

JAVA 和 XML 之间的关系和作用

新茜 (首都经济贸易大学信息经济系 100026)

摘要: 在目前的 INTERNET 上, JAVA 和 XML 的应用都在以前所未有的速度递增。本文主要讨论它们之间的关系和作用。

关键词: JAVA XML DOM API SAX 解析器

Internet 上 XML 的应用正飞速发展着, 它的作用不仅在于可能替代 HTML, 更重要的体现于数据交换的格式上。XML 在把数据从一个数据库转到另一个数据库的时候发挥重要作用, 因此对于 JAVA 程序员来说, XML 是一项非常值得关注的技术。

自 XML 问世以来它就和 JAVA 有着紧密的联系。XML 可以使 JAVA 做更多的事情, 反过来 JAVA 可以使 XML 用处更大。XML 自身只是一种文档格式, 你必须编写程序来操纵它, JAVA 可以实现这一点。下面从 XML 和 JAVA 的设计思想和体系结构两方面来予以说明。

1 XML 和 JAVA 的设计思想及体系结构

1.1 XML 和 JAVA 的设计思想

XML 和 JAVA 从起源和设计思想来看有两个相似之处:

(1) 他们都是明确设计来用在分布式系统中的。XML 在开始就是被设计用在 WEB 上的, JAVA 开始的设计思想是通过网络来发布程序, 是一种跨平台的语言, 它的这一特征对 WEB 也很适用。

(2) 这两种语言都是从复杂的难于学习的语言中简化而来的。其中

JAVA 是从 C++ 发展而来的。由于 C++ 保持并继承了 C 的某些特征, 因此并没有实现完全的面向对象, 并且过于复杂, 难于掌握。而 XML 是从 SGML 发展而来, 它是 SGML 的一个子集。而 SGML 语言过于庞大, 复杂。所以 XML 和 JAVA 都是精炼过的语言。

1.2 JAVA 的体系结构

除设计思想之外, JAVA 的体系结构的许多特征也使它很适合作为 XML 的编程语言。

(1) 支持 UNICODE。许多语言都只是支持 ASCII 码, 同 ASCII 相比, UNICODE 更象是真正的字母表, 它包含 39, 000 个字符, 并且还可扩展。目前只有 JAVA 从底层支持 UNICODE, 而 Perl 等语言主要适用于 ASCII 码, 对 UNICODE 不太适合。

(2) 包结构。包的结构使 JAVA 很容易和别的语言共享代码, 所有 JAVA 的类都适合 UNIX 和 WINDOWS 的文件系统的结构。JAVA 中的类 com.myPackage.MyClass 就是文件 com/myPackage/MyClass.java。如果你掌握了这种结构并映射到你的文件系统, 你就可以直接利用第三方的代码, 用 JAVA 编译器来编译它。

(3) 动态类载入。动态载入可以

使 JAVA 类在被请求的时候才被装入。它有两种

实现方法: 一种是在类被第一次请求时由 JAVA 虚拟机隐式的载入。另一种方法是由类方法 Class.forName() 方法显式的调用。

(4) JAVABEANS 技术。每一个 JAVABEAN 都封装了一系列可读写的属性。BEANS 在描述 XML 时特别有用, 因为他们有一个很相似的数据模型, 并可被继承, 子类可以在基类的基础上添加新的属性。这种特征使它能够很容易的表达 XML 所包含的内容。

尽管 XML 是被设计成完全独立于某种特定语言的, 但是 JAVA 的这些特征使它成为 XML 的很好的选择, 使 JAVA 和 XML 很自然的可以在一起使用。在 JAVA 编程中, 根据客户端接收到的 XML 消息, 应用程序可以分步下载。在 JAVA 中我们可以只下载表示当前元素的类。每当有新的元素出现时, 才去找对应的类并在需要的时候下载它。甚至可以在开始的时候我们可以不知道类的名字。例如, 在一个 Myshop.com 的站点上, 有一个名叫 PurchaseOrder 的文档, 它包含



一个 LineItem 的元素。我们将它所对应的类 LineItem 放在包 com.myshop.purchaseOrder 中, 它可以通过调用 com.myshop.com.purchaseOrder.LineItem 而被调用。当应用在文档中看到一个 LineItem 元素时, 它根据包结构和元素名产生一个字符串从 myshop.com 站点载入类 LineItem 从而产生一个 LineItem 对象。应用程序并不需要预先知道关于 LineItem 的信息。并且, 我们还可以通过映射加入一个新的文档, 而这个文档可以被看作是另一个文档的元素。

根据以上两个方面, 我们可以说, JAVA 和 XML 是一种伙伴关系。他们共同构筑未来的 INTERNET。Java 就象 INTERNET 的大脑, 而 XML 文档就象他们之间交流所用的语言。目前 SUN 公司已经将 XML 接口包含进 J2EE 中, 使 JAVA 和 XML 可以更方便的在一起使用。

2 用 JAVA 编写 XML 应用

在 JAVA 的应用中, XML 正被越来越多的用来存储数据和用于消息机制。用 JAVA 来编写 XML 应用, 主要有两种建立 XML 应用的基本方法: 树的遍历和基于事件的模型。

树的遍历:

文档被分解成一个树结构, 应用遍历树找到所要找的节点。主要标准是文档对象模型 (DOM API)。DOM API 是 W3C 的推荐书。它独立于特定语言。它相当于内存中的 XML 树。作为一个内存中的表示, XML 树包含许多节点, 文档作为根节点, 元素作为子节点, 它还可以再有元素子节点和文本节点。XML 文档中的其他组成部分, 如实体等, 可作为元素或文档的子节点。应用程序遍历这棵树, 找到各节点的属性和子元素。DOM 树作为一个 XML 文档的表示方法, 可以

不知道文档的类型而很容易的得到文档的内容。

树的遍历也被用来在一个 JAVA 类中寻找文档中出现的元素类型。随着每个元素的解析, JAVA 类产生一个实例, 形成一个包含特别信息和行为的对应于解析树的一个对象, 即 JAVABEANS。

事件流模型:

在 GUI 中, 用户输入触发各种事件, 整个应用有一个事件模型, 这个事件模型监听每个用户事件, 分析并引发相应的处理代码, 处理代码做一些计算并返回事件模型。JAVA 经历了两种事件模型:

第一种是简单的返回机制。当事件发生时, 就将一些参数传递给对应的方法并触发它。在 XML 应用中这个标准是 SAX。

另一种事件模型是在 JDK1.1 中实现的。在这种事件模型中, 关于事件的信息被封装在一个事件对象中, 一个对象作为事件源, 而这个事件有可能触发的对象, 作为一个接听器注册到 GUI 组件或适配器上。当事件发生时, 一个封装了事件信息的对象将被传递给所有注册过的接听器, 在 XML 的解析过程中, 当碰上一个开始的标志时, 一个“元素开始”的事件将被产生, 并送至接听器, 由接收器来处理事件。

当在 JAVA 中引入 XML 时, 需要解决很多问题。首先, 我们解决在 JAVA 中如何读 XML 文档的问题。我们可以用封装在 JAVA 中的一系列函数来解决这个问题。事实上, XML JAVABEANS 结构不仅使程序员写起程序来更简单而且使非编程人员也能写出 XML 应用来。

3 实现 XML 应用的步骤

不管用什么应用环境, 实现 XML

应用都需要有三个步骤:

首先, 确定将要用哪种 XML 文档。每个 XML 文档都会定义他们自己的元素集和标记。这些元素的名字, 类型和他们之间的联系被文档类型定义 (DTD) 所定义。

第二步, 决定如何来读 XML 文档。应用可以直接读 XML 文档, 即把他作为一个文件来读, 也可以用一个 XML 的解析器来读。XML 解析器对 XML 文档的内容提供了可靠的接口, 它通常将 XML 文档作为一个事件流或树结构来解析, 即上面所说的两种方法。

最后, 写出操作 XML 文档内容和结构的商业逻辑。例如, 我们希望对一个应用重新组织文档后, 另一个应用可以很容易的根据元素名 email 从文档中读出 email 地址来。

4 构造 XML 应用的基本框架

下面以一个简单的显示一个单位组织结构的程序为例子, 通过树的遍历的模型, 来建立一个 XML 的应用, 大致说明形成 XML 基本框架的过程。我们将由一个树结构来表示一个单位的内部组织关系。

XML 对于结构性数据提供了方便的显示方法。我们用 XML 来描述单位的体系结构, 这个结构首先被 DTD 文件所定义。DTD 文件如下所示:

```
<?xml encoding="US-ASCII"?>
  <!ELEMENT organization
  ((organizationemployee)*) >
  <!ATTLIST organization name
  CDATA #REQUIRED>
  <!ELEMENT employee
  (#PCDATA)>
  <!ATTLIST employee name CDATA
  #REQUIRED>
  <!ATTLIST employee email CDATA
```

#REQUIRED>

它描述了一个单位的下属单位及雇员的体系结构。单位有一个属性“名字”来标志它,每个雇员都有一个“姓名”和“电子信箱”的属性。一个单位可以有好多下属单位和雇员。下面是一个对应的XML文档:

```
<?xml version="1.0"?>
<!DOCTYPE organization SYSTEM
"organization.dtd">
<organization name="spark company">
  <organization name="sail
department">
    <employee name="lily"
url="lily@yeah.net">lily</employee>
  </organization>
    <employee name="tom"
url="tom@263.net">tom</employee>
  </organization>
```

我们用所做的应用程序来读文档,但并不是直接来读XML文档,它是通过一个代表XML文档的DOM树

来读的。DOM树包含了不同的节点,如文档,元素,属性,内容,进程,纯文本数据和注释等。

关于XML应用必须对XML文档所做的基本操作,如打开一个XML文档,产生一个DOM树,以及保存XML文档等,一般解析器已经能实现。别的操作,如遍历DOM体系,产生节点,删除节点,改变数据等已能用DOM接口来完成,这里就不再另叙。

下一步是以图表方式显示数据,编写和DOM接口的商业规则。我们可以用JAVA的Swing包里的JTree类来表示一个组织。用Swing包里的DefaultTreeModel来描述树模型。树模型包含一系列节点,其中叶子节点表示雇员,树干节点表示组织。对于每个XML文件中的组织和雇员,都会产生一个节点并放到一定的位置上。(关于一个纯Swing树的构造的程序,这里也不再另述。)

对于用Jtree来显示一个组织结

构,我们首先需要读入一个XML文档,然后用XML解析器形成一个DOM树。对于一个特定的XML文档,我们先实例化一个解析器,由解析器来读文件流,返回XML文档的DOM树,用文档的根元素来形成Jtree的TreeModule。(关于实例化解析器的程序,这里不再另述。)

然后我们就可以在此基础上增加这个组织显示程序的显示功能。比如增加,删除组织或雇员,或者组织或雇员需要增加一些新的信息等。这样一个XML的基本应用的框架就基本形成了。

综上所述,可以看到JAVA技术在编写XML应用的时候,可以有效的提高代码的重用性,缩短编程时间,并且使用已有的技术可使代码的编写更简单,从而使JAVA和XML紧密的结合在一起,共同构筑高效的互联网应用。■