

# XML 及 GDI+ 在便携式电子相册中的应用<sup>①</sup>

史桂红

(健雄职业技术学院 软件与服务外包学院, 太仓 215411)

**摘要:** 针对便携式电子相册的优越性, 综合运用 XML 及 GDI+ 技术在嵌入式移动设备上开发了电子相册系统. 相册底层图片信息采用 XML 文件存储, 图片文件单独放置在设定好的文件夹中进行管理, 上层应用程序基于 .NET 平台使用 C# 编程, 嵌入式移动设备以 Windows CE 6.0 作为软件平台, 系统采用 XML 反序列化的方式获取 XML 文档数据, 最后通过 GDI+ 技术将图片信息显示到电子相册系统界面中, 实现了图片的浏览和编辑.

**关键词:** XML; 便携式电子相册; GDI+; 反序列化; 嵌入式移动设备

## Application of XML and GDI+ to the Portable Electronic Album

SHI Gui-Hong

(School of Software and Services Outsourcing, Chien-shiung Institute, Taicang 215411, China)

**Abstract:** According to the portable electronic album superiority, the electronic album system is developed in the embedded mobile device by using of XML and GDI+ technology. The album picture information is stored in the form of XML file, image files is managed individually in the folder. The upper application program is programmed by using C# in the .NET platform, embedded mobile device with Windows CE 6.0 as software drive, system obtains XML documents data by using XML deserialization, finally image information is displayed to the electronic album system interface through the GDI+ technology. The picture browsing and editing is fulfilled in the electronic album.

**Key words:** XML; portable electronic album; GDI+; deserialization; the embedded mobile device

随着数字化技术的不断发展, 一系列数码设备(如数码相机及数码摄像机等)走入家庭, 电子相册遂即诞生. 电子相册以欣赏方便、交互性强、易于保存等特点迅速得到现代人的青睐. 利用原有的 PC 机来管理数字照片已不能满足当前需求, 有时可能需要带有照相功能或摄像功能的便携式设备在任何地方捕捉一些场景, 然后在任何时间利用电子相册来管理这些数字照片, 在任何地点可以进行浏览. 同时电子相册具有传统相册无法比拟的优越性, 图文声像并茂的表现手法, 随意修改编辑的功能, 快速的检索方式, 永不褪色的恒久保存特性, 以及廉价复制分发的优越手段.

鉴于电子相册的这些优越性, 本文在基于 Windows CE 6.0 操作系统的嵌入式移动设备上实现了电子相册的开发. 在电子相册中图片采用分类管理的

方法, 用户可以用手写笔或手指拨动相应的图片索引, 实现图片类别的选取、图片的查看以及编辑.

## 1 XML及GDI+

### 1.1 XML 及其文档结构

XML(eXtensible Markup Language)<sup>[1,2]</sup>, 可扩展的标记语言, 是由 W3C 于 1998 年 2 月发布的一种标准. XML 不像 HTML 那样提供了一组事先已经定义好了的标记, 而是提供了一个标准. 利用这个标准, 用户可以根据实际需要定义自己新的标记语言. XML 是一种与平台无关的数据描述语言, 以简单的文本格式存储数据. 在嵌入式环境下, 由于硬件存储资源有限以及软硬件平台的多样性, 利用 XML 来存储数据量不是很大的电子相册数据, 是一个很好的解决途径. 相

<sup>①</sup> 基金项目:江苏省现代教育技术研究 2011 年度技术应用重点课题(19249)

收稿时间:2012-08-26;收到修改稿时间:2012-10-06

对于采用数据库存放数据, XML 更简单、更通用. 所以本文在电子相册的制作中采用 XML 文档来存储电子相册中的图片信息.

XML 中的完整数据集就是 XML 文档, XML 提供了一种结构化的数据组织方式—分层(树型)组织<sup>[3-5]</sup>. 本文中的电子相册图片信息表示主要依据大类图片与小类图片对象结构和相互之间的关系建立 XML 结构化树形文档描述结构, 该描述结构如图 1 所示.

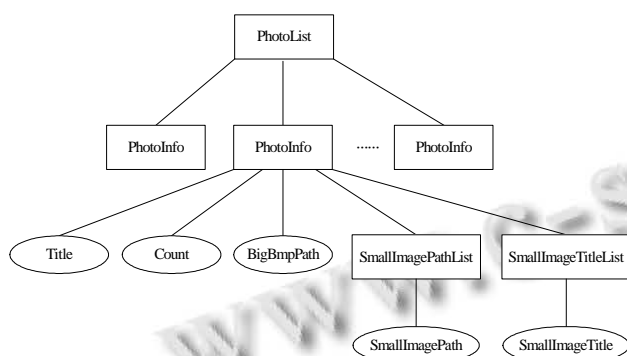


图 1 XML 树形文档描述结构

### 1.2 XML 序列化与反序列化

序列化是把对象以文件流的形式存入磁盘并以 XML 文件进行保存的过程, 例如, 可以序列化一个对象, 然后使用 HTTP 通过 Internet 在客户端和服务器之间传输该对象. 在另一端, 反序列化将该流重新构造对象. 序列化与反序列化这两个过程结合起来, 就可以轻松地存储和传输数据, 从而实现了异构多变环境下数据的交互.

序列化分为 XML 序列化、二进制序列化和 SOAP (Simple Object Access Protocol) 序列化<sup>[6]</sup>. 在嵌入式开发环境下, .NET Compact Framework 仅支持 XML 序列化, 所以在便携式电子相册的制作中采用的是 XML 序列化的方式完成对图片信息的处理. 为了开发针对 XML 文件的嵌入式应用程序需在定制 Windows CE6.0 操作系统的过程中添加 .NET Compact Framework3.5 组件的支持.

### 1.3 GDI+简介

GDI+是 GDI (Graphics Device Interface 图像设备接口) 的一个新版本, 它提供了各种丰富的图形图像处理功能, 在 Windows CE 嵌入式操作系统中, 它充当应用程序和硬件设备之间的中间层. 随着 .NET Compact Framework 的出现, 微软将 GDI+非托管的 API 进行包装, 形成类库放进 .NET 精简版的框架中,

这种类库称为 GDI+库. 在图形绘制时只需调用 GDI+库输出的类的一些方法及属性即可完成图形操作, 真正的绘图工作由这些方法交给特定的设备驱动程序来完成. GDI+使得图形硬件和应用程序相互隔离<sup>[7,8]</sup>.

在嵌入式移动设备上使用 GDI+进行绘图时, 得到的图形可以放在两种目标环境中, 一种是内存中的位图, 另一种是屏幕上窗口, 两种目标环境中的绘图都是通过创建 Graphics 对象来完成的. 在 .NET Compact Framework 中, 有四种方法可以创建 Graphics 对象<sup>[9]</sup>. 本文在电子相册制作中绘制内存中的位图采用的是 Graphics 类的 FromImage 方法, 即通过图像对象创建一个 Graphics 对象. 在屏幕上绘制图像是通过窗体的 paint 事件, 在事件的 PaintEventArgs 中获得 Graphics 对象<sup>[10]</sup>, 然后通过编写相应的代码完成图形的绘制及显示操作.

## 2 电子相册的设计

### 2.1 功能设计

电子相册主要实现的是图片的分类管理, 用户可以按类别进行图片的浏览, 每个界面显示当前的三个类别的图片信息(包括类别图片显示、类别标题), 如图 2 所示, 通过手写笔或手指拨动相应的图片索引可以进入下一个相邻的三个类别的图片信息, 通过手写笔或手指单击中间图片可以进入该类别的所有图片的显示—图片详细信息界面, 如图 3 所示. 在图片详细信息界面中用户可以通过按钮进行所有图片的浏览, 同时还可以单击添加图片按钮可以完成相应类别的图片的添加, 添加图片运行界面如图 4 所示.



图 2 图片分类信息显示



图 3 图片详细信息显示



图 4 添加图片

## 2.2 电子相册 XML 文档结构设计

根据图 1 电子相册图片信息的树形文档描述结构提取生成的 XML 文档结构如下:

```
<?xml version="1.0" encoding="utf-8"?>
<PhotoList
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
nstance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
>
  <PhotoInfo>
    <Title>传统国画</ Title >
    <Count>6</Count>
    <BigBmpPath>\001.jpg</ BigBmpPath >
    <SmallImagePathList>
      <string>\001\Chart01.png</string>
      .....
      <string>\001\Chart0n.png</string>
    </SmallImagePathList >
    <SmallImageTitleList >
      <string>百鸟图</string>
      .....
      <string>洛神图</string>
    </SmallImageTitleList >
  </PhotoInfo>
  <PhotoInfo>
  .....
  </PhotoInfo>
  .....
</PhotoList>
```

## 2.3 XML 文档反序列化类的设计

将 XML 文档反序列化为类(对象)的具体做法是:首先创建用来保存图片信息的类,然后实现反序列化,返回反序列化图片信息类对象<sup>[1]</sup>.

### 1) 节点 PhotoInfo 类的设计

为了能够按照面向对象的方式获取和处理电子相册 XML 文档中的数据,需将保存的 XML 文档反序列化为 PhotoInfo 类型的对象,这样就可以通过调用 PhotoInfo 类型对象的成员获得 XML 文档中某一 PhotoInfo 节点中的各项数据.为此,需建立 PhotoInfo 类,在类内将 XML 文档结构中的各节点定义为公有的字段.

### 2) 实现反序列化的 XMLDB 类的设计

要实现一个 XML 文档的反序列化,首先应根据给定的 XML 文档创建一个文件流对象,然后创建一个指定类型(typeof(PhotoInfo[]))的 XmlSerializer 对象,最后根据文件流对象参数调用 Deserialize()方法返回反序列化 PhotoInfo[]类型对象.在进行反序列化操作前需先添加 System.Xml.Serialization 命名空间.XMLDB 类中自定义的 XMLDeserialize 方法的主要代码如下:

```
public class XMLDB
{
  public PhotoInfo[] XMLDeserialize(string
  XmlFile)
  {
    Stream sf=new FileStream(XmlFile, FileMode.
    Open, FileAccess.Read, FileShare.ReadWrite);//创建文
    件流对象
    XmlSerializer xmls=new XmlSerializer
    (typeof(PhotoInfo[]));//声明指定类型的 XML 序列化对
    象实例 xmls
    PhotoInfo[] XmlData=(PhotoInfo [])xmls.
    Deserialize(sf);// 返回反序列化 PhotoInfo[]类型对象
    sf.Close();
    return XmlData;
  }
}
```

## 2.4 图像对象获取类 PhotoDB 的设计

PhotoDB 类用来获取图像对象,在该类中实现两个功能,功能一是根据给定的图片文件路径返回一个 Image 类型的图片对象(类中定义 GetImage()方法实现);功能二是根据指定的宽度和高度创建一个位图对象和 Graphics 对象(类中定义 GetDataInfoByImage Data()方法实现).主要代码如下所示:

```
public class PhotoDB
{
  public Image GetImage(string strFile)
  {
    string strPath=@ " \设备中文件存放的路径
    "+strFile;
    if(File.Exists(strPath))
    {
```

```

        return new Bitmap(strPath);
    }
    else
    {
        return null;
    }
}
public void GetDataInfosByImageData(int width,int height)
{
    public Bitmap bitmap=new Bitmap (width,height);
    public Graphics graphics=Graphics.FromImage(System.Drawing.Image)bitmap);
}
}

```

### 3 电子相册的实现

#### 3.1 图片显示功能的实现

电子相册中图片显示功能实现的思路是：先在内存中创建一个位图，由位图创建一个绘图表面，所有的绘图操作在内存中的位图中进行，然后再将要绘制的图形显示在屏幕上。这样做的优势是当所需的图像包含有大量的基本图形时，在内存中的 BitMap 位图上进行绘图比直接在屏幕上绘图要快的很多。

在图 2 所示的窗体类的构造函数中完成三个操作：第一，实例化 XMLDB 类，调用其中的 XML Deserialize 方法完成 XML 文档的反序列化。第二，实例化 PhotoDB 类型对象，然后调用 GetDataInfos ByImageData 方法完成位图对象以及 Graphics 对象的创建。第三，在遍历 PhotoInfo[] 数组的过程中，以 PhotoInfo 对象中的图片文件路径作为参数，调用 PhotoDB 类型对象的 GetImage 方法，获取相应的图片对象。这样就完成了内存位图的处理。

在图 2 所示的窗体的 Paint 事件的处理方法中完成将经过处理的内存位图显示在屏幕上的功能。同时当窗体绘制区无效时，也调用该方法进行窗体重绘。主要代码如下所示：

```
private void MainForm_Paint(object sender, PaintEventArgs e)
```

```

{
    Graphics graphics;
    Public PhotoInfo[] pif=null;
    Rectangle rect=new Rectangle(0, 0, 128, 128);
    Font font=new Font("宋体", 16, FontStyle.Bold);
    StringFormat sf=new StringFormat();
    sf.Alignmen=StringAlignment.Center;
    sf.LineAlignment=StringAlignment.Center;
    //定义白色画笔绘制背景
    graphics.FillRectangle(new SolidBrush(Color.White),0,0,this.Width, this.Height);
    //写标题文本
    string strInfo=pif[GetIndex(PhotoDB.IndexType.Current)].Text;graphics.DrawString(strInfo,font,new SolidBrush(Color.DarkBlue), new RectangleF(0,10,320,20), sf);
    //绘制图 2 所示界面中的中心图片
    graphics.FillRectangle(new SolidBrush(Color.Black),new Rectangle(93, 53, 128, 128));
    graphics.DrawImage(pif[GetIndex(PhotoDB.IndexType.Current)].bmp, 90, 50);
    //绘制图 2 所示界面中的左边图片
    graphics.FillRectangle(new SolidBrush (Color.Black), new Rectangle(-7,77,90,80));
    graphics.DrawImage(pif[GetIndex(PhotoDB.IndexType.Front)].bmp, new Rectangle(-10, 74, 90, 80), rect, GraphicsUnit.Pixel);
    //绘制图 2 所示界面中的右边图片
    graphics.FillRectangle (new SolidBrush(Color.Black), new Rectangle(232, 77, 95, 80));
    graphics.DrawImage(dif[GetIndex(PhotoDB.IndexType.Next)].bmp, new Rectangle(228, 74, 95, 80), rect, GraphicsUnit.Pixel);
    //利用 e.Graphics 生成 Graphics 对象将内存中绘制完成的位图显示在屏幕上
    e.Graphics.DrawImage(bitmap, 0, 0);
}

```

#### 3.2 图片添加功能的实现

在电子相册图片添加的实现中，对 XML 文档中数据的访问和存取是通过 DOM(Document Object

Model)提供的编程接口实现的, DOM 是将所有信息写成树状的结构以文档形式储存起来的<sup>[12,13]</sup>. .NET 平台下提供了一些类对 XML 文档进行遍历和解析, 包括 XmlDocument 类、XmlNode 类和 XmlElement 类等, 所有这些类都包含在命名空间 System.Xml 下. 本文在电子相册制作中主要使用了 XmlDocument 类和 XmlElement 类. XmlDocument 对象表示整个 XML 文档, 它使用 Load()方法加载 XML 文档, 它的属性 DocumentElement 用于获取 XML 文件的根节点. XmlElement 对象表示 XML 中的元素节点, 其 ChildNodes 属性用于获取当前节点的所有子节点, Name 属性用于获取当前节点的名字, InnerText 属性用于获取当前节点的值, CreateElement 方法用于创建一个节点, AppendChild 方法用于将指定的节点添加到当前元素的子节点列表的末尾, Save 方法用于保存 XML 文档.

电子相册添加图片在进行 XML 解析时采用多重循环嵌套多分支语句. 关键代码如下:

```
//创建 XmlDocument 对象
XmlDocument Xmldoc = new XmlDocument();
//使用 Load 方法读取 XML 文件
Xmldoc.Load(@"\XML 文件所在的路径");
//获取根节点
XmlElement topelement = Xmldoc. Document
Element;
//循环遍历根节点的子节点
foreach (XmlElement element in topelement.
ChildNodes )
{
    bool flagSmall = false;
    if(flagBig)
    {
        break;
    }
    if(element .Name == "PhotoInfo")
    { //循环遍历 PhotoInfo 节点的子节点
        foreach (XmlElement nodelist in element.
ChildNodes )
        {
            switch (nodelist.Name)
            {
```

```
//修改 PhotoInfo 下子节点 Count 的
case "Count":
    nodelist.InnerText=(int. Parse (nodelist.
InneText) +1).ToString ();
    break ;
    .....
//创建 SmallImageTitleList 节点的子节点
string
case " SmallImageTitleList ":
    XmlElement elem2=Xmldoc. Create
Element ("string");
//为 string 节点添加标题文本
elem2.InnerText=txtTitle .Text .Trim ();
//将 string 节点添加到 SmallImageTitleList
节点中
nodelist.AppendChild (elem2 );
flagBig=true;
break ;
}
}
}
}
Xmldoc.Save (filename);
MessageBox.Show("新增成功!");
```

#### 4 结语

本文利用 XML 数据存储和反序列化机制以及 .Net Compact Framework 提供的 GDI+类库中的方法和属性在基于 Windows CE 6.0 的嵌入式移动设备上实现了电子相册系统, 解决了传统相册不易保存、浏览不方便等问题. XML 的平台独立性、简洁、基于文本、自描述格式等优点解决了嵌入式应用软件开发中多样的软硬件平台下数据存储与交换的难题. GDI+使得嵌入式开发人员在编写上层应用程序时, 无需直接跟显示设备的驱动程序打交道, 提高了嵌入式应用软件开发效率.

#### 参考文献

- 1 Bray T, Paoli J, Sperberg-McQueen CM, et al. Extensible Markup Language(XML)1.0(Third Edition), W3C Recommendation 04 February 2004.

(下转第 68 页)

贵,软件模块可以通过插件的形式增强软件功能;⑥ Mantis 支持地形数据库,支持场景管理。

### 3.4 原型系统

飞参数据驱动的实兵演练回放系统的主要硬件配置:二维态势计算机为酷睿双核 2.8GHzCPU, 2GB 内存, GTX560 显卡, 三维场景工作站为 Intel Xeon 5506 四核 CPU, 4GB 内存, GTX560 显卡, Quadro FX5800 显卡。软件平台为 Windows XP 操作系统, 软件开发平台为 Microsoft Visual Studio2010、MapX7.0 及 Mantis2.3.0。系统运行的二维态势和三维场景如图 4 和 5 所示。



图4 二维态势



图5 三维场景

## 4 结论

实兵演练回放系统利用从飞参时间序列数据中提取的飞机位置和姿态数据作驱动,在基于 DEM 三维地形表示和真实遥感卫星影像数据配准的前提下,以 Creator 与 Terra Vista 为基础工具,生成飞机模型和地景模型,以 Mantis 作为开发平台,构造大范围地形的视景仿真三维虚拟环境,采用 MapX 组件与面向对象

可视化编程语言集成的二次开发模式实现二维态势显示,对实兵演练的飞参数据从不同视角进行可视化诠释。系统场景逼真,信息丰富,基于事实,为部队和机关对实兵演练进行讲评或者研究新战法提供了一个新的平台。

### 参考文献

- 1 董梅,董伟凡,韩连平.飞行再现系统的实现及其在飞行参数处理中的应用.计算机仿真,2002,19(1):63-65.
- 2 付战平,邸亚洲,尚希良,邱智.基于 OpenGL 的飞行过程再现与仿真.系统仿真学报,2002,14(9):1197-1199.
- 3 倪世宏,张瑞峰,史忠科,谢川,王彦鸿.基于 DirectX 技术的飞机飞行过程再现.计算机工程,2004,30(24):131-133.
- 4 高辉,张茂军.基于真实地景的三维飞行再现系统的设计与实现.计算机工程,2006,32(12):232-240.
- 5 王炜,刘少华,高辉,徐玮.飞参驱动的虚拟飞行再现系统设计及实现.系统仿真学报,2007,19(9):2000-2007.
- 6 徐瑞,黄兆东,阎凤玉.MATLAB2007 科学计算与工程分析.北京:科学出版社,2008.181-188.
- 7 MultiGen-Paradigm inc. MultiGen Creator User's Guide (Version 4.1). MultiGenn-Paradigm Inc., 2001.
- 8 张尚弘,张超,郑钧,王兴奎.基于 Terra Vista 的流域地形三维建模方法.水力发电学报,2006,25(3):36-39.
- 9 DMSO.Mantis User Manual. Quantum3D, Inc. http://www.quantum3d.com.2006.

(上接第 165 页)

- 2 Goldfarb CF.张晓晖译.XML 手册.第 4 版.北京:电子工业出版社,2003.5-10.
- 3 Birbect M.斐剑锋译.XML 高级编程.第 2 版.北京:机械工业出版社,2002.15-18.
- 4 Rusty E, Menas HWS.孔小玲,商艳莉译.XML 技术手册.北京:中国电力出版社,2002.25-32.
- 5 陈海燕.XML 解析技术在影票销售系统中的应用.濮阳职业技术学院学报,2011,24(1):150-151.
- 6 刘建芳,夏栋梁.XML 序列化机制在客户关系管理中的应用.电脑开发与应用,2011,24(2):31-37.
- 7 鹿庆,何文雪,程彬.GDI+在轨道交通综合监控系统软件中的应用.计算机安全,2010,(1):9-12.
- 8 刘燕丽.GDI+在图形学教学中的研究与应用.软件导刊,2011,10(4):193-194.
- 9 王浩.基于 XML 技术在 WINCE 模拟器实现火车时刻表查询.滁州学院学报,2009,11(6):106-107.
- 10 尉辉根,李家树.基于 GDI+技术的高质量特殊统计图设计与实现.电脑与电信,2012(4):57-59.
- 11 陈忠睿,刘齐宏.NET Framework 中 XML 序列化研究.郑州轻工业学院学报(自然科学版),2005,20(2):94-95
- 12 周长敏.基于 XML 的网络考试系统的研究与实现.贵阳:贵州师范大学,2008.
- 13 杜源,张永安,周剑超.XML 解析技术在 IPTV 机顶盒中的应用.计算机系统应用,2012,21(6):197-199.