基于MapInfo的GIS应用系统开发技术

刘迎春 （浙江工业大学集成化信息系统研究所 310014）
严寒冰 （浙江工程学院信息与电子学院 310014）

摘要：本文详细探讨了MapInfo中的集成化地图技术，该技术不同于基于MapInfo平台的简单二次开发，而是能实现MIS与GIS的集成和一体化。并举出用Delphi作为应用前台，后台调用MapInfo，在应用系统中实现相应的GIS功能的一些主要方法。

关键词：GIS MapInfo Delphi 集成化地图

1 前言

MapInfo具有其特有的编程工具——MapBasic。MapBasic是一个能驱动(Automate)和修改(Customize)MapInfo的内置软件包。MapBasic是一个开放的开发环境，它集成有一个编译器，能把MapBasic程序编译成一个可在MapInfo中执行的可执行文件，并且所有的MapBasic程序都不能脱离MapInfo平台而运行。

虽然MapBasic是一个全功能（All-featured）的编程语言，其所含有的300多个函数和命令，几乎能实现MapInfo所有的处理空间地理数据和属性数据的能力，但基于其上的应用程序，开发过程困难，应用程序功能有限并且界面粗糙，使用不便。因此，MapInfo在Windons的OLE技术的基础上，提供了一种MIS&GIS一体化开发技术——集成化地图技术。

2 集成化地图（Integrated Mapping）的定义

在MapInfo中，集成化地图是指在一些可视化编程软件（Delphi等）的应用程序窗口中，集成一个MapInfo地图窗口，在该窗口中可以由MapInfo控制并完成MapInfo中几乎所有的地图操作，并通过OLE或DDE实现信息的通信，可以理解为把MapInfo的应用元素（如地图窗口）集成进其他应用程序（Delphi应用程序）。

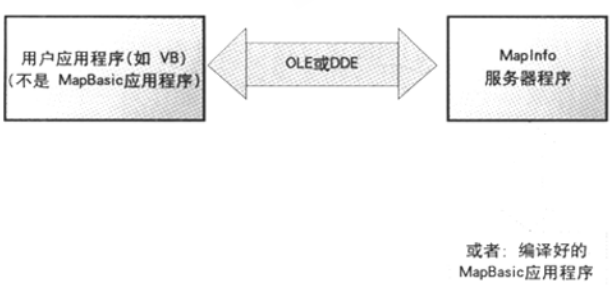

图 1 集成化地图的应用过程

在Delphi等的程序中，首先要有一个命令在后台调用MapInfo，并构建一些字符串来代表MapBasic的命令，用OLE自动化或DDE把这些字符串传递给MapInfo，MapInfo对合法的MapBasic命令进行执行操作。可用图1来表示集成化地图的应用过程：

3 集成化地图的技术

3.1 系统要求

(1) 必须在Microsoft Windows环境下，使用MapInfo 4.0以上版本。

(2) 用户机必须有足够的资源来同时运行MapInfo和用户应用程序。

(3) 用户应用程序（如Delphi）必须能作为OLE自动化的操作者或作为DDE的用户，并且用户应用程序能创建用户界面，并能决定用户的Windows句柄。

(4) 集成化地图使用的是OLE自动化(Automation)，对OLE嵌入（Embedding）不能使用，所以在把MapInfo的地图窗口嵌入到你的应用程序中时，要给MapInfo一系列命令把MapInfo窗口设置为应用程序的子窗口。

3.2 主要编程方法

(1)调用MapInfo （Starting MapInfo）。以Delphi为例，用Delphi的Createoleobject()函数，并把其返回值赋给一个Delphi的variant变量，如已定义'MapInfo'为一个variant变量，则下句将调用MapInfo:

MapInfo:=CreateoleObject('MapInfo.application');

这时MapInfo将在后台自动被运行。

(2)把命令传给MapInfo（Sending Commands To MapInfo）。在后台成功地运行了MapInfo后，可以构建一些文本式的字串来代替MapBasic命令。如想在MapInfo中打开States.TAB表，用MapBasic的Open Table命令，

在Delphi中可构建如下的字串:

```
Dostring := 'Open Table  States.Tab  inteactive';
```

然后用OLE自动化将这个MapBasic命令字串传给MapInfo:

```
MapInfo.do(Dostring);
```

当用这个OLE的do方法时，MapInfo就执行了Dostring字串中的MapBasic命令。

(3)从MapInfo中得到数据（Querying Data From MapInfo)。我们常常需要在MapInfo中得到它的一些系统信息或运行结果，或得到MapBasic表达式的值，同样可以用字串来构建这些信息或表达式。如想知道当前窗口的ID号，用下面字串表示:

```
Dostring := 'WindowID(0)';
```

然后用OLE自动化把字符串传给MapInfo

```
var
  Result :string;
Result := MapInfo.Eval(Dostring);
```

这里用了OLE的Eval方法，MapInfo执行这个MapBasic函数，并把结果以字串形式传回给Delphi的变量Result。

(4)重新父化MapInfo窗口（Reparenting MapInfo Windows)。在启动MapInfo后，在Delphi用界面上建立一个地图载体panel1，首先传给MapInfo一个Set Next Document命令，把MapInfo的地图窗口设置成Delphi程序的子窗口，并用MapBasic命令Set Application Window来把MapInfo的对话框和出错信息交与应用程序(Delphi)来管理。下例将在Delphi的panel1中显示一张已打开的地图States（hwnd为Delphi的String型变量)。

```
str(panel1.handle,hwnd);
Dostring :='set next document parent '+hwnd+' style 1';
MapInfo.do(Dostring);
Dostring := 'Set application Window ' + hwnd;
mapinfo.Do(Dostring);
Dostring := 'Map From States';  // 在此之前已打开了States表
MapInfo.do(Dostring);
```

Set Next Document命令中，确定了Delphi的控制句柄，当下次再创建MapInfo窗口时，最新被创建的窗口的将被重新父化，把Delphi的应用程序作为它的父亲。

(5)集成MapInfo的工具按钮（Integrating MapInfo Toolbar Buttons)。当想把MapInfo的命令按钮集成进用户程序中，可用RunMenuCommand方法，如MapInfo的放大，缩小，漫游按钮，可在Delphi的Button控件的Click事件中加入以下相应的语句:

```
MapInfo. RunMenuCommand 1705;  /* 放大 */
MapInfo. RunMenuCommand 1706;  /* 缩小 */
MapInfo. RunMenuCommand 1702;  /* 漫游 */
```

其中1705，1706，1702是在MapBasic.Bas头文件中定义的放大、缩小、漫游的宏。

(6)关闭MapInfo（Terminating MapInfo)。当在Delphi中用CreateOLEObject()函数把MapInfo作为一个对象调用后，MapInfo将在它所对应的对象变量释放时自动关闭。如果定义的对象变量是局部变量，它将在当前过程运行终止时自动释放，如为全局变量，可定义它的值为Nothing.

```
MapInfo: = Nothing;
```

(7)回调技术(Callback Technology)。在应用程序中，可以通过OLE的do方法把Mapbasic的命令和函数传给MapInfo执行，如果需要把MapInfo信息自动地传给应用程序，则要用回调技术，即产生一个事件来让MapInfo去调用应用程序。

例如在MapInfo窗口中选择了某个对象，在回调方式下，MapInfo将把这个信息自动传给应用程序，应用程序可对这个信息进行处理。下面以Delphi程序为例来说明这种回调技术:

· 在Delphi的Main_Unit中定义并调用MapInfo:

```
var
  Form1: TForm1;
  MapInfo:variant;
  dostring,hwnd:string; // 定义部分
implementation
uses
  do_unit; // 调用单元
  {$R * .DFM}
procedure TForm1.FormActivate(Sender: TObject); // 调用MI
  begin
  MapInfo:=createoleobject('MapInfo.application'); // 调用
  str(panel1.handle,hwnd);
  dostring:='set next document parent '+hwnd+' style 1';
  MapInfo.do(dostring);   // 把下一个MI窗口设为Panel1的子窗口
  dostring:='open table "d:＼liuyc＼states.tab"';
  MapInfo.do(dostring); // 打开States表
  dostring:='map from states';
  MapInfo.do(dostring); // 创建地图窗口
```

```
end;

procedure TForm1.ButtonClick(Sender: TObject);
begin
  query_point; // 调用 do_unit 中的自定义过程
end;
```

· 在 do_unit 单元中定义过程 query_point

```
interface
uses
  custom_server,OLeAuto,main_unit;
    // 调用 custom_server 单元中的 Point_query 过程
  procedure  query_point;
  var
  ole_obj:Toleserver; // 定义部分
implementation

procedure  query_point;
begin
  ole_obj:=Toleserver.Create;  // 创建一个 Toleserver 类
  MapInfo.setcallback(varfrominterface(ole_obj.autodispatch));
    // 调用 OLE 的 setcallback 方法
  dostring:='Create Buttonpad "Demo" as ToolButton id 2001
drawmode 34 calling ole "Point_query"';
  MapInfo.do(dostring);
    // 在 MI 中创建一个调用 Point_query 过程的按钮
  dostring:='run menu command id 2001';
  MapInfo.do(dostring); // 调用并执行 Point_query 过程
end;
end.
```

· 在 custom_server 单元中具体传递回调的主要信息

```
interface
uses
  OLeAuto,Dialogs,main_unit,SysUtils;
type
  Toleserver=class(TAutoObject)
    Automated
     procedure Point_query(dostring:string);
   end;
implementation
procedure Toleserver.Point_query(dostring:string);
Var strx,stry:string;
begin
dostring:='dim fx as float';
MapInfo.do(dostring);
dostring:='dim fy as float';
MapInfo.do(dostring);  // 在 MI 中定义了 fx,fy 两个变量
dostring:='fx=commandinfo(1)';
MapInfo.do(dostring);
    // fx的值为用户鼠标在地图上单击位置的地理X坐标
  dostring:='fy=commandinfo(2)';
  MapInfo.do(dostring);
    // fy的值为用户鼠标在地图上单击位置的地理Y坐标
  strx:=MapInfo.eval('fx');
    // 用 EVAL 方法把 fx 值传给 Delphi 变量 strx
  stry:=MapInfo.eval('fy');
    // 用 EVAL 方法把 fy 值传给 Delphi 变量 stry
  showmassege(strx);
  showmassege(stry); // 在 Delphi 信息框中显示已被传递
的信息
end;
end.
```

在例子中，通过Delphi的Button控件，单击MapInfo的地图窗口，在Delphi中创建一个Toleserver类，并运用OLE的Setcallback方法，相当于在MapInfo中创建了一个ID号为2001按钮，这个按钮将执行Delphi中的point_query过程。所以通过这种回调技术，在MapInfo窗口的单击，使MapInfo响应，并把单击中的一些坐标等信息反传回Delphi，在用户看来，好象Delphi工具在地图上单击而直接获得地图上该点的坐标信息和其他相关信息。

4 小结

通过OLE技术，把地理信息系统软件MapInfo和其他应用系统软件如Delphi、VB、C/C++等进行集成，让MapInfo重点管理与空间，图形有关的数据，让Delphi或VB重点进行用户界面，应用逻辑，数据库等管理工作，既发挥了MapInfo软件空间数据的操纵能力，又使开发的应用程序美观，方便，实用。特别在某些既要管理有广阔的空间跨度的实体，又要有严密的数据库管理能力的系统中，GIS与MIS软件的集成运用，是系统开发的首选方案。■

参考文献

1 边馥苓．地理信息系统原理和方法．测绘出版社．1996.8

2 MapInfo Corporation, MapBasic Reference Guide, MapInfo Corporation, 1997.6