MIS 开发的先进开发工具和开发环境

赵丹亚　（首都经济贸易大学经济信息管理系　100026）

摘要：本文主要讨论可视化开发环境、事件驱动机制、客户机/服务器解决方案和更高层次的开发平台对 MIS 开发的影响，以及如何充分利用先进的开发工具和开发环境，高效地开发高质量的 MIS。

关键词：可视化　事件驱动机制

近几年来，出现了大量的、先进的开发工具和开发环境，使得已往既繁琐，又费时费力的编程序、写代码以及编制文档的工作越来越多地由有关的工具或环境自动完成，使得 MIS 的开发工作更为方便和高效，也使得 MIS 的应用人员加入到开发 MIS 的队伍中来成为可能。本文主要通过 Visual Basic、Visual FoxPRO、Delphi 和 PowerBuild 等新型的开发工具和环境说明 MIS 开发的一些新特点，为计算机应用软件开发人员以及更广范的 MIS 应用人员开发 MIS 提供参考。

一、可视化集成开发环境

早期国内开发 MIS 普遍采用 XBASE 系列开发工具，其中使用最多的有 dBASE、FoxBASE 等。由于其提供了方便的对关系数据库定义、操作的方法，因而有力的促进了对 MIS 的开发。但是严格地说，XBASE 系列工具不能算是真正的开发工具，至多只能算是一种数据库语言，而且还不能算是完善的关系数据库语言。随着 MIS 的需求量增加，结构的复杂、庞大，特别是随着以 Windows 系统为代表的图形用户界面的出现，早期的 XBASE 系列工具已不能适应越来越广泛的开发高质量 MIS 的需要。

图形用户界面可以说是计算机软件发展史上的一个转折点。现在的计算机用户不再需要记忆复杂、繁琐且语法严格的近乎苛刻的命令，也不再需要用键盘键入它们，而只需使用鼠标点选所需的功能菜单或形象直观的图标即可。而在过去，要开发这样的应用软件只有微软公司的 S.D.K 一种。而使用 S.D.K 除了必须具备 C 语言的经验外，还需掌握多达几百个函数，特别是有关函数调用时参数的传递，稍有不慎就可能招致各种各样的错误。

自从微软公司推出 Visual Basic 后，使得开发 Windows 平台下的图形用户界面的应用软件变得轻松快捷。使用 Visual Basic，不但不需要 C 语言的知识，甚至没有任何软件开发经验的人，稍加学习和训练亦可开发出界面友好，功能强大的图形用户界面的应用软件。Visual Basic 系统首次引入可视化程序设计的概念和机制。在用户界面程序设计过程中真正实现了“所见即所得”。

可视化开发环境的突出优点是直观，设计用户界面就好象“画”用户界面一样。而且修改方便，当需要改变窗体布局、修改有关控件的属性时，都可以直接在窗体上完成。例如要移动某个按钮的位置，改变某个文本框的大小，只需直接用鼠标拖拽即可。要修改有关控件的属性也只需在相应控件的属性表中，选定相应的属性，然后再输入或是选择所需的属性即可。而所有这些都无需编写代码，或是只需编写很少的代码。而在过去，这部分内容是 MIS 开发人员最为耗时费力，也最易变更的。由于利用可视化环境开发 MIS，所需的编码量可以大幅度减少，因而可以大大节约开发、调试所需的时间，相应的维护工作也就更为简单。甚至相当多的工作完全可以由 MIS 应用人员直接参与完成。

传统的程序开发大多是通过编辑器进行的，编写好代码后，需要编译，发现错误后，要回到编辑器里修改，然后再编译。编译正确后，可以运行，当出现错误还需要再编辑、编译、运行。而先进的可视化开发环境都是高度集成的。界面的设计、代码的编辑、编译以及程序的运行都集成在一起，使用起来十分方便。例如在编写 Visual Basic 代码的过程中，当键入了某个控件对象名后，会自动弹出该对象可使用的方法或函数名列表供候选。当键入了某个函数或过程名时，又会弹出相应的参数个数、类型等列表供参考。又如在编写 PowerBuild 代码过程中，系统会自动根据键入代码的类别是变量、常量、对象、属性还是方法而标以不同的颜色，每当结束一行的输入，都会立即进行语法检查。而且在设计的任何阶段，随时可以预览设计的结果，甚至可以单独运行某个窗口。从而给代码的编写和调试带来了极大的方便，也大大提高了开发效率。

二、事件驱动机制

传统的 MIS 程序都是采用过程控制的。一个 MIS 划分成若干模块，再进一步分成多个功能独立的过程。系统运行时，由专门的程序负责根据用户的选择和操作指令调用相应的过程，完成相应的操作。这种控制方法在文本界面或命令行方式工作时还可以适应，但是在图形用户界面下则很难控制。

图形用户界面的应用软件其工作方式都采用事件驱动机制。可视化开发环境通常都为图形界面上的各种控件预先设置了不同的事件。例如窗口有打开、关闭、单击、双击、隐藏、显示、拖放等事件，命令按钮有单击、得到焦点、失去焦点等事件。对不同对象的每个事件，系统都会产生相应的消息并传递给相应的消息响应函数或过程（这些函数或过程通常都是以“<对象名>-<事件名>(<参数>)”的形式来命名的，例如 Text1-KeyPress (Ascii)）。开发者可以根据需要，针对具体的消息响应函数或过程编码。

事件驱动机制的突出优点是使复杂、庞大的代码编写工作简单化、小型化。开发者无需关心如何去接收或判别用户所作的操作，如何进行程序流程的控制，而将主要的精力都放在要实现的任务上。只需简单地针对各个控件的一些事件编写有关的代码即可。显然这些代码都是很简短的（大多只有几行、十几行），而且功能相对独立，因此编写、调试以及修改、扩充都很方便。

按事件驱动机制开发的 MIS，将程序分成了许多独立的片段，某段程序只同与之关联的控件和事件关联。系统运行时，所有的程序不再是一条指令一条指令的顺序执行，而是每段程序只有当相关联的事件发生时才会执行。因此，实际上是用户而不是程序在控制运行。因此开发人员必须学会将控制权交给系统、交给用户的控制机制。

三、客户机/服务器方案

最早的 MIS 是建立在中型乃至大型计算机上的，可以高效地管理和高速地处理大量的数据。但是价格昂贵、使用不便、升级换代困难是其致命的缺陷。PC 机发明以后，其发展速度较大、中型计算机要快得多。现在的 PC 机的功能早已超过了以前的大型机。已广泛地应用于各行各业的管理中。但是众多的 PC 机中的数据如何共享、如何保持一致则是难以解决的问题。

90 年代以来，客户机/服务器方案越来越广泛地应用于 MIS。所谓客户机/服务器方案就是利用通信技术和网络技术，将大型机强大的数据存储和处理能力与 PC 机方便的用户交互能力有机的结合起来，从而达到最佳的应用效果。一般的客户机/服务器系统是通过局域网连接在一起的一些计算机。其中有一台高性能的 PC 机（也可以是小型机、中型机或大型机）作为服务器（也可以是多个服务器，分别承担不同的服务），负责提供数据存储、检索和其他处理功能；其他 PC 机则作为客户机，负责处理所有的用户界面的任务。系统工作时，客户机/服务器相互配合，协同完成有关的操作。例如客户机接收用户的查询要求，生成相应的结构化查询指令传送给服务器，服务器根据指令完成检索工作，并将查询结果传送给客户机，用户可以在客户机浏览查询结果。

客户机/服务器方案通常都以大型关系数据库，例如 Sybase、Oracle、Informix 等为基础。开发这样的 MIS，应该选择 Delphi 或 PowerBuild 开发工具。XBASE 系列工具虽然也属于数据库，但是其数据格式单一，而且主要是针对单用户存取设计的，特别是在数据的安全性与完整性上功能较为薄弱。而 Delphi 或 PowerBuild 等开发工具都支持多种数据库，可以根据需要选择功能、价格上都合适的数据库。使用大型数据库同 XBASE 系列数据库有较大的差异，XBASE 系列数据库是以关系（二维表）为基本单位，每个关系即为一个数据库文件。而在大型数据库中，是将多个表，以及查询、视图等作为一个数据库文件存储的。初看起来，大型数据库控件复杂，层次多，但是由于其提供的前端开发工具都是面向对象的，所以更加符合 MIS 人员的业务习惯，设计起来更为直观。而且由于其都提供了功能强大 SQL（Structure Query Language，结构化查询语言），使得原来使用 XBASE 系列工具开发时需要设计的各二维表之间的联动关系等都对用户透明，由系统根据用户对数据库的定义和查询的要求自动实现。所以从某种意义上说，特别是对于 MIS 应用人员来说更为简单了（或者说需要学习和掌握更高层次的知识）。

使用 Delphi 或 PowerBuild 等开发工具开发客户机/服务器模式的 MIS，可以保证所开发的 MIS 具有较强的数据与程序的独立性。像 Delphi 提供了 BDE（Borland Database Eninge，Borland 数据库引擎）、DBD（Datebase Desktop，数据库桌面）、DFE（Database Form Expert，数据库表单专家）等数据库应用工具以及多种形式的数据访问控件和数据意识控件。通过 BDE 可以将应用程序开发的接口与连接数据库的部分分开。这样当应用系统需要改换其他目标数据库时，基本上不用修改代码，而只需简单地重新配置 BDE，使之作相应改变即可。科学使用各种数据访问控件和数据意识控件可以提供多种层次上的数据与程序的独立性。例如可以使用 Database 控件定

义使用的数据库集合；在此基础上使用 Table、Query 等控件定义使用的表或查询；然后使用 DataSource 控件指定某个应用的数据来源，最后使用 DBGrid（数据库网格）、DBText（数据库文本框）、DBListBox（数据库列表框）、DBNavigat（数据库浏览按钮）等各种数据意识控件定义数据库操作界面。这样无论需要对哪一层控件对象的变动，对系统的其他部分的影响都会控制在最小。例如需要将某个数据来源由一个 Table 改为另一个 Table 或是某个 Query，只需修改相应的 DataSource 的某些属性即可，与之相关的各种数据意识控件都无需变动。而使用 StoreProc、BatchMove 等控件，可以将访问在某个远程服务器上数据库的 SQL 应用过程或是 SQL 应答建成标准化的过程，供多个应用过程重用。

随着 Internet/Intranet 的应用日益广泛，一些 MIS 系统开始采用浏览器模式，越来越多的单位在网上建立自己的站点和形式多样的主页，也有越来越多的企业开始建立自己企业内部网。对于这种模式的 MIS，对数据和程序的独立性提出了更高的要求，因为这时的客户机和服务器可能相距甚远，多种查询要求都要通过主页上的热键或是控件来实现。对此除了可以使用上述 Visual Basic、Delphi、PowerBuild 等开发工具外，Java、ActiveX 控件都是可选的方便高效的开发工具。

四、更高层次的开发平台

以上介绍的主要是针对大型 MIS 开发的一些新特点。在我国还有更多的计算机应用于较为简单的 MIS。例如某个部门的资料档案管理、某个商店的销售管理、某个仓库的库存管理等。对于这些应用，当然可以使用上述工具来开发，但是，如果由应用人员直接利用有关的应用软件，在更高的开发平台上实施应该效果更好。

例如 Office 97 办公套件中的 Excel 97，是一个功能强大的表格式数据管理和分析软件，可以方便地进行各种查询、统计、分析和辅助决策工作。特别是它还提供了 Visual Basic for Application（VBA）作为其宏语言（Office 办公套件中的各软件均提供 VBA），为使用这些软件的广大用户提供了一个新的、更高层次的开发平台。

VBA 实际上是 Viual Basic 与相应的应用软件结合的产物，它是根据其嵌入软件的不同，增加了对相应软件不同对象的控制功能。例如 Excel 97 的 VBA，主要增加了关于 Excel 工作簿、工作表、区域、数据透视表等对象的属性、事件和方法。在 Excel 97 中使用 VBA，可以更方便地操作 Excel，更好地控制 Excel，进一步深入发掘 Excel 的强大功能，甚至可以在很短的时间内就在 Excel 环境中开发出所需的完整的管理信息系统。

使用 VBA 在 Excel 等应用软件的基础上开发 MIS，其最大的优点是可以直接利用该应用软件的强大功能。例如根据原始数据制作各种统计图表，对数据进行排序、统计、查询等均可通过简单操作实现。甚至像求解线性方程、非线性规划这样复杂的问题也有相应的命令可以解决。虽然 VBA 涉及到的对象、属性、方法和语句众多，完全掌握有一定困难，但是利用其提供的宏记录器，将需要执行的操作录制下来，转换成相应的 VBA 代码，在此基础上再稍加修改即可得到所需的应用程序。将编码过程变成“示范”过程，大大提高了编码的效率。再加上由于 MIS 应用人员经常使用有关的应用软件，相对来说对其比较熟悉，因而可以更快地进入“角色”，甚至可以直接参与开发。

在应用软件基础上，使用其提供的宏语言进行 MIS 开发的大致步骤如下：

- 根据系统功能划分模块；
- 将较为复杂的任务分解成若干简单的任务；
- 针对每个简单任务录制或编写相应的宏；
- 编辑、优化宏；
- 将有关的宏装配在一起。

结论

可视化开发是软件开发方式上的一场革命，它使 MIS 开发的代码量大大减少；集成开发环境则使得开发过程更加方便、快捷；客户机/服务器以及 Internet/Intranet 模式进一步提高了数据和程序的独立性，从而使得程序维护的工作量减少到最小；更高层次的开发平台利用相应应用软件的众多功能，可以开发工作更为简单、直观。所以，使用先进的开发工具其开发效率和开发质量都是传统的开发方法所无法比拟的。而所有这些，最有意义的是它使软件开发从计算机专业人员的手中解放出来，使得广大 MIS 应用人员参与 MIS 的开发成为可能。而只有广大 MIS 应用人员参与 MIS 的开发，才是彻底解决所谓“软件危机”的根本出路。

参考文献

[1] 《Delphi 大全》Harold Davis 著，宋建云 等译，电子工业出版社

[2] 《PowerBuild 5 应用程序开发指南》Bill Hatfield 著，史森 等译，清华大学出版社

[3] 《Visual Basic 5》Michael Halvorson 著，赵丹亚 等译，人民邮电出版社

[4] 《Excel 97 Visual Basic》Reed Jacobson 著，赵丹亚 等译，人民邮电出版社

（来稿时间：1998 年 3 月）