

桥梁 CAD 软件开发中的新技术

李 新

(交通部公路科学研究所 北京 100088)

摘要 本文介绍了公路桥梁 CAD 软件开发技术, 所述内容在作者正在开发的一个桥型布置软件中均得到了应用。
关键词 桥梁 CAD ARX MFC OLE

New Techniques in Bridge CAD Software Development

Li Xin

(Research Institute of Highway, Beijing)

Abstract In this paper, some new techniques raised in recent years have been introduced. All the techniques introduced would be useful to the developers of highway bridge CAD software and have been used in a bridge layout CAD software which is being developed by the author.
Key words Bridge CAD ARX MFC OLE

0 引言

通过近 10 余年的发展, 公路行业 CAD 技术已具有一定基础, 国内各设计单位结合生产自行开发了不少 CAD 软件。但由于公路桥梁的结构形式繁多, 几何模型各异, 其 CAD 软件的开发难度就相对大一些, 现有的大多数桥梁 CAD 软件都是以二维的施工图的绘制为目标。总的来看, 这些 CAD 系统与其它先进行业相比还较为落后, 主要体现在:

(1) 运行平台落后。现存的绝大多数 CAD 软件都是基于 DOS 平台的, 这与今后作为主流的具有图形界面的 32 位操作系统很不相宜。(2) 用户界面较差。大多数现存的 CAD 软件中, 有的具备有限的交互功能, 有的则根本不具备交互功能, 只是将数据依次地从数据文件中读入, 然后直接生成绘图文件。(3) 代码复用性差。目前的桥梁 CAD 软件大多使用面向过程的程序设计语言, 代码难以被他人继承复用, 而先进的面向对象的程序设计方法应用得较少。

另一方面, 系统软件、开发工具以及微机硬件都在快速地发展着。具有图形用户界面的 32 位操作系

统的推出, 各种可视化的快速开发工具的出现, Pentium 系列微处理器性能的极大增强, 都为工程 CAD 软件的开发提供了极为有利的条件。因此, 桥梁 CAD 软件的开发正面临着很多新兴的软件开发技术。

1 32 位 Windows 为 CAD 软件提供了理想的运行环境

自从廉价、易用、功能强大的 Windows 95 这个 32 位图形界面操作系统推出以来, 越来越多的人将自己的操作环境转移到 Windows 之下, 而逐步抛弃了基于文本界面的 DOS 操作系统。尽管 DOS 还将在今后一段时期内被应用, 但其被 Windows 取代的趋势已是必不可免。无论是将已有的 CAD 软件移植到 Windows 平台之上还是学习在 Windows 平台上的程序设计方法, 对每个 CAD 软件开发人员来说都是极为必要的。

尽管 Windows 环境对 CAD 软件开发人员提出了新的课题, 但它带来的益处却更大, 因为它为工程 CAD 软件提供极为理想的开发与运行环境, 这主要表现在:

(1) 32 位 Windows 为应用程序提供更多的可用系统资源, 特别是内存问题。只要物理存储器允许, 应用程序几乎可以完全不必考虑内存的制约。

(2) 通过在程序中使用 Windows 资源, 可以非常容易地实现风格一致的图形用户界面, 不必象 DOS 程序那样花费大量精力用绘图函数去自己绘制用户界面。

(3) 中文显示方便。无论是中文 Windows 还是西文 Windows 外挂中文平台, 在程序的用户界面中显示中文提示和标注都十分容易, 而且一般不会有任

何问题。

(4) Windows 程序的帮助系统都是统一风格的, 利用 Windows 系统或开发工具提供的帮助编译器, 可以方便地将固定格式的

帮助源文件编译成应用程序帮助系统, 这比 DOS 环境下帮助系统的制作要简便得多。

(5) Windows 程序均是靠消息驱动, 不同的应用程序之间可以进行相互通讯, 交换数据。应用程序间的数据交换可通过系统消息、动态数据交换、OLE 自动化等方式实现。

虽然 Windows 程序运行机制比较复杂, 但利用各种可视化的快速开发工具, Windows 应用程序的开发却十分的简单易学, 有时甚至比开发 DOS 程序还要简单。目前 Windows 应用程序的开发工具有好几种, 最为流行的有 Visusl Basic、Visual C++、Delphi、PowerBuilder 等等, 这些都是可视化的开发环境, 支持面向对象的编程方法, 并具有数据库的连接访问能力, 为快速开发具有良好图形用户界面的 Windows 应用程序提供了强有力的支持。

2 获得 AutoCAD 的图形支持

AutoCAD 是最早进入中国市场的国际著名 CAD 图形系统, 已占领了中国绝大部分 CAD 应用市场。众多的设计单位早已熟悉并习惯了 AutoCAD 作为绘图工具或作为支撑系统进行二次开发。AutoCAD 在 R11 版中就提供 ADS (AutoCAD Development System) 作为其 C 语言二次开发环境, 由于 ADS 可编译成可执行文件, 直接控制硬件, 能够使用 Windows 资源, 因而成为目前普遍使用的编程手段。在 AutoCAD R13 中提供了一种新的应用程序接口实时扩展器 ARX (AutoCAD Runtime eXtension), 它是新一代的基于 C++ 的应用程序接口, 使得开发人员能够生成更快、更便于使用的面向对象的应用程序。ARX 应用程序是作为 AutoCAD 的动态连接库 DLL 而运行

的, 它共享 AutoCAD 的内存地址空间, 并且被 AutoCAD 直接调用执行, 而不是象 ADS 程序那样作为一个独立的进程。靠进程间通讯通过 AutoLISP 解释器来加载和调用。对于需要频繁与 AutoCAD 进行通讯的应用程序来说, ARX 的执行效率要比 ADS 高得多。ARX 把对图形数据库的访问和操作均包装在若干个 C++ 类中, 以类库的形式提供编程接口, 这就使得 90 年代的主流程设计方法——面向对象的程序设计方法可以自然地应用到 CAD 软件开发中来。在 AutoCAD R14 中, 为了强调这一强有力的编程工具所具有的面向对象的基本特性, ARX 被重新命名为 ObjectARX。由于 ARX 是通过类的数据成员和成员函数访问和操作图形数据库, 不必象 ADS 那样借助于结果缓冲区操纵复杂的 DXF 组码, ARX 的编程方法也相对要简单易学。下面分别给出了将改变实体的层属性的 ARX 和 ADS 代码, 我们可以看到完成同样功能的 ARX 代码要比 ADS 代码简短得多, 而且也更容易理解和掌握。

(源程序代码 1)

```
void ARXChangeLayer (const AcDbObjectId& entld, const
char * pNewLayerName)
{
    AcDbEntity *pEntity;
    acdbOpenObject (pEntity, entld, AcDb::kForWrite);
    pEntity->setLayer (pNewLayerName);
    pEntity->close ();
}

void ADSChangeLayer (ads_name entityName, const char
* pNewLayerName)
{
    struct resbuf *pRb, *pTempRb;
    pRb= ads_entget (entityName);
    for (pTempRb= pRb; pTempRb->restype != 8;
        pTempRb= pTempRb->rbnext)
    {;}
    free (pTempRb->resval_rstring);
    pTempRb->resval_rstring = (char *) malloc (strlen
(pNewLayerName) + 1);
    strcpy (pTempRb->resval_rstring, pNewLayerName);
    ads_entmod (pRb);
    ads_retrb (pRb);
    ads_retvoid ();
}
```

在 AutoCAD 中, ARX 类库被设计成可以与 ADS 程序一起协同工作, 这样开发者就可以使用他们最熟悉、最适合的编程工具, 用 ADS 程序实现应用程序

的用户界面，用 ARX 程序实现对图形数据库的操作。

3 Visual C++与 MFC

尽管有多种功能强大的 Windows 程序可视化快速开发工具供 CAD 软件开发选用，然而从开发基于 AutoCAD 的 CAD 系统，特别是编写 ADS/ARX 程序的观点来看，Visual C++却是最为理想的。因为从 R13 版开始，Windows 版本的 AutoCAD 只提供了 Visual C++的 ADS/ARX 连接库，这就使得 Visual C++成为开发基于 AutoCAD 图形系统的 CAD 软件的首选开发语言。还有一个值得考虑的重要原因就是 Visual C++ 4.0 与 32 位 Windows 环境下的 Fortran PowerStation 共同集成于 Developer Studio 开发环境之中，可以实现 Windows 环境下二者的混合编程，这样就使得以前开发的大量用于分析计算的 Fortran 代码可以得到利用。Visual C++ 虽不如 Visual Basic、Delphi 等易学易用，但它全面实现了真正意义上的面向对象（封装性、继承性和多态性）的开发方法，具有高度的兼容性和复用系统资源的灵活性。与 Visual C++包装在一起的 MFC（Microsoft Foundation Class Library），是专门为那些在 Windows 系统中编写 C++程序的软件开发人员而设计的健壮的应用程序框架，它包括 100 多个常用类，其代码经过严格测试在应用程序中可直接使用，极大地简化了 Windows 繁琐的事件处理工作，并可以保证应用程序界面的一致性。使用 MFC 开发 Windows 应用程序比直接使用 SDK 可免去花费更多时间开发程序结构部件，而且亦不必担心 Windows API 的过多细节，使开发周期大大缩短。发展到 5.0 版本，MFC 已具有 120 000 行代码，内容极为丰富，包括应用体系结构类、可视对象类、OLE 类、数据库类等各方面，是软件开发者的重要的可复用资源。

AutoCAD 在其 ADS 库中为我们定义了一个派生于 MFC 的 WinApp 的类 ADS_APP，它可直接作为 MFC 程序的应用程序类来使用，而且它自动管理了 ADS 程序的初始化及与 AutoLISP 间的通讯，使 ADS 程序可以利用 Visual C++的各种便利工具，如 Class Wizard。MFC 中的文档和视图类，对于 ADS 程序来说意义并不是很大，可以不用，因而 AppWizard 生成的代码框架并不非常适合于 ADS 程序，尽管进行若干修改后就可以顺利运行，但较好的方法是自己编写程序框架，不用 MFC 的文档、视图类。ADS-APP 中的初始化实例虚函数 InitInstance（）按以下源程

序代码 2 来重载。

（源程序代码 2）

```

BOOL ADS_APP::InitInstance（）
{
    .....//搜索出 AutoCAD 图形窗口句柄 hwnd_AutoCAD
    if（CWinApp::InitInstance（））
    {
        for（int i=0; i< ARRAY_SIZE（app_funs）; i++）
            if（InsertExternFunc（app_funs[i]）== -1）return
FALSE;

        m_pMainWnd= new CMainWindow（）;
        m_pMainWnd-> Show Window（m_nCmdShow）;
        m_pMainWnd-> UpdateWindow（）;
        return TRUE;
    }
    else
        return FALSE;
}

```

在上述代码中，类 CMainWindow 是程序主窗口类，其余标识符均在 ADS_APP 中已有定义。ADS 程序的用户界面最好是作为 AutoCAD 窗口的子窗口出现，这样当用户使用 AutoCAD 交互作图时，就不会被 AutoCAD 窗口盖住，使用起来就较为方便，所以 ADS 程序中就要首先搜索出 AutoCAD 窗口句柄 hwnd_AutoCAD，然后在类 CMainWindow 的构造函数中加以创建 ADS 程序窗口。

（源程序代码 3）

```

CMainWindow::CMainWindow（）
{
    Create（NULL,
        MyWindowName, //ADS 程序窗口名
        WS_POPUP|WS_VISIBLE|WS_CAPTION|WS_
SYSTEMMENU|WS_OVERLAPPED,
        rectDefault,
        CWnd::FromHandle（hwnd_AutoCAD）,
        MAKEINTRESOURCE（IDR_MYADSMENU）,
        //IDR_MYADSMENU 为 ADS 程序的菜单标
        识号
        0
        NULL）;
}

```

4 OLE 技术的应用

OLE 最初是 Microsoft 公司为了加强复合文档的创建和管理而建立的一套应用程序之间交换数据的技术规范，后来 OLE 的应用从复合文档扩展为范围更广的部件软件开发，发展成为 2.0 版，现已成为跨应

用程序的通用的客户化和集成化的规范。在应用到 Internet 上之后, Microsoft 将其改名为 ActiveX。随着软件开发部件化的发展趋势, OLE 技术已走近每个软件开发人员的身边, 成为提高软件开发质量的一种重要技术手段, 而不再是一个抽象的名词。例如在 AutoCAD R14 中就支持了 OLE 自动化。OLE 自动化是 OLE 技术的一部分, 是使一个应用程序可以为其它应用程序、开发工具或是宏语言提供对象的技术, 是一种使一个应用程序驱动另一个程序的较为灵巧的方式。通过 AutoCAD 对 OLE 自动化的支持, 用户可以使用常见的且易学易用的编程语言和工具, 如 Visual Basic, Delphi 等编写应用程序来对 AutoCAD 进行客户化或二次开发。

OLE 技术另一个对软件开发来说极为有用的一部分就是 OLE 自定义控制件, 即 OCX 控制件。由于 OCX 控制件加载迅速并支持直接接口, 能够快速改变属性并执行函数调用, 而且不存在传统的 VBX 控制件的局限性, 因此 OCX 成为 VBX 的替代品。我们知道, 控制件对于软件的用户界面和易用性具有极其重要的作用, 当所用开发工具提供的内嵌控制件不能满足需要时, 就有必要自己开发或借用第三方开发商提供的控制件。例如, 在很多工程计算或 CAD 软件中都需要输入大量的计算数据, 而这些数据有相当数量是以二维数组形式存储的, 如果能有一个类似于 Excel 那样电子表格控制件用于显示、输入和编辑这些二维数组中的数据, 不但用户使用方便, 易于掌握, 而且软件开发者的工作量也将大大减小。在 Visual C++ 中虽然有一个名为 Grid 的控制件, 但它只适合于显示数据, 而不能实现实时编辑数据, 应用起来并不很理想, 因而开发一个自己的实用表格控制件将很有意义, 在今后的软件开发中可以反复重用。出于这种想法, 笔者针对桥梁 CAD 软件的特点, 使用 Visual C++ 开发了一个名为 Egrid 的 OCX 控制件, 使它不但可以显示数据, 而且还可以随时输入或编辑数据, 工作方式极其类似于 Excel。当然, Visual C++ 并不是唯一的 OCX 控制件开发工具, Visual Basic 5.0 版也具有了开发 OCX 控制件的能力。

OLE2.0 的内部工作机制是极为复杂的, 数量繁多的 OLE API 和很多不熟悉的新概念也会令人不知所措。但借助于 MFC 的封装, OCX 控制件的开发过程得以大大简化, 开发人员不必对 OLE 了解过多即可编写 OCX 控制件。通过 Visual C++ 的 Control Wizard 生成的代码框架包含了一个 OCX 所需的最少代码, 用户需要向从 COleControl 派生的控制件类添

加所需的属性 (Property)、方法 (Method) 和事件 (Event) 来实现预期的设计功能。

在对应于 Egrid 的控制件类 CEgridCtrl 中, 我们需要增加自定义属性 m_RowHeight 用以表示表格的每行行高。为了让 Egrid 对不同外部事件分别做出响应, 我们不但在 CEgridCtrl 中加入了库存 (Stock) 事件 KeyPress, KeyDown, MouseMove, ClickDbClick, 还增加了自定义事件 RowColChange 和 GetFocus, 分别用来处理当前单元格变化和获得输入焦点的事件。在 CEgridCtrl 中定义了大量的方法作为调用接口供应用程序控制 Egrid, 这些方法主要可以进行表格建立初始化, 设置表格行数及顶部行的行号, 设置及显示文本和插入光标 (Caret), 获取单元格内的数据, 控制滚动条, 处理各种键盘事件。为了让 Egrid 响应 WM_KEYDOWN 消息来处理诸如按下箭头键这样的事件, 应该重载 CWnd::PreTranslateMessage 成员函数使 WM_KEYDOWN 消息可以发送到 Egrid 窗口而不被过滤掉。为了正确显示文字和插入光标, 在单元格中滚动显示较长文本, 类 CEgridCtrl 中还需定义若干私有数据成员来记录当前表格顶部行行号、当前单元格、光标所在位置、示出文本在实际文本中的位置, 以及表格各列的宽度等信息。仿照 Windows 应用程序和控制件风格, Egrid 中亦加入了 COPY、CUT 和 PASTE 的热键操作。

5 结束语

90 年代以来的计算机技术的发展之快有些另人目不暇接, 各种系统软件与开发工具的版本在不断地更新, 一些新兴的软件开发技术为工程 CAD 软件的开发提供了更加有力的技术手段, 为 CAD 软件开发人员提供了可以驰骋的更加广阔的空间, 但同时也对开发人员提出了新的挑战。作为 CAD 软件开发人员, 我们应及时地将这些先进的软件技术应用到软件开发中去, 提高我国目前桥梁 CAD 软件的开发与应用水平。本文所述仅是作者在软件开发过程中所遇到的一些新技术, 只是当今软件开发技术中的很少的一部分, 希望能对广大 CAD 软件开发有所帮助。

参考文献

- 1 王书庆, 石洞, 石志源. 工程 CAD 集成系统发展. 公路工程计算机应用论文集, 1996
- 2 软件开发文集 (第六至九辑). 科学出版社, 1996~1997
- 3 Microsoft Corporation. Visual C++ Tutorials (Version 4.0). 1995
- 4 方铁. AutoCAD C 语言高级编程. 清华大学出版社, 1995