不再是理论上争议的问题,新的技术已解决了 MIS 的核心问题,因此,“我们将进入客户/服务器”已不再是一个问题,而仅仅是将什么时候进入客户/服务器,这周还是下周。根据贸易杂志的调查及对如 Forrester 公司, Gartner 集团的独立研究,如果你在 Fortune 1000 公司工作的话,那么对于后一问题的回答是:“去年”。

“好吧,”你说“我将不会在另一时刻跃过这趋势,我已下了决心,我将从今天开始学习 C 语言并成为世界上最好的转变到客户/服务器开发的前 COBOL 程序员,我将以迅雷不及掩耳之势占领 MIS 世界,并很好地把我的工作保持到下一世纪”。不要如此着急,我们永远不会试图阻止 C 语言,计算机语言本身将几乎不会在猎头(headhunter)处给你一个最强烈的简历(术语 headhunter 是指数据处理供职专家——为 MIS 工作吸收新技术开发人员的人们)。C 程序员从每一个社区学院,技术学院以及其他一般语言学院冒出的速度比拼出 GOTO 语句的速度还要快,你将需要更多。特别是,你需要以理论为后盾——客户/服务器的概念和工具,面向对象方法的概念和工具,GUI 概念和工具——并且你需要学习一种或更多种的 GUI 客户/服务器 4GL 如 PowerBuilder (Powersoft 公司)、Visual Basic (Microsoft 公司)、SQL Windows (Gupta 公司)或是 PARTS Workbench (Digitalk 公司)。

1.2 C 或者不是 C

3GL 不再适于试图编码的 MIS 世界,因为学生中的某一人提出了管理的要求:“更优、更廉、更快”(另一个学生告诉我们,该词的含义是“开发系统更优、更廉、更快”实际上是“开发系统更廉、更廉、更廉”)。COBOL 是 3GL(你也可把 COBOL 看作是“浏览文库后得出的汇编语言”),C 也是一种 3GL,适用范围广、功能强,非英语化。以美国关于制造更优、更廉、更快商业应用程序的观点(特别是可维护的更优,更廉,更快)C 不比 COBOL 优越,能做到更优、更廉、更快的是客户/服务器、面向对象 4GL 工作平台。

^① GUI——图形对象接口,是一种通用图标/基于彩色窗口/鼠标驱动的用户界面展示风格,包括在 Microsoft Windows, OS/2 和 Motif 这样的产品中。

4GL 工作平台已经存在几十年了,传统的以字符为基础的 3270 大型机 4GL 如 APS (InterSolv)及 CA-Telon(计算机协会)还遗留着 80 年代计算机辅助系统工程的方法,并且代表了一种用宏类型语言开发系统更优、更廉、更快的可证明方法,集成了调试工具,具有数据存取绘图能力。传统的 4GL 工作平台已经大量产生了一组“新潮 4GL 工作平台”,如 Power Builder、Visual Basic、SQLWindows 及 PARTS Workbench^①这些产品被设计用来允许开发人员——在一些情况下为高效的终端用户——来构造 OS/2 或基于 Microsoft Windows 的图形客户/服务器应用程序。

这些新潮(客户/服务器,面向对象)4GL 工作平台不仅仅是非过程程序设计语言。新潮 4GL 工作平台包括面向图形的、智能化和全集成开发工具集(集成了在过去十年中,在硬件和软件技术中所取得的大多数进步),它们基于个人计算机和局域网开发产品,特征是适应性强,功能强,类英语,非过程的语言,具有宏和函数库、集成工具集、集成调试工具等等。新潮 4GL 工作平台为在各种不同操作系统及大多数 PC/LAN 硬件平台上运行的 GUI、客户/服务器及面向对象 PC 应用程序的快速及简便的开发工作(更优、更廉、更快)而被设计并加以优化。作为最后一点,应明白即使集中所有注意力,这些从出版物中得到的产品,包括 4GL 工作平台是技术 S 曲线的底部,这一点是很重要的。

S 曲线代表了关于新技术的趋势方向的当代经济理论的一个比喻。在 S 曲线底部,技术发展来得快而便宜。而在 S 曲线顶部,同等尺度的技术进步来得慢而昂贵。如果他们全部到来的话,大型计算机及实际上所有相关的大型机技术——DB2 可能除外——是在 S 曲线的顶部,厂家已对该技术很少投资,或者有时根本不投资。

个人计算机、局域网同所有相关个人计算机技术(也就是,C/S-OO-GUI)在 S 曲线的底部,对个人计算机而言,技术功能同价格价值比正飞速沿 S 曲线上升。硬件价格(用处理能力衡量)则像一月份温度计中的水银柱一样下降。当 IBM 公司依然想成为针对传统应用程序的大型机客户/服务器的主宰时,当你读到这篇文章时不要打赌,大企业中先前不能接触的大规模应用程序不会考虑客户/服务器技术。

沿着 CS-OO-GUI 既定不变路径是不可避免的数据处理。我们认为预测一种新潮 4GL 工作台的使用,如 PowerBuilder、Visual Basic、SQLWindows、PARTS Workbench 或其他类似产品,是一种 C/S-OO-GUI 应用开发既成事实的看法并不过分。换句话说,这些产品,用我们的观点,代表了一种面向对象 GUI 客户/服务器应用开发的趋势并且将渗透到下一代计算机技术中。

因此如果你是一个想要调整自己的能力使之适应于市场的开发人员的话,我们着重推荐你应学会这本书讨论的产品的一个版本,这些产品正统治现在的 C/S-OO-GUI 市场,我们觉得这些产品将统治下一十年的市场,其最好的解释是,这些产品是超级工具。

1.3 C/S-OO-GUI 概要

并不是厂商没有任何困难告诉你这些。实际上,开始介绍各种技术时,我们想,学会当今 C/S-OO-GUI 技术的行话在今后是适合的(贸易杂志上到处存在的行话,我们认为所有读者

^① 我们应当注意到厂商(Powersoft、Microsoft、Gupta 及 Digitalk)已给出来自它们的市场文献的术语 4GL 的所有参考文献。它们描述它们的产品为面向对象,客户/服务器应用开发工具集

应相当熟悉),看一下下面任一读者都可能在任一数据处理数据库杂志上看到的广告:

你想寻找一个客户/服务器、GUI、面向对象方案去解决你的开发问题吗?欢迎到高技术世界,欢迎选购 Acme GUI——客户/服务器工作台,该产品基于 Windows 非过程开发工具,产生可重复使用应用成分的方法(通过面向对象),同一些基于 PC/LAN 相关的 DBMS 紧密集成。

你对段中的许多行话将对多数读者很重要感到疑惑吗?或者意味着一个已经过去的模糊概念?你能想象一个比 MIS 和软件开发更不适合的职业吗?让我们用广告行话开始并看我们能否从清晰、易于了解及有用的定义开始。

1.3.1 客户/服务器计算

客户/服务器计算,代表了 90 年代^①乃至以后的新 DBMS Holy Grail。没有比命名产品为客户/服务器更快的方法去卖该产品的了。不幸的是,客户/服务器技术的绝对商业成功已经使它作为新技术而不可抗拒,来自于每一行业的产品已经采用客户/服务器的概念。

不像关系数据库技术(Codd 博士对 ACM 的 12 条总结),面向对象(Tom Atwood 面向对象声明)及 GUI 技术(Charles Petzold 的 Microsoft GUI 设计的论文),没有一个被 MIS 客户/服务器厂家用作对客户/服务器定义的单一源定义,或客户/服务器十条建议。即使有这种形式的欠缺,让我们用最基本和最广泛的客户/服务器允许特征来开始我们自己的定义,并且包括客户/服务器的更专业化、更紧密及最新的定义。

- 客户是请求处理的任一程序。
- 服务器是传递处理的任一程序。
- 客户/服务器计算是任一应用程序系统,分离自由程序之间互相请求并传递处理。

如果把该定义用到它的逻辑结论,则能得到现有数据处理应用程序客户/服务器计算的相当好的一个百分率,例如,一个传统调用,使用应用程序界面:

- 来自“Program A”
 - …过程逻辑
 - 调用“programB”,使用字段 1,字段 2,字段 3
- 来自“programB”
 - …过程逻辑用于字段 1、字段 2、字段 3 返回

大多数联机主机应用程序系统,在该定义下都有成为客户/服务器的资格(独立批处理 COBOL 模块除外),其他有效客户/服务器计算(通过这种最基本的定义)的例子包括数据库查询与报告软件包如 QMF(IBM)、Paradox(Borland),FoxBase(Microsoft),dbExpress(计算机概念联合公司),以及各种不同类型的服务器产品(文件、传真以及数据库服务器如 Sybase/SQLServer,Oracle,Informix,XDB,SQLBase,DB2/2 等等)。

“这很好”,你说,“但是如果考虑本企业中的已运行二十五年的 COBOL/CICS 主机应用程序作为客户/服务器系统,虽然我可能按你的定义在技术上是精确的,我将为将来的 Sybase/PowerBuilder 类除去某些潜在的产品”。当代数据处理/MIS 文化(由 Microsoft、IBM 及其他公司为代表),已经配制了一个松散的——但是被广泛接受——客户/服务器技术的

^① Forrester Research 公司估计全部客户/服务器开发工具市场在全球范围内将在 1996 年达到 4 亿美元。

伪定义,它与已发表的理论或数学模型相比更有一点专门化,并且——不是太令人惊奇——同某些商业成功软件相关联。

换句话说,还依然没有由所有参与基于客户/服务器工具开发与产品发布的团体同意的客户/服务器的正式定义,然而,有一些传统上认为是客户/服务器产品的商业软件共享的特点。

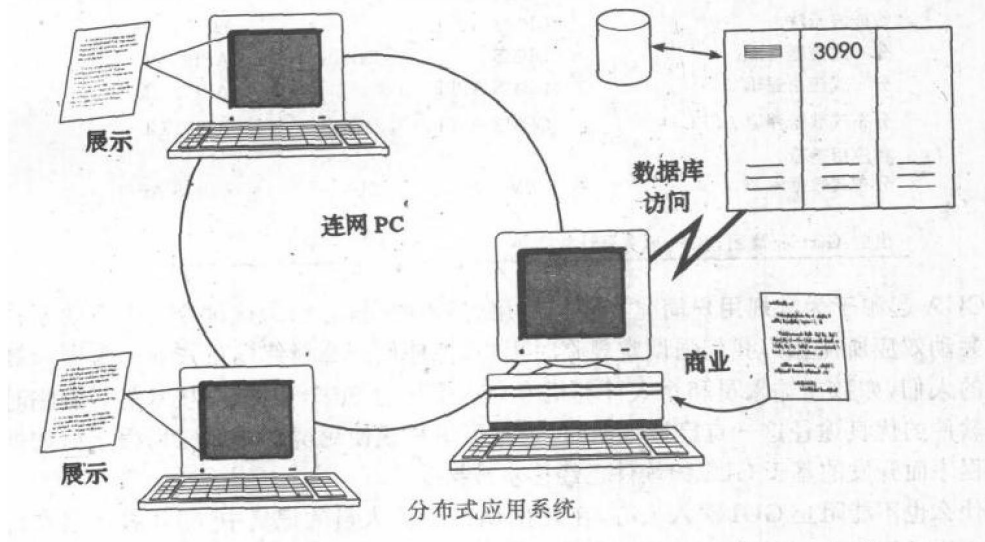
具有这些通用特征的客户/服务器计算的好的定义可以在《客户/服务器计算》中找到,该书由 Alex Berson 写成, McGraw-Hill 发行。

分布式计算,在那里至少应用程序系统的一部分驻留于 PC 机或智能工作站...

虽然, Berson 的定义看起来比我们开始的通用定义稍微明确一点儿,它实际上在覆盖被作为客户/服务器接受的范围上起到了很好的作用,但没有讨论中的技术那样通用。理解 Berson 定义的关键术语是分布式。

1.3.2 能分布什么

应用程序系统能被分为三个独立的功能组件:用户界面——表示屏幕、画面、报告等等,商业用户进行操作的应用程序的所有部分;商业逻辑——表示正式的策略和过程,由软件中的逻辑或代码自动操作;数据库存取——来自于一个或更多的电子文件系统信息的存储与检索,客户/服务器技术提供了在各种不同硬件/软件平台,如 PC 机,工作站,局域网乃至大型计算机上分配这三种应用程序系统组件的方法,这种分配方式允许应用程序设计者同开发人员为数据更好存取、系统应答、执行及同其他应用程序或商业软件集成来优化系统。



1.4 面向企业客户/服务器结构

结构(在多数 MIS 讨论中)是指支持应用开发和传递的主机或硬件、软件工具。现在,大多数计算环境都具有相当复杂的构造,包括大型机、服务器(传统定义的局域网)、个人计算机或独立工作站。

为了试图描述分布用户界面(UI),商业逻辑(APPL)和数据访问(DBMS)的多数有用方

法,对于包括大型机、服务器和 PC 机的企业,Gartner 集团发明了现在由 Fortune 1000 采用的各类型客户/服务器结构的分类方法(表 1.1)。

所正如看到的,有六种不同的应用系统组成分布的结合,在第二章客户/服务器环境中的用户图形界面下,我们将作更深入的讨论(大多数关于客户/服务器)。

1.4.1 这就是有关客户/服务器的所有内容吗

是的,这就是这一章的意图。然而,主要的客户/服务器厂家(Microsoft, Powersoft, Sybase, Gupta 等公司)正通过技术与商业手段改善他们的产品,并且热衷于结合其他先进的当代技术到客户/服务器中,这些先进技术包括图形用户界面(GUI)、面向对象方法以及关系 DBMS。

GUI 或图形用户界面软件是用于最终用户表现传递的现代基准,不像客户/服务器,GUI 有标准的定义,该定义在 80 年代末 IBM 公司在他们的 CUA(通用用户存取)专用声明,并在文档中作了详细解释,该 CUA 是系统应用界面的指南,IBM 公司的 SAA(系统应用结构)的一部分,采用图形图标、窗口、下拉菜单和点设备(鼠标)技术,代替了 PF-key(程序功能键),基于字符的,非标准化界面及传统的传递基于 3270 大型机应用的方法。

表 1.1 应用组成分布方法

客户/服务器结构	主机系统	服务器	个人计算机
单机	UI APPL		
	DBMS		
智能表示法	DBMS APPL		UI
分布式数据存取	DBMS	DBMS	APPL UI
分布式应用逻辑	DBMS APPL		APPL UI
分布式数据存取 APPL	DBMS APPL	DBMS	APPL UI
和应用逻辑			
分布式数据库	DBMS	DBMS	DBMS APPL
			UI

出处:Gartner 集团,客户/服务器计算摘要

CUA 起源于为常规用户简化与规范应用(不幸的是,它已为这种令人振奋的新技术引起的轰动效应所掩盖),我们强调常规临时用户,把他们同高级终端用户或经常同计算机打交道的人们,如数据输入员和个人计算机专业人员区分开来,当 GUI 技术是开发临时用户使用软件的优良途径这一点已被证明时,为高级用户或使用屏幕操纵应用程序如数据收集应用程序而开发的基于 GUI 的软件已经逐步被接受。

什么也不能阻止 GUI 深入人心,在现在 MIS 专业人员的记忆中,GUI 技术已在传送系统。为终端用户能达到较短学习曲线的系统,有更先进功能的系统,已被证明能在用来降低昂贵的数据输入错误费用的系统上取得了成功。相当简单,GUI 应用被作为能提供较优用户界面而接受。但是一个较好的用户界面(像 GUI)价格一般较高。

1.5 GUI:技术领域中的天鹅

如果曾经看见过天鹅在水池中优雅地游泳,你可能认为它们做到这一点毫不吃力。然后

经过距离的观察后,你会发现天鹅的腿像小小的自动活塞一样移动,用很大的努力以保持其轻松、优雅。

GUI 技术传递与开发其表面上简单而实际上是艰巨而复杂的。同上面所述是一个道理,当一个 GUI 应用的最终产品已经完成,要达到使用轻松并且具有简单、吸引人、效率高的用户界面(图 1.1),为建立 GUI 而进行设计编程付出的努力是难以言传的。

事实上,Gartner 集团,在 1992 年七月关于客户/服务器的报告中称,在很多情况下,仅仅从客户/服务器技术中得到的可证实的节省主要是降低了数据输入错误,原因是 GUI 前端处理机在应用中的结论。

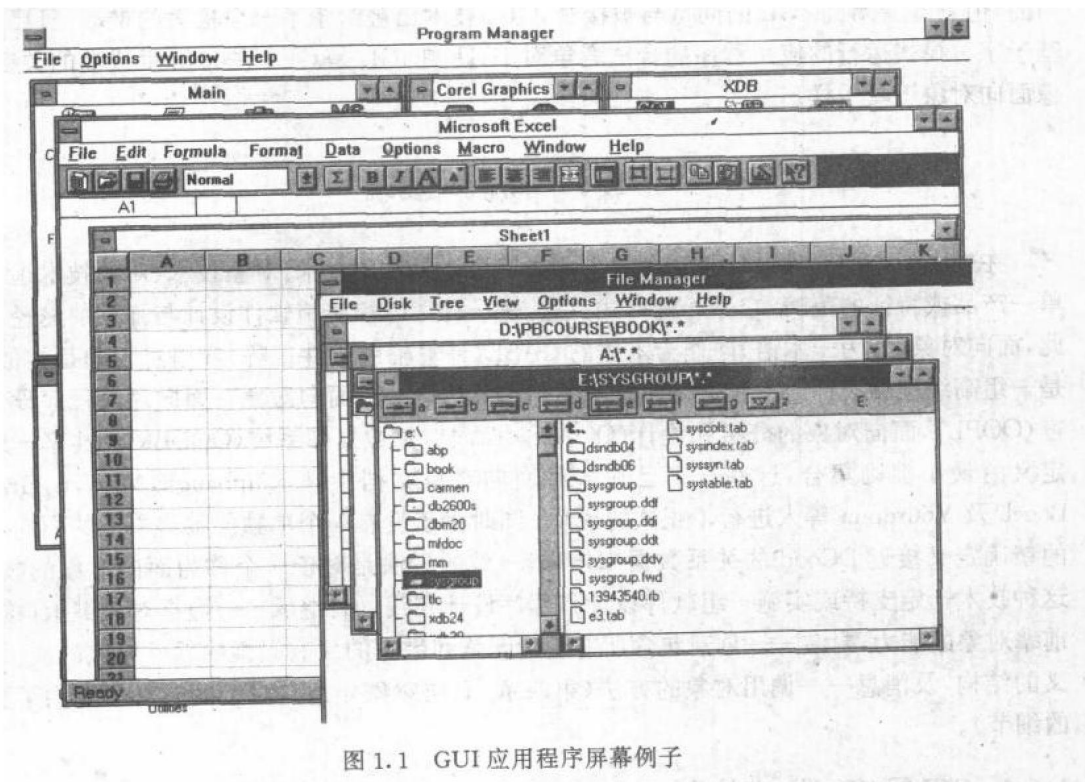


图 1.1 GUI 应用程序屏幕例子

1.5.1 Windows 和 OS/2 API

直到 4GL Windows 开发工具如 SQLWindows、PowerBuilder 及 Visual Basic 问世后,Microsoft Windows GUI 及 Presentation Manager 软件的程序设计就用 3GL 和面向对象编程语言如 C、C++ 和 Smalltalk 来完成,其中 Smalltalk 调用 Windows 和 OS/2 API(应用程序接口)中的函数和宏。

需要编写甚至很简单的 Windows 应用程序的代码。该应用程序使用本机 C/Windows API 程序编制,是事件驱动的^①通常很复杂。对于大型机开发人员——或者在这种情况下,任一开发人员,有经验的 C 程序员除外——编写是困难的(C/Windows API 程序编制对于主机 COBOL 人员包括一特殊陡峭的曲线)。然而,随着新潮 4GL 工作台如 PowerBuilder、

^① 事件驱动指的是一种过程程序设计和应用设计。任何外部事件(鼠标单击、鼠标移动、跳行等等)都会应在应用系统中引起事件发生。这些事件或者被排队,或者被代码包立即处理,这些代码专门为特定事件编写。

Visual Basic、SQLWindows 及 PARTS Workbench 的问世,可用这些产品生成 C/Windows GUI 代码并允许开发人员用建立基于字符应用的相等同的努力来建立复杂的 GUI 应用程序(越过初始学习曲线),这不再是一个问题。

1.5.2 新潮 4GL

GUI 技术同我们的新潮 4GL 之间的协作不是偶然的。因为用 Windows/API-C 程序设计来开发传统应用的困难在于没有一个简化方法来快速而廉价地产生传统 GUI 应用。GUI 技术(暂时)有成为装饰平台的危险,仅仅对有 C/系统程序设计背景的软件开发人员才是可用的,但是随着新潮 4GL 的问世与被接受,GUI 技术已被给予了一个极大的推动,且已成为基于字符模式运行的应用程序的合法竞争对手,新潮 4GL 是 GUI 在 MIS 上成功的关键,就像面向对象工具一样。

1.6 面向对象

在过去十年中软件设计与开发的主要成就之一是对象技术。对象技术(对象技术远超出单一产品或特征的范畴,它是一种形式方法,是一种考虑和实施软件设计与开发的途径。因此,面向对象(作为一术语)同作为术语的 CASE(计算辅助软件工程)或信息工程是类似的,是一组高级程序设计语言和数据库共享通用指导法则——面向对象。面向对象程序设计语言(OOPL),面向对象的分析和设计(OOAD)、面向对象数据库系统(OODBMS)共享一组被定义的最小准则集合,这些规则已在各种面向对象文档中被 Rumbaugh, Martyn, Grady/Booch 及 Yourdoun 等人进行了正式的说明。同时由于没有一个单独的教科书上的面向对象的精确定义接近 DCodd 的关系数据库定理,一致的看法是对于一个称为面向对象的技术,这种技术一定支持或实施一组软件设计和程序设计实践,如:继承——数据对象继承(接收)前辈对象的能力,封装——创建包含所有逻辑元素和数据的所有元素的软件对象的能力,定义的结构,及消息——调用对象的方法(第四章“面向对象介绍”,对于面向对象提供了更多的细节)。

1.6.1 GUI 同 OO:进一步协作

随着软件方向和趋势的改变,DP 世界好像在真空状态什么也没发生。过去十年中,面向对象同 GUI 已经成为软件开发世界的趋势,并且(不知你是否知道),它们共享重要的特征。

作为结果,窗口(GUI)应用开发是最好的被访问语言,该语言涉及图形对象而不是仅在一控制块中的字段和记录,窗口宏和控制存在于类库(另一 OO 构造)中,同时使用消息来同另外一个或终端用户通信。GUI 程序设计的许多概念和术语在 OO 技术中有他们自己的根,因此,协作自然存在于 OO 与 GUI 之间。你可在该书后面章节看到,重要 C/S-OO-GUI 厂家的大多数已经将一个由小到大面向对象的等级包含在它们相应产品的应用开发环境中。

1.7 四种产品概述

到了讨论我们将用来建立样本商业应用的 4GL Workbench 产品的时候了。这一章仅对每一产品是什么及对每公司的背景作一介绍。

下面的章节将用实际开发过程来描述产品的使用。

1.7.1 PowerBuilder 3.0

PowerBuilder 是一面向对象,图形化,客户/服务器开发工具集,用于建立并传递商业 Microsoft Windows 应用。PowerBuilder 1.0 在 1991 年 6 月开始提供,很快作为高级图形开发平台被开发人员广泛接受。PowerBuilder 2.0 1992 年 7 月开始提供,提供了大量面向对象扩展功能,以及改善了的 SQL 开发功能。PowerBuilder 3.0 1993 年 9 月开始提供,对面向对象进行了进一步扩展,同时提供更高级 SQL 功能,包括对 DBMS 单击开发界面的扩展及更复杂的图形功能。

PowerBuilder 的目标是大型 Fortune 1000 公司和政府机构,它们注意平衡收益和战略商业机遇的关系,该机遇同基于客户/服务器的信息系统相关。Powersoft 是一家公共控股的贸易公司,它称 1993 年 6 月以来,2600 多个组织正使用 PowerBuilder,超过 13400 的合法拷贝已被售出,为了管理这种极其快速的增长,Powersoft 公司已经赞成建立几个支持机构。

教育伙伴 以 Novell 公司和它们的资格程序为样板,Powersoft 已经建立了一个分级教育伙伴程序,该程序提供了课程教授、独立训练以及 PowerBuilder 的查询证明书。课程是在 Burlington、Mass 及遍布美国的许多其他城市的公司总部提供的,大约有超过三十种 Powersoft 教育伙伴,该程序也被提供到国际上,通过 Powersoft 欧洲总部和 PowerBuilder 的训练代理进行。

支持、咨询同其他服务 Powersoft 也提供给顾客广泛的各种支持计划,包括咨询服务、电子公告牌系统,FAXBACK 系统(提供技术支持),以及 CompuServe 论坛功能。Powersoft 保留十个现场办公室在美国和加拿大来销售和支持它的产品。Powersoft 也通过七十多个授权代理商的关系销售 PowerBuilder 及附加服务。欧洲的运行总部设在英格兰的 Maidenhead。Powersoft 在欧洲、拉丁美洲、太平洋附近地区提供了产品维护、教授、资格和咨询服务。

支持的平台和 DBMS 支持的开发平台,运行平台(PowerBuilder 生成的应用在其上可运行),及 DBMS 在表 1.2 中显示,应当注意到这个表每季都增加,并且如果需支持的技术未在该表中出现,请同该公司联系以进行更新。

表 1.2 PowerBuilder 平台和产品支持

开发平台	执行平台	DMBS
Microsoft Windows 3.1 OS/2 2.1	Microsoft Windows 3.1 OS/2 2.1	ALLBASE/SQL Database Manager 和 DB2/2 Informix Micro Decisionware Database Gateway to DB2 ODBC Oracle SQL Server GUPTA SQLBase Sybase Watcom SQL XDB

1.7.2 Digitalk

于 1983 年由 Jim Anderson、George Bosworth 及 Barbara Noparstak 建立的 Digitalk 公司,于 1985 年成为提供商业可用对象技术的第一家公司,该技术是在 PC 机上提供一个名叫 Methods 的产品,在 1986 年被称为 Smalltalk/V, Digitalk 有 Smalltalk/V 在 DOS、Macintosh、Windows、OS/2 的系列版本及 1993 年在 IBM RS/6000 平台上的 AIX 版本。

Digitalk 的成熟及它的面向对象的 Smalltalk 对开发人员很大的便利,因为 Smalltalk/V 在该领域的使用比其他 PC 对象技术产品时间长,它有丰富的类库集合——大约是大多数通常 C++ 编译器的两倍,这些类的宽度同深度使应用开发快而简便。组件的重建与测试确保了 Smalltalk/V 目标类是可靠的。今天,世界上超过 100,000 的人正在使用 Smalltalk/V。这个已打下的基础成为接收 Digitalk 新技术的坚实基础。

在 1990 年 10 月, Digitalk 同 IBM 宣布了一个主要联盟:即 Digitalk 的 Smalltalk/V for Windows 和 OS/2 及它的新 PARTS 产品系列的开发特许协议。该协议在 1991 年 9 月被戏剧性的扩充,当时 IBM 指定 Digitalk 为它的 AD/Cycle 国际联盟,即由于它们有共同顾客而被 IBM 认可的由八家公司组成的特别集团。

1992 年 10 月, Digitalk 推出 PATRS 工作平台,第一个基于允许从预建立软件组件进行快速、可视应用程序构造的部件组装和重用工具集技术的系列产品,这允许软件开发人员通过一次次重用这些组件来平衡组件的创建。PARTS 技术基于 Smalltalk/V 对象技术,1993 年 1 月, Digitalk 为 OS/2 推出 PARTS Workbench 的 2.0 版本,它具有新的 32 位结构,使得它比 1.0 版本更小更快速。那一年也推出了 PARTS COBOL Wrapper,是 PARTS Workbench 的伙伴产品,允许与 COBOL 代码集成并支持 Micro Focus COBOL。PARTS Workbench 的另一伙伴产品, PARTS 关系数据库接口允许同关系数据库集成。它推出了第一个系列数据库界面部件,支持 IBM OS/2 Extended Services Database Manager。PARTS Workbench 的特征在表 1.3 中列出。

表 1.3 PARTS Workbench, 平台及产品支持

开发平台	运行平台	DBMS
OS/2 2.0, 2.1	OS/2 2.0, 2.1	SQL Server
Microsoft Windows	Microsoft Windows	Oracle
	[即将发行]	IBM 扩展服务
		Database Manager
		DB2 和其他

Digitalk 专业服务 Digitalk 专业服务是关于 Smalltalk/V 设计方法, 面向对象技术及管理理论, 技术评价与转化及项目定义的信息源, 提供了咨询同教育服务。

直接联系 直接联系是 Digitalk 联机支持和信息服务。直接连接是到 Digitalk 技术支持的直接链路, 能经由三十个国家的七百五十多座城市中的本地电话号码使用, 直接边接服务提供 Digitalk 开发环境的最新信息、访问 Digitalk 技术支援人员的途径, 同其他 Digitalk 产品用户及扩展数据库进行技术信息(包括代码、共用请求问题及问题报告)交流的途径。

1.7.3 Microsoft Visual Basic 3.0

Visual Basic 是 Microsoft 开发出的上百种产品之一。如果你不知道它们是什么, 让我首先欢迎你来到地球上。Microsoft 成立于 1975 年, 自从 IBM PC 机于 1980 年问世后它已家喻户晓。Microsoft 开发了用于个人计算机的操作系统(MS-DOS), 奠定了世界上最大软件销售商的现有地位。实际上, 初始的 IBM PC 是随 MS-DOS 1.0 及 Microsoft Basic 解释程序到来的。

尽管 Basic 已经问世许多年了, Microsoft 通过使它能为大众使用后才使该语言备受瞩目。该语言用来作为学习程序设计基础的教育工具来使用——这样看起来 Basic 将不会成为主程序程序设计语言。

Microsoft 明显感觉到不同。Microsoft 通过推出几种 Basic 编译器使 Basic 成为 PC 机应用程序开发的可行语言。Basic 开始作为一种功能强, 易于学习、易于使用的 PC 程序设计语言而被接受。1986 年, Microsoft 推出 Windows, 随着 Windows 的成熟, 开发人员就开始要求一个创建 Windows 应用程序的开发环境和 4GL 语言。

今天, Microsoft Visual Basic 3.0 版本是 Windows 的客户/服务器应用程序开发的领先工具之一, 它的成功已造成几种事实: 产品、组织和支援。

作为一种高级软件包, VB 的很大一部分的成功归功于由其他厂家开发的扩充产品, 这些包括对数据库 API 中间工具的任一附加控制, 它给了 Visual Basic 极大的厂家支援的益处。Microsoft ODBC 数据库连通性使 VB 能访问一些最流行数据库格式, 允许重要的企业连通性。

作为一个组织的 Microsoft 也对产品的成功有贡献。销售、支援、微软大学课程, Microsoft Consultan Relations Program、Microsoft 开发者网络, 及这些它们在 CompuServe 上维护的论坛都是开发人员可依赖的重要服务。

1.7.4 SQLWindows

SQLWindows 是由 Gupta 公司开发的客户/服务器开发工具, 并在几年前就已经成为在