

使用 XML 实现 JAVA 软件界面维护

陈胜凯, 胡众义, 徐 晓

(温州大学信息科学与工程学院, 浙江温州 325035)

摘 要: 本文介绍了如何通过 XML 配置文件实现 JAVA 软件的界面维护。通过 JNI 技术, 程序可以自动检测当前屏幕的分辨率, 并利用 JDOM API 读取相应的 XML 文件设置, 实现软件界面上控件的精确定位。

与传统的通过布局管理器管理 JAVA 软件界面方式相比, 该方法具有定位简便精确、便于维护的优点。

关键词: 界面维护; JAVA 文档对象模型; JAVA 本地接口

中图分类号: TP391 **文献标识码:** TP **文章编号:** 1008-309(2004)06-0052-12

一、引言

在传统的 JAVA 程序中, 通常采用布局管理器管理程序界面。有 BorderLayout、GridBagLayout 等多种布局管理器可用于负责排列构件容器上的控件, 但它们的使用都比较复杂、难以精确定位, 并且与其他语言的界面编程方式很不一致, 不符合传统程序员的思维。为此, JAVA 引入了一种不使用布局管理器进行界面设计的方式, 即在使用 setLayout 函数设置布局管理器时使用 null 作为参数。但在这种方式下, 程序员必须手工确定界面上各个控件的位置, 并且当程序运行于不同的环境, 比如 WINDOWS 平台的不同分辨率下时无法自动调整控件位置。为解决这一缺点, 我们引入了 XML 配置文件, 在该文件中指定各种控件的类型和不同分辨率下的位置, 然后在 JAVA 程序中读取配置进行界面构造。由于 XML 文档能简单地通过记事本等工具进行修改, 所以即使在程序发布后, 用户也可以根据具体需要修改控件的类型和位置, 进行界面维护。界面初始化的处理过程流程图如下:

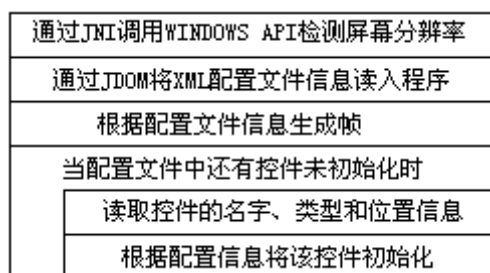


图 1 界面初始化处理流程

二、XML 配置文件的建立

XML (Extensible Markup Language) 是新兴的技术, 它已经广泛用于信息技术的数据交换、应用集成等领域。XML 是一门用来定义其他语言的元语言, 它既无标签集也无语法, 具有很好的可扩展性和自描述性^[2]。因为 XML 提供了一种非常简易的数据描述方式, 所以它很适合用来编写配置文件。配置文件可以使用 notepad 等文本编辑器建立。

收稿日期: 2004-08-15

作者简介: 陈胜凯(1977-), 男, 浙江温州人, 助教, 学士, 研究方向: 网络与数据库应用

三、使用 JNI 获取屏幕分辨率

为了使程序能够确定使用 config.xml 文件中的何种配置, JAVA 程序必须能够获取当前所处环境的屏幕分辨率。由于 JAVA 语言并不提供直接读取分辨率的函数, 因此我们使用其提供的 Java Native 接口 (JNI) 技术, 并调用 WINDOWS API 实现该功能。

JNI 定义了一个标准的命名和调用的准则, 使 Java 程序可以调用本机上的库^[3]。这意味着可以用 C 或其他语言编写系统相关代码并在 Java 中使用。在下面的例子中, 我们构造一个 TestLayout 类, 该类将通过 JNI 装载本地 TestLayout 动态链接库, 并调用库中的函数 GetDisplay 获取屏幕分辨率函数。GetDisplay 函数将通过调用 WINDOWS API 函数 GetSystemMetrics 返回当前分辨率大小。整个过程分为四个步骤, 分别如下:

(一) Java 程序中动态链接库的装载

首先生成并编译主调的 Java 类, TestLayout 类的源代码如下:

```
public class TestLayout {
    public native int GetDisplay();
    static { System.loadLibrary("TestLayout"); }
}
```

在该类的 GetDisplay 函数声明中使用了 native 保留字, 它表明该函数是一个本地方法, 包含于本地库中。通过 System.loadLibrary 函数加载名为 TestLayout 的动态链接库。将该调用放在静态初始化函数中可以确保每个类只加载一次这个库。

(二) 使用 javah.exe 生成头文件

创建生成动态链接库所需的 C 头文件, 它用来定义本地函数说明。使用 JDK 提供的工具 javah 可以完成这步, 但要确保该 javah 的 JDK 版本与步骤一中编译 TestLayout 类的 javac 一致。在命令行运行 javah TestLayout 即可生成头文件 TestLayout.h。

(三) 实现本地方法

编写相应 C 函数实现时, 要注意保持与头文件中的声明一致, 实现函数 GetDisplay 的代码如下:

```
#include "jni.h"
#include "TestLayout.h"
#include <windows.h> // 包含 Windows API 库声明文件
JNIEXPORT jint JNICALL Java_TestLayout_GetDisplay(JNIEnv *env, jobject obj)
{
    jint i=GetSystemMetrics(SM_CXSCREEN);
    return i; }
}
```

首先注意函数的名称, 我们在 JAVA 程序中将通过 GetDisplay 调用函数, 但在 C 实现中需要加上 Java 以及主调函数所在类的类名并用符号_分隔。其次是被返回的参数数据类型, JNI 提供了 JAVA 程序到本地方法之间的数据类型映射, 比如 JAVA 中的 boolean 型在本地方法中映射为 jboolean, 具体的类型映射表可以参考 sun 公司的在线文档。由于例子中的 GetDisplay 函数需要返回一个 int 基本类型, 所以在 C 实现中声明返回的类型为 jint。最后是 GetSystemMetrics 函数的使用。该函数属于 WINDOWS API, 可以用于读取多种系统配置设置值。当函数带有参数 SM_CXSCREEN 时, 返回值为屏幕的整型宽度像素值。

(五) 编译动态链接库

该步骤可以使用大多数 C 或 C++ 编译器, 我们采用 VC 进行编译。在使用 VC 创建动态链接库时, 需要在工程设置的资源选项卡中指明 Java_Home\include 和 Java_Home\include\win32 两个包含路径, 因为编译时将使用到其下的 jni.h 和 jni_md.h 两个头文件。生成动态连接库 TestLayout.dll 后, 将其放到 java 程序目录下就可以调用了。

四、使用 JDOM 读取配置并初始化

为了读取 XML 配置文件，通常在程序中我们需要使用 XML 解析器。程序使用解析器 API 中定义的方法调用接收从 XML 文档中读取的信息并检查其形式合理性。有多种解析器可供选择，比如 SAX 和 DOM。但由于我们是在 JAVA 程序中读取配置，因此 JDOM API 似乎更能满足需求。JDOM 是一个开放源码处理 XML 的 JAVA API，它允许你在不利用 DOM 特性的条件下以树形格式处理 XML 文档。同时，它具有 SAX 的高效性，能够实现快速解析。该 API 包含在 org.jdom 包中，我们可以从 <http://www.jdom.org> 下载到该包的最新版本。和 DOM 一样，JDOM 将 XML 文档表示为树，包括其中的元素、属性等；与 SAX 不同的是，它可以随时访问树中的任何部分^[1]。JDOM 本身不包含分析器，而是依赖于 SAX 分析器，因此必须保证使用的 JDK 版本中包含了 SAX 解析器。

（一）使用 JDOM 读取文档中的配置

首先需要使用 org.jdom.input.SAXBuilder 类从存在的 XML 输入源中创建 JDOM 文档。其关键代码如下：

```
try { SAXBuilder builder=new SAXBuilder();
Document doc=builder.build(new File("config.xml"));
}catch(JDOMException jdome) {System.out.println(jdome.getMessage());}
catch(IOException ioe) {System.out.println(ioe.getMessage());}
```

上述代码使用 SAXBuilder 构造函数构造了 SAXBuilder 对象，然后使用该对象的 build 方法读取配置文件并在内存中建立 Document 型对象 doc。该方法如果在读取文档时遇到问题，将抛出 IOException 对象；如果建立文档时遇到问题，将抛出 JDOMException 对象。

有了 Document 对象后，可以使用该对象的 getRootElement 方法得到配置文件中的根元素 root，元素类型为 Element。然后使用根元素的 getChild 方法可以得到其子元素。在本例中使用 root.getChild("frame").getChild(resolution) 可以得到相应分辨率下描述帧大小的元素。这里参数 resolution 是字符串变量，由字符串 layout 和使用 JNI 技术检测到的分辨率值 800 或 1024 连接而成。最后使用 Element 对象的 getAttribute 方法得到相应的描述长宽的属性元素，并用属性元素的 getIntValue() 进行类型转化。如果转化不成功，该函数将抛出 DataConversionException 异常。该段读取帧大小设置的关键代码如下：

```
Element root=doc.getRootElement();
Element xmlframe=root.getChild("frame").getChild(resolution);
try { width=xmlframe.getAttribute("width").getIntValue();
height=xmlframe.getAttribute("height").getIntValue();
}catch(DataConversionExceptiondce){System.out.println(dce.getMessage());}
frame.setSize(width,height);//设置帧的大小
```

读取组件类型、名称和属性的方法与读取 frame 元素的相似，但由于 component 元素的个数不确定，所以使用了 java.util 包中的 List 类，通过 Element 对象的 getChildren 方法可以将配置文件中根元素下所有的 component 元素读入 List 中。

（二）使用迭代器(Iterator)进行控件初始化

迭代器是 util 包中一个接口。在 JAVA 中，任何集合对象都可以生成 Iterator 类型的对象，包括 List。List 接口的 iterator 方法可以返回一个 iterator 对象，将原始 List 中的所有对象的引用按照顺序封装起来，以供 iterator 接口方法访问。该接口的 hasNext 方法检测是否有下一个对象并返回 boolean 型结果。其 next 方法获得下一个位置的对象，返回值为 Object 类型。这两个方法结合就可以循环读入 List 中的每一个 component 元素，然后读取其类型、名称等属性，进行软件界面的控件初始化处理。其关键代码如下：

```
java.util.List components=root.getChildren("component");
Iterator iterator=components.iterator();
while (iterator.hasNext()) {
Element component=(Element)iterator.next(); //Object 类型要强制转化成 Element 型
String name=component.getAttribute("name").getValue();
String type=component.getAttribute("type").getValue();
if (type.equals("button")) //如果控件类型为按钮
{ Button b=new Button(name);
b.setBounds(positionx,positiony,width,height);
// positionx,positiony,width,height 为从 xml 中读取的配置值
frame.add(b); //将控件加入帧}
if (type.equals("textfield"))
{ 将该类型控件加入软件界面的相应位置}
进行其他类型控件的处理 }//for while
```

上述代码中, 首先使用元素属性的 `getValue` 方法获取控件的名称跟类型, 该方法由于不进行数据类型转化, 不会抛出异常。然后通过条件语句根据各种控件的情况进行相应初始化。

五、结束语

本文给出了一种通过 XML 配置文件对 JAVA 软件界面进行维护的方法。通过这种方式, 软件用户在程序发布后能根据具体情况修改界面的控件分布, 非常简便。而且具有一定的灵活性, 比如在 `TextField` 输入字段不够长时将其替换为 `TextArea` 控件。当然程序尚需进一步完善。由于其使用 `WINDOWS API` 而失去了跨平台的特性, 但这可以通过在程序中检测运行时的操作系统版本, 然后进行相应不同平台的分辨率检测的方法解决。改进后的方法可以完全实现布局管理器同样的功能, 但却不存在布局管理器使用复杂, 窗口控件定位困难的缺点。是一种较好的 JAVA 软件界面维护方案。

参考文献

- [1] [美] Brett McLaughlin(孙兆林译). *JAVA 与 XML*[M]. 北京: 中国电力出版社, 2001
- [2] [美] Elliotte Rusty Harold(刘文红译). *Java 语言与 XML 处理教程*[M]. 北京: 电子工业出版社, 2003
- [3] [美] Calvin Austin, Monica Pawlan(前导工作室译). *Java2 高级编程*[M]. 北京: 机械工业出版社, 2001
- [4] [美] Ivor Horton(马树奇, 孙坦译). *Java2 编程指南(JDK1.3 版)* [M]. 北京: 电子工业出版社, 2001

JAVA Software Interface Maintenance Using XML

CHEN Shengkai, HU Zhongyi, XU Xiao

(College of Information Science & Engineering, Wenzhou University, Wenzhou, China 325035)

Abstract: The paper introduces how to implement java software interface maintenance using xml configuration file. Using JNI technology, program can automatically detect the resolution of current screen. It can get the position of components from the xml configuration file by using Jdom api. Comparing with traditional JAVA software maintain interface using layout manager. This solution has the advantages of precisely orientation and easy maintenance.

Key words: Interface maintenance; JAVA; JNI