

跨平台 Java 多媒体显示系统创建研究

单 振,舒 平

(东南大学 计算机学院,江苏 南京 210096)

摘 要:随着 LED 大屏及等离子大屏技术的成熟,大屏幕多媒体显示在各种场合被越来越多地使用。虽然已经有很多大屏显示系统被开发,但现时并没有太多使用 Java 开发的 LED 大屏多媒体播放系统,它们一般都存在无法跨平台使用的问题。简介了一个使用 Java 开发的具有跨平台性能的多媒体播放系统,图示了整个系统的基本构架;并对其中的一些关键技术,如:屏幕分割、分层显示、系统对事件监听的处理、文字处理框、数据库连接及同步、表格显示等进行了阐述。

关键词:大屏显示;屏幕分割;事件监听;分层显示;数据库连接及同步

中图分类号:TN919.85

文献标识码:A

文章编号:1673-629X(2007)08-0212-03

Research on Construction of Platform Independent Multimedia System Based on Java

SHAN Zhen, SHU Ping

(School of Computer Science & Engineering, Southeast University, Nanjing 210096, China)

Abstract: With development of LED and plasma display unit, more and more large screens are used in media. Though a lot of systems were developed for them, few of them were developed with Java and so suffered from the problem that they can't be used in different platform. This text introduced a new media show system that was developed with Java. It visualizes the main structure and also outlines some key technology such as: screen cut, multi-layer show, system events listening and texture editing, database connection and table.

Key words: large screen show; screen separation; event listening; hierarchical display; database connection and synchronization

0 引 言

所谓多媒体,就是将文字、图像、视频等连接起来,完成表示信息和传播信息的载体。现在的多媒体显示方面,在多种媒体的组合和应用方面,做得比较好的是 Macromedia 公司的 Flash 动画,它成功地将文字、各种格式的图像和声音等组合起来,并可以对其定义多种动作和处理,甚至可以编写脚本程序来简化定义,或完成各种输入信息地处理。而且,Flash 还很好地处理了最后文件的压缩处理问题。但是 Flash 也有一些问题,比如不能很方便地在 Linux 平台上运行,同时它只是一个小程序,不能进行远端控制和更新,不能连接数据库等。

1 多媒体显示设计概述

2005 年中国银行苏州分行开始在各处安装等离

子大屏作为显示各种银行信息之用。为此需要一个全新的大屏显示系统,其媒体显示的基本要求是要处理带有图片背景的多媒体播放,需要诸如滚动字幕和活动广告图片的显示、根据最新数据库信息进行的动态利率显示、各种广告视频的播放等。同时,由于基于 Windows 并不是一种开源免费平台,为了降低开发和应用的成本,需要该系统具有跨平台开发和处理的能力,以便信息中心可以在 Windows 平台上进行多媒体编辑,然后再在 Linux 平台上(大屏端)进行播放。

在之前类似的多媒体系统,已经由多家公司进行了开发,如常州金石开公司等。但是这些系统都是使用 C 或者 Basic 等语言开发的 Windows 平台专用系统,不具备跨平台性。为了使系统具有跨平台性,一般有两种方法:使用一般编程语言编写,然后经过修改,在不同平台下进行编译,使用这种方法可以找到很多已有的成熟控件,开发十分便捷,但只可以使相同的程序在不同的系统上运行,而跨平台之间程序通信、调用,和文件的传输、格式处理等问题很难解决。第二种方法就是使用 Java 这种具有跨平台性能的编程语言进行编写,虽然没有现成的控件可以使用,但是只要开

收稿日期:2006-10-12

作者简介:单 振(1981-),男,江苏苏州人,硕士研究生,研究方向为计算机图像、图形学等;舒 平,副教授,硕士生导师,研究方向为计算机软件理论、计算机操作系统、计算机图形学等。

发成功,便可以很简单地解决所有跨平台问题。

为了达到上述的各种基本要求,本系统的主要问题有如下几个:文字、图片以及声音和视频的显示和播放问题;多媒体显示的层次问题;文字、图片、视频等的各种动作的定义和实现问题;序列化问题,即如何将新编辑的多媒体文件保存下来并通过网络传播;数据库访问、显示及其容错问题。

2 多媒体显示设计的关键技术及实现

2.1 总体构架

Java 作为一种面向对象的语言,其面向对象的思想给了很好的开端,本系统使用基于 MVC 思想的树型的分层构架,其基本构架如图 1 所示。

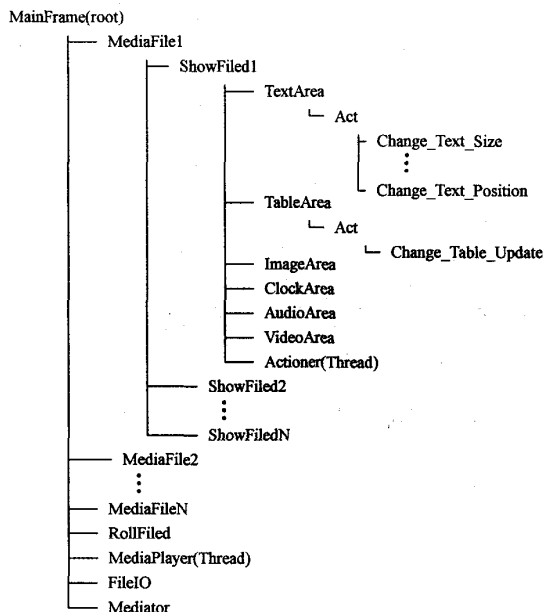


图 1 基本系统架构图

一方面这样的实现便于代码的编写,另一方面,由于这样的结构在编辑新的多媒体时更容易方便地显示整个文件的结构,方便使用者修改。下面简单介绍该系统的构架、组成及主要媒体的处理类。

2.2 媒体页 MediaFile

本系统采用了类似网页的概念,一个多媒体文件,一般应该由多个子文件构成,比如在银行系统中,可能要显示利率、广告等,这些信息不可能同时显示在一页上,所以必须分开显示,每一页可以设定显示时间,自动轮换;而同时,由于为了满足让客户自主地翻看各种信息,也必须将不同类型的信息显示分开处理并保存,这样一个子文件概念类便由 MediaFile 来实现。同时主窗口生成一个 MediaPlayer 线程,用来实现各个子文件之间的自动切换处理。

2.3 屏幕分割类 ShowArea

(1) 使用多线程支持动作处理。

考虑到在大屏显示时常常会将屏幕进行分割显示,比如在屏幕的右上方显示时间,下方显示滚动字幕,这个屏幕分割的任务,由 ShowArea 来实现,它是一个 JPanel 类的派生类^[1],可以在 ShowArea 区域放置各种实际的媒体显示类(如 TextArea, TableArea 等)。虽然多线程已经成了现代编程的流行方式,每个子媒体显示类可以定义各自的动作和设置,然后由各自产生线程来负责处理,但是考虑到本系统在一个单一页面上可以定义多个媒体显示类,这样会构成十几个甚至几十个线程。如果一个系统有太多的线程也是不太合适的,除了资源开销外,还会有诸如数据同步等多种问题,所以,本系统决定将动作的处理由 ShowArea 类生成单一线程集中处理。其所有的子类,拥有一个指向所属 ShowArea 类的引用,每当新建一个动作,便通过该引用向其父类进行注册。父类计算所有这些动作的总起始和结束时间,使用单一线程通过一个 while 循环来处理所有这些动作的执行。

(2) 显示的分层实现。

ShowArea 类另一个需要处理的问题是媒体类的分层显示。本系统使用了 jbuilder2005 的 XYLayout 类作为布局控制,由于在 java(1.4) 的 Panel 类中的布局中不存在分层显示的概念^[2],所以必须开发代码来实现,本系统采用了重写 panel 类的 add 方法类来实现,将每一层所有的 panel 保存在单独的矢量(Vector 类)数组中($n_1 \sim n_9$),并在每次调用 add 类时,ShowArea 类首先查询新加入媒体类子窗口的所属的层数 n_1 , 以及其坐标和大小,然后查询将层数 $n_2 = n_1 + 1$ 的矢量,对其中保存的每个媒体类子窗口,根据其坐标及大小,确定其与新添加在窗口是否有覆盖关系,如果有则将其保存在一个单独的矢量中,并从 ShowArea 中移除,之后,调用 Panel 类的 add 方法将新添的媒体类窗口加入 ShowArea 中,然后,再将刚移除的媒体类、子窗口重新加回 ShowArea 中。值得注意的是,如果是要在最底层 n_1 加入新的媒体类,那么该方法会引起一系列的计算(从 n_1 层到 n_9 层的一系列添加和删除)。暂时没有其它更好的方法。所幸,一般底层都可作为静态背景使用,调用一次 add 后一般不会再次调用。

2.4 使用 JMF 实现音频和视频的播放

VideoArea 和 AudioArea 分别是处理视频和音频的类,它们只是简单地应用了 JMF, Java 多媒体框架(JMF)中包含了许多用于处理多媒体的 API。它是一个相当复杂的系统,完全了解这个系统可能需要花上几周的时间,但是对于一般应用而言只需知道几个核

心接口和类,现在有很多介绍这方面的文章,只要稍微看一下,便能用很少的代码编写出很好的多媒体组件,本系统只是简单进行了应用,借鉴一个网上现成的媒体播放组件,在其中新添了自己的鼠标右键弹出式菜单,通过新添鼠标事件的监听,来实现通过鼠标拖动进行视频窗口的缩放功能^[3]。

2.5 文字处理类 TextArea

TextArea 类作为文字显示类,在 Java 中已有现成的类,但不能对其进行鼠标拖动,相应方便的是使用图形界面进行文字字体格式的设置。为此,在通过继承在 Java 已有的 JTextField 类上进行改进,新添了对鼠标的拖动以及鼠标右键弹出式选单的监听,可以对文本区域进行缩放和移位;由于 Java 不支持鼠标双击事件,所以通过一个变量,旧的记录鼠标单击事件发生的时间,并于新事件发生的时间进行比较来确定是否发生了双击事件^[4]。如果发生了双击,则开始进入编辑模式,激活主窗口上的文字格式显示器,并将一个自身的引用传给它,使其对该编辑区域的文字设定字体格式等,同时通过去除鼠标事件的监听,以关闭文本区域鼠标拖动的相应位置,使该文本区处于固定状态;通过新添对焦点事件(focus-Listener)的监听,在编辑完毕后,只需在文本区域外任意点击,使该文本区域失去焦点,引起一个事件响应,重新将鼠标拖动以及进行鼠标右键弹出式选单的监听,并通知主窗口文字格式显示器,使其失效,便可取消编辑模式,进入一般模式。

2.6 数据库表格处理显示类 TableArea

Java 本身具有一个处理数表格的类 JTable,但由于其显示格式单调,而银行不但要显示数值数据,还可以根据数值进行绘图,所以,通过重写 paint 函数自行绘制表格和图形。另一方面,由于涉及数据库的远程访问,可能会有各种情况(如远程数据库不可访问等),也必须添加代码来处理,由于 Java 有很好的违例差错控制,只需要简单地在数据库访问代码的 catch 中添加处理方法即可^[5]。一般有两种处理方法:立即停止数据的显示,或先使用旧的数据,并在屏幕上写字提醒。

远程数据库访问失效,一般有两种情况:远程数据库故障和网络故障。通过一个简单的网络 Socket 函数,可以将故障发送到服务器端,如果不是网络故障,

服务器端收到报告后,回一个消息给客户端,决定是否停止该表格数据的显示。

2.7 媒体文件的序列化

由于这个多媒体文件是由一系列的文件集合而成,所以其保存比较麻烦,笔者采用将所有的动作定义和文本文件序列化成单独的文件,并将其和所有的文件(图片、音乐和视频)全部打包成一个 zip 文件,然后自动通过网络发送到每一个在线的客户端,再由客户端解包并装载。由于压缩和解压缩需要大量计算,所有如果一个多媒体片断由多个视频文件组成的话,保存和打开文件可能会花一定的时间^[2]。

2.8 最后实现

在服务器编辑模式下,系统运行的大致窗口为:



图 2 基本运行外貌

3 总 结

Java 是一个功能强大的编程工具,但是现今,使用 Java 创建多媒体系统还只是一个开始,仍然有许多值得研究的地方。本系统使用 Java 编程语言,应用分层构架模式,实现了自动网络文件传输、屏幕分割、各种媒体信息的分层显示、数据库表格处理,构建了一整套文字处理控件、实现图片、音乐和视频整合编辑功能,以及实现了对所有媒体进行各种动作的设定,并以多线程的方法进行了实现。

参考文献:

- [1] Eckel B. Thinking in Java[M]. 2nd Edition. New York: Pearson Education, 2001.
- [2] 吴益华,陈志远. Java 技术手册[M]. 北京:中国电力出版社, 2002.

```

}
//TODO: Add your control notification handler code here
Engine * ep;
if(! (ep = engOpen(NULL))) //
{
    MessageBox("Can't start Matlab engine");
    return;
}

engEvalString(ep, "load tire;");
engEvalString(ep, "nbc = size(map,1);");
engEvalString(ep, "wname = 'haar'; lev = 3;");
//wavedec2 多尺度二维小波分解
engEvalString(ep, "[c,s] = wavedec2(X,lev,wname);");
engEvalString(ep, "alpha = 1.5; m = 3.5 * prod(s(1,:));");
//wdcbm2 获取压缩阈值
engEvalString(ep, "[thr,nkeep] = wdcbm2(c,s,alpha,m);");
//wdencmp 用小波进行信号压缩;
engEvalString(ep, "xd = wdencmp('lvd', c, s, wname, lev, thr, 'h');");
engEvalString(ep, "colormap(pink(nbc));");
engEvalString(ep, "figure(1);");
engEvalString(ep, "subplot(121);");
engEvalString(ep, "image(wcodemat(X,nbc));");
engEvalString(ep, "title('原始图像');");
engEvalString(ep, "subplot(122);");
engEvalString(ep, "image(wcodemat(xd,nbc));");
engEvalString(ep, "title('压缩图像');");
//engClose(ep);
}

```

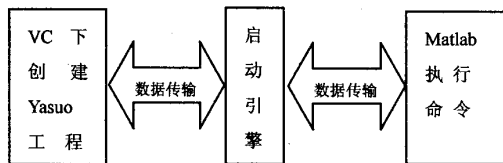


图 1 工程执行流程图

其中, engOpen 函数声明为:

Engine * engOpen(const char * startcmd); 参数 startcmd 设为 NULL。

实现向 Matlab 发送命令的 engEvalString 函数, 函数声明为:

int engEvalString (Engine * ep, const char * string);

engClose 函数声明为:

int engClose(Engine * ep)。

在上述程序中若启用 engClose 函数, 由于该函数在程序执行完成后迅速释放资源, Matlab 窗口显示的图像压缩处理结果将会马上消失。因此, 为了更好地观察程序运行结果, 在程序执行前可以先注释掉该行代码。

程序运行结果如图 2、图 3 所示。

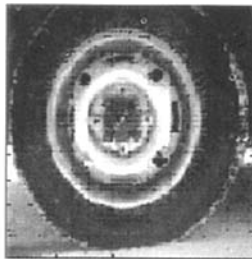


图 2 原始图像



图 3 压缩图像

5 结 论

以图像的压缩处理为例, 在 VC 环境下利用 Matlab 引擎实现 Matlab 与 VC++ 接口完成图像压缩处理, 充分发挥了 Matlab 在图像处理上的技术优势, 简化了算法, 以精练的代码达到了图像处理的预期效果, 同时节省大量的系统资源。该方法弥补了用单个软件开发出来的应用程序在功能上的不足, 同时也可以为熟悉 VC++ 语言的工程技术人员在开发实用图像处理应用程序时提供方便, 是提高程序开发效率的一种常用手段, 具有非常广阔的应用前景。

参考文献:

- [1] 王周益, 刘继兴, 柳长安. VC++ 与 MATLAB 混合编程研究及开发实例[J]. 计算机应用研究, 2006(5): 154-155.
- [2] 刘志俭. Matlab 应用程序接口用户指南[M]. 北京: 科学出版社, 2000.
- [3] 任 哲. MFC Windows 应用程序设计[M]. 北京: 清华大学出版社, 2005.
- [4] 刘志俭, 潘献飞. Matlab 外部程序接口(6. X)[M]. 北京: 科学出版社, 2002.
- [5] 张兆礼. 现代图像处理与 MATLAB 实现[M]. 北京: 人民邮电出版社, 2001.
- [6] 周 伟. Matlab 小波分析高级技术[M]. 西安: 西安电子科技大学出版社, 2006.

(上接第 214 页)

- [3] 原 野, 沈钧毅, 邢东山. 基于 Java 与 JMF 的在线媒体教室系统[J]. 计算机工程, 2002(3): 8-9.
- [4] 邓光伟, 李丙午, 朱 志, 等. Java 2 核心技术(卷 2: 高级特性)[M]. 北京: 机械工业出版社, 2001.

- [5] 李焱明, 莫 倩, 徐 明. 基于 Java 技术的 Web 环境下分布式数据库互操作性的实现[J]. 计算机系统应用, 1998, (12): 131-150.