

# 基于.NET的流程图绘制程序的设计与实现

黄烟波,赵旭华,刘中宇

(中南大学 现代教育技术中心,湖南 长沙 410083)

**摘要:**对流程图绘制程序的设计进行了详细、整体的阐述,介绍了.NET环境下GDI+绘图机制,分析了流程模块的绘制、插入、修改、智能拼接等功能的实现原理,利用树形结构数据存贮流程记录信息,并结合XML与ADO.NET数据交互技术,实现了流程绘制、信息存贮、数据转换、图形输出等功能。

**关键词:** .NET;流程图;GDI+;绘制;XML

**中图分类号:** TP391.72

**文献标识码:** A

**文章编号:** 1673-629X(2007)05-0231-03

## Design and Realization of a Program for Drawing DFD Based on .NET Platform

HUANG Yan-bo, ZHAO Xu-hua, LIU Zhong-yu

(Education Technology Center of Central South University, Changsha 410083, China)

**Abstract:** Presents the whole design of a program for drawing the data flow diagram, introducing the GDI+ drawing method based on the .NET platform, analyzing the realization principles of such functions of those flow modules as drawing, inserting, modifying, and auto-linking etc, finding that through integrating XML into ADO.NET and making use of the tree structure to memory the data, this program can realize such functions as DFD drawing, data memory, data transforming, and picture outputting etc.

**Key words:** .NET; DFD; GDI+; drawing; XML

### 0 前言

在衡量计算机水平的标准体系中,程序设计是一个很重要的评定指标。因为流程图是图形化了的程序,它形象地描绘了程序的结构及其所采用的算法,能使读者一目了然地理解程序设计者的思想,所以可以用设计者所绘制的流程图来检测与评定其程序设计能力。

.NET是微软公司继ActiveX技术之后,于2000年推出的开发应用程序的最新平台,它内置了多种类型的已封装好的库类,利用其内置类能实现多种繁杂而实用的功能,如网络应用程序、多媒体应用程序、计算机图形程序等。.NET Framework为操作图形提供了GDI+[1]应用程序编程接口(API)。GDI+是Windows Graphics Device Interface (GDI)的高级实现,通过使用GDI+,可以创建图形、绘制文本以及将图形图像作为对象操作。XML(Extensible Markup Language)可扩展标记语言,作为.NET平台的核心数据存

贮技术,它不仅提供一种描述内容的标准方法,尤其是提供了以灵活方式创建数据结构的方法,而且它的自描述性质使其不依赖于任何平台和编程语言,能方便地用于异质系统或应用之间通信或交换数据。程序开发环境为Microsoft Visual Studio 2005,所选编程语言为VB.NET。

### 1 程序功能模块

根据程序流程图绘制的需要,程序应包含以下主要功能模块:流程绘制模块、流程修改模块、文件操作模块等。程序功能模块图如图1所示。

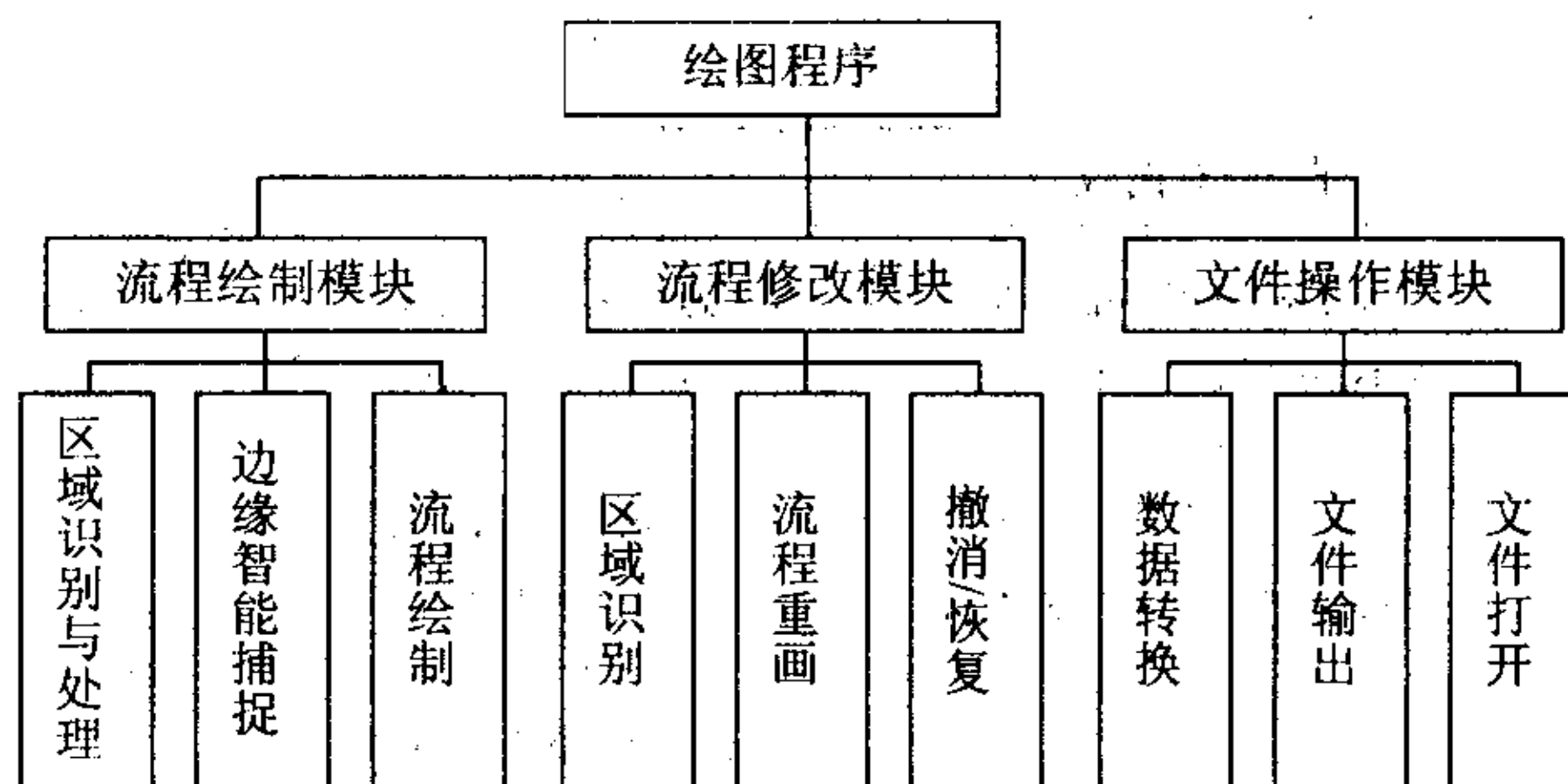


图1 程序功能模块图

各模块主要功能如下:

收稿日期:2006-08-24

作者简介:黄烟波(1959-),男,湖南邵阳人,硕士,教授,主要研究方向为网络教育技术、信息系统开发。

1) 流程绘制模块:主要解决流程图中各模块的绘制,如语句块、分支块、循环块、开关块、流程连接线的绘制、插入等操作。

2) 流程修改模块:主要解决流程图中各模块的修改,包括各模块的删除、插入、修改、撤销/恢复等操作。

3) 文件操作模块:主要解决流程图记录信息的存档(XML 格式文件)、流程图的输出(JPG 图形文件)、读取等操作。

## 2 记录数据结构

### 2.1 流程记录信息数据结构

由于流程图信息的复杂性,线性的数组存贮不能有效地解决流程模块的插入、修改、删除等操作,故程序采用树型结构来记录流程信息,其数据结构如下:

Structure record '定义流程图记录数据结构、类别、源点、终点的坐标

Dim index As Integer '定义其索引号,子结点索引永远比父结点索引大

Dim parent As Integer '定义其父结点

Dim cate As String '定义其类型

Dim text(1-7) As String '定义文本区域文字 1-7

Dim ps As Point '定义起点坐标

Dim p(1-8) As Point '定义各关键点坐标

Dim pe As Point '定义终点坐标

End Structure

### 2.2 XML 磁盘数据存储结构

XML<sup>[2]</sup>文档结构是一种文档树,其主干是父元素,分支和页是子元素,此结构分别与 ADO.NET 中的数据集、数据表、数据行相对应,故可通过 ADO.NET 与程序进行数据交互。

<Dataset> '定义的数据集

<Debatable> '定义的数据表

<Data row> '定义的数据行

<Data column> '定义的数据列

... '数据列的值

</Data column>

... '其它的数据列

<Data row>

'其它数据行

</Debatable>

</Dataset>

用 XML 格式保存的流程图记录文件数据实例如表 1 所示。

## 3 关键实现技术

### 3.1 绘图机制

.NET 环境下的 GDI+ 绘图机制与其以前版本 GDI 不同,集成于 .NET 中的控件不能直接进行绘图,而必须通过创建图形对象 Graphics<sup>[3,4]</sup> 来实现。其步骤如下:

(1) 创建图形对象,有三种方法可用于创建图形对象。一是在窗体或控件的 Paint 事件中接收对图形对象的引用,作为 PaintEventArgs 的一部分;二是调用某控件或窗体的 CreateGraphics 方法以获取对 Graphics 对象的引用;三是从 Image 继承的任何对象创建 Graphics 对象。第三种方法在需要更改已存在的图像时十分有用,本程序中因为要画许多不同的流程模块并能加以修改,因此采用从 Image 继承的对象中创建 Graphics 对象。

Dim bmp As New Bitmap(940, 664) '定义位图以填充图形控件,提供创建图形对象的图像

Dim g As Graphics = Graphics.FromImage(bmp) '从位图中创建图形对象

(2) 使用图形对象的方法在窗体或控件上绘图。

Picturebox1.image = bmp '定义流程图显示在图像框中

g.DrawLine(mypen, ps, pe) '绘制直线' mypen 为所定义的画笔,ps,pe 分别为直线的起点与终点

(3) 绘图完毕,销毁图形对象。

g.dispose() '销毁图形对象

### 3.2 绘制直线、有向线、语句块、分支块、循环块、开关块、文字块流程图

流程图由矩形、菱形、直线组合连接而成,表 2 展示了 GDI+ 绘图类的基本方法<sup>[3]</sup>。

表 1 XML 格式保存的流程信息记录实例

直线/有向线	语句块	循环块	分支块	开关块
<index>1</index>	<index>2</index>	<index>3</index>	<index>3</index>	<index>4</index>
<type>line</type>	<type>rect</type>	<type>cycle</type>	<type>ploy</type>	<type>switch</type>
...	<text1>	<text1>	<psx>365</psx>	<text1>switch
<pex>415</pex>	i=scanf("d",&i);	while i<1000	<psy>229</psy>	i</text1>
<pey>104</pey>	</text1>	</text1>	<pex>365</pex>	<text2>case
	...	<text2>i+=1;</text2>	<pey>229</pey>	3000</text2>
	<pex>416</pex>	...		...
	<pey>103</pey>	<pex>414</pex>		<pex>414</pex>
		<pey>152</pey>		<pey>354</pey>



表2 GDI+ 绘图类方法

绘制类型	函数	参数说明
直线	DrawLine(mypen, ps, pe)	Mypen 画笔, ps 直线起点坐标, pe 直线终点坐标
菱形	DrawPolygon(mypen, pf)	Mypen 画笔, pf 为菱形四个顶点的坐标
矩形	DrawRectangle(mypen, p, x, y)	Mypen 画笔, p 矩形的左上顶点坐标, x 长度, y 高度
文字	DrawString("s", Font, mypen, ps)	s 文字, Font 文字文体, Mypen 画笔, ps 文字坐标

## (1) 绘制直线、有向线。

直线、有向线的绘制比较简单,只需在控件中确定直线的起点、终点即可两点画线。

## (2) 绘制语句块、分支块、循环块、开关块。

这四种模块的绘制比较复杂,它们由直线、矩形、菱形组合而成。绘制时先需对其坐标进行区域识别,确定其是插入操作还是增加操作,如是插入操作则将欲插入模块信息与其父模块信息增加至流程记录中,并根据插入的位置扩充其父模块的边界,再画出更新后的所有模块;如是增加操作则将欲增加模块的流程信息增加至流程记录中,画出新增模块即可。

## (3) 绘制文字块。

文字块的绘制比较特殊,其步骤如下:先取得当前鼠标所点击的坐标,再遍历已保存的流程记录,根据各模块的坐标位置最终确定欲添加文字的模块,此时将隐藏的文本框显示在特定位置,供用户输入文本,以回车键结束输入,并将文字保存在模块的文字区域中,最后根据更新后的流程记录信息重画流程模块即可实现各模块中的文本输入与保存。

程序中定义了一个画图类,用以绘制流程图中各流程模块,限于篇幅,在此仅列出画语句块的代码。

```

Class DrawModel()
Sub Dwrect()
Dim insert Boolean = False
Dim i As Integer
For i = Max-num To 0 Step -1 '遍历已保存的流程记录信息,查找欲插入流程模块的父流程模块
If rd(i).p1.X < ps.x AndAlso rd(i).p1.Y < ps.y AndAlso rd(i).p8.X > ps.x AndAlso rd(i).p8.Y > ps.y Then '根据坐标信息确定父流程模块, p1、p2 父模块左上/右下坐标, ps 子模块坐标
'此处将欲插入模块的信息增加到流程记录信息中,画出流程模块
End If
Insert = True '设置此模块为插入
Exit Sub '退出父模块循环
Next
If insert = False Then
'此处将欲增加的模块的信息增加到流程记录信息中,画出

```

流程模块

End If

End sub

End DrawModel

## 3.3 流程对象的插入修改

流程图可由单独的流程模块组成,也可由嵌套的流程模块组成。在父流程模块中插入子流程模块应先通过坐标信息遍历流程信息记录数据,找到欲插入子模块的父模块并将父模块信息写入子模块的数据中,画出子模块,再视子模块插入情况相应扩充父模块的边界即可完成模块插入。

流程图的修改也与上述过程类似,先由坐标信息经遍历找到要修改的流程模块,取出其数据内容放入可交互控件中,修改后再将其覆盖原始数据即可完成模块修改。

以下代码说明了循环流程模块中插入分支流程模块的操作。

```
Dim i As Integer
```

```
For i = Max-num To 0 Step -1 '遍历已保存的流程记录信息,查找欲插入流程模块的父流程模块
```

```
If rd(i).cate = "cycle" Then
```

```
If rd(i).p1.X < ps.x AndAlso rd(i).p1.Y < ps.y AndAlso rd(i).p8.X > ps.x AndAlso rd(i).p8.Y > ps.y Then '根据坐标信息确定父流程模块, p1、p2 循环模块左上/右下坐标, ps 分支模块坐标
```

```
Exit For '查到后退出循环
```

```
End If
```

```
End If
```

```
Next
```

```
rd(i + 1).cate = "pjoy" '为子流程模块的数据赋值
```

```
rd(i + 1).parent = i '设置子流程模块的父流程模块,即为树结点的上级父结点
```

```
...
```

```
rd(i).p2.Y = rd(i).p2.Y + rd(i + 1).p8.Y - rd(i + 1).p1.Y '父流程模块修改相应数据,扩充边界
```

```
rd(i).p3.Y = rd(i).p3.Y + rd(i + 1).p8.Y - rd(i + 1).p1.Y '父流程模块修改相应数据,扩充边界
```

```
'此处放置根据更新的数据重画各流程图的代码
```

## 3.4 以 XML 格式保存与打开流程记录

程序中利用 ADO.NET<sup>[5]</sup>实现流程图记录与 XML 之间的数据交互,限于篇幅,在此只列举保存分支块的程序代码。

```
Dim usernum As DataSet = New DataSet("user_no") '定义数据集
```

```
Dim record As DataTable = usernum.Tables.Add("record") '定义数据表
```

```
Dim index As DataColumn = New DataColumn '定义“索引”数据项
```

(下转第 237 页)

后,每次需要访问子系统信息或者模块信息时,直接访问这两个静态类,不需要重新访问数据库。

## 4 防止非法的访问或者攻击

### 4.1 SQL 注入攻击

SQL 注入通过网页对网站数据库进行修改。它能够直接在数据库中添加具有管理员权限的用户,从而最终获得系统管理员权限。

ASP.NET 环境下避免 SQL 注入主要有两种方式:在数据库访问层对数据库进行操作时,用 parameter 的形式传递参数;对用户输入的字符进行过滤,将一些敏感或者特殊的字符过滤掉。文中选用第一种方法,有效地避免了 SQL 注入攻击。

### 4.2 跳过登陆页面直接访问

如果非法用户获得了应用系统内部页面的地址(除登录页面以外),就可以通过直接在 IE 的地址栏中输入内部页面地址,来直接进入系统,从而跳过了验证用户合法性验证的登录页面<sup>[2]</sup>。

文中使用 ASP.NET 自带的 Forms 身份验证方式,在没有登陆时会自动引导到指定的页面,例如:login.aspx 登陆页面。系统主界面是一个大的框架,在运行过程中,浏览器地址栏的内容保持不变,系统的功能页面都在右框架之中动态加载,系统的弹出页面也都全部将地址栏屏蔽,这样有效地防止了用户直接获得并访问系统内部页面地址。

(上接第 233 页)

```
index.DataType = 111.GetType()'定义其数据类型为整数
index.ColumnName = "index"定义其名称为“索引”
record.Columns.Add(index)'将“索引”项加入到数据表中
...加入其它数据项,如“模块类型”、“文本内容”、“始/末坐标”等。
```

Dim row As DataRow'定义一数据行

```
row = record.NewRow()'从数据表中新建数据行
row("index") = rd(k).index'为表中索引赋值
row("type") = rd(k).cate'为表中模块类型赋值
row("text1") = rd(k).text1'为表中文本内容赋值
row("psx") = rd(k).ps.X'为表中坐标赋值
...
record.Rows.Add(row)'将此行加入到数据表中
userno.WriteXml("user_no.xml")'将数据集中数据以 XML 格式输入至文件
userno.Dispose()'将数据集资源释放
```

## 4 结束语

程序采用微软最新的 .NET 技术开发,整个程序

## 5 结束语

实现了一个基于 ASP.NET 技术的 B/S 信息系统的权限控制系统。该系统利用子系统编码、模块编码、页面编码和功能编码以及它们之间的关系来实现权限的控制,达到了页面级别和功能级别的权限控制,很好地解决了传统权限控制设计的不足,大大地提高了信息系统的灵活性与可扩展性。系统提出的模型和算法均来源于一个实际的项目,并取得了良好的效果。文中提出的思想不仅仅局限于 ASP.NET 平台,可以扩展到在其他技术平台(例如 J2EE 平台等),具有良好的扩展性。

### 参考文献:

- [1] 高正宪,李中学. Web 环境下基于角色的访问控制策略及实现[J]. 计算机工程,2004,30(8):133-135.
- [2] 宋维平,曾 一,涂争光,等. B/S 模式下 OA 系统的权限控制设计与实现[J]. 计算机工程与应用,2004(35):199-201.
- [3] 杨官平,陈鸿伟,李永华. B/S 模式的电厂耗差分析系统权限管理的实现[J]. 计算机工程与设计,2006,27(3):497-530.
- [4] 任善全,吕 强,钱培德. 基于角色的权限分配和管理中的方法[J]. 微机发展,2004,14(12):65-66.
- [5] 李安志,崔 蔚,徐永红. 基于角色的网页访问权限控制方案[J]. 电脑与信息技术,2004(6):4-7.

由绘制流程、修改流程、文件操作三个模块组成,在 VB.NET 2005 中调试通过。完整地实现了程序流程图的绘制、修改、存贮、输出等功能。设计了简明直观的用户界面,以 XML 格式保存的流程信息可与多种异质数据交互,具有良好的扩展性与通用性。程序可用于绘制程序流程图,如加入流程信息文件比较功能也可用于测评学习者的程序设计能力。

### 参考文献:

- [1] Microsoft. Microsoft Developer Network (MSDN)[EB/OL]. 2005. <http://www.Microsoft.com/msdn/>.
- [2] Birbeck M. XML 高级编程[M]. 裴剑锋等译. 北京:机械工业出版社,2002.
- [3] 李 海,龚学纲,李文亮. 深入研究 Visual Basic. NET[M]. 北京:电子工业出版社,2003.
- [4] 田 原. VB. NET 绘图功能研究[J]. 软件导报,2005,22(22):23-25.
- [5] Oancea G. 用实例学 Visual Basic. Net[M]. 魏迎梅等译. 北京:电子工业出版社,2002.