

Java Plug-in 的应用技术

余俊 (青岛大学电气及自动化工程学院 266071)

摘要: Sun公司最新发布的Java Plug-in软件使得浏览器能直接用Sun公司自己的JRE来运行Web页面上的Java applets或JavaBeans组件。本文介绍了Java Plug-in的主要功能,讲述了Java Plug-in的原理,并详细地论述了如何在IE和Navigator浏览器中实现Java Plug-in。

关键词: Java Plug-in JRE applets JavaBeans HTML

1. 引言

Java Plug-in软件(以下简称Java Plug-in)是Sun公司Java软件家族的一个新产品,Sun公司于98年发布了Java Plug-in的1.1版本。在Internet/Intranet上用户使用浏览器浏览Web页面时,Java Plug-in能直接用Sun公司的JRE(Java Runtime Environment)来运行Web页面上的Java applets或JavaBeans组件,从而替代浏览器上缺省的java虚拟机。这样,企业在发布Java applets时能让applets充分利用Java平台最新的功能和特点,并保证applets可靠地、一致地运行。

Java Plug-in支持两种最流行的浏览器Internet Explorer(简称IE)和Netscape Navigator,Java Plug-in能运行在IE 3.02或更高的版本上、Navigator 3.0或更高的版本上。IE运行在Win32平台(Windows95/98和Windows NT4.0)上;Navigator不仅能运行在Win32平台上,而且能运行在Solaris平台(Solaris2.5/2.6)上。这样,Windows 95/98、Windows NT 4.0或Sun Solaris 2.5/2.6平台上的用户都能使用Java Plug-in。有了Java Plug-in,开发者可以自由地使用最新的JDK版本的所有功能和特点,在Internet/Intranet上发布100%纯Java applets。Java Plug-in是免费的,它提供给那些需要在Internet/Intranet上发布基于JDK1.1的applets的人使用。

2. Java Plug-in 的主要功能

(1)完全的JDK1.1支持。能使开发者在开发和发布Java applets时充分利用JDK1.1的特点和功能(如JavaBeans、RMI、JNI等)。

(2)完全的Java一致性支持。Java Plug-in使用最新的Sun公司自己的JRE软件,它和Java兼容包测试套件完全一致。

(3)及时升级。当Sun公司开发出新版本的JDK时,只须下载相应的Sun公司的JRE新版本,即可保证

版本的一致性。

(4)免费的公共下载和容易的安装。访问指定的Java Plug-in的Web页面,浏览器能下载和安装所有需要的文件,只需很少的人工干预。

(5)免费Java Plug-in HTML转换器。自动地修改旧的HTML文件,使新的HTML文件能使用Java Plug-in替代浏览器自身的Java虚拟机。

3. Java Plug-in 的原理

Java Plug-in并不覆盖浏览器中的Java虚拟机,它只是允许客户访问Web页面时用Sun公司的虚拟机替代浏览器本身的虚拟机。

Java Plug-in利用IE的扩展机制,使得Sun公司的JRE能运行在IE的内部。这是运用微软公司的COM/ActiveX技术实现的。使用HTML的OBJECT标记,Web页面编写者能运行ActiveX控件或COM组件作为Web页面的一部分。IE提供了精心设计的机制来下载和存储ActiveX控件。这使Web浏览器可以使用JRE而只需很少的人工干预。

Java Plug-in利用Navigator的Plug-in体系结构,使得能在Navigator内部运行JRE,就象用Plug-in运行QuickTime电影一样。使用HTML的EMBED标记,Web页面编写者能运行Plug-in作为Web页面的一部分。

Web页面编写者必须指定如何通过Java Plug-in使用JRE,因此,编写者必须修改Web页面以便调用JRE。

4. Java Plug-in 在IE 中的使用

现在,applet程序在Internet上广泛地使用。假设在HTML文件中applet被定义为下面的HTML代码1:

```
<APPLET code = "XYZApp.class" codebase = "html/"  
align = "baseline"  
width = "200" height = "200">
```

```
< PARAM NAME = " model " VALUE = " models/
HyaluronicAcid. xyz " >
```

```
No JDK 1.1 support for APPLETT!
```

```
</APPLET>
```

在上面的代码中,APPLET 标记指定有关 applet 程序的信息,PARAM 标记用于将辅助参数传递给 applet。为了要在 IE 中使用 Java Plug-in,必须使用 OBJECT 标记。下列 HTML 代码 2 是将 HTML 代码 1 从 APPLET 标记到 Java Plug-in 的 OBJECT 标记的映射。

```
< OBJECT classid = " clsid: 8AD9C840 - 044E - 11D1 -
B3E9 - 00805F499D93 "
```

```
width = "200" height = "200" align = "baseline"
```

```
codebase = "http://java.sun.com/products/plugin/1.
1/jinstall-11-win32.cab#Version=1,1,0,0" >
```

```
< PARAM NAME = "code" VALUE = "XYZApp.class" >
```

```
< PARAM NAME = "codebase" VALUE = "html/" >
```

```
< PARAM NAME = "type" VALUE = "application/x-java-
applet; version = 1.1" >
```

```
< PARAM NAME = " model " VALUE = " models/
HyaluronicAcid. xyz " >
```

```
No JDK 1.1 support for APPLETT!
```

```
</OBJECT>
```

在上面的代码中,OBJECT 标记包含了与 APPLET 标记相似的信息,这些信息用于在 IE 中启动 Java Plug-in。OBJECT 标记的 classid 内容是 Java Plug-in 的类标识符。当 IE 在 OBJECT 标记中发现这个类标识符后,会试图在浏览器中加载 Java Plug-in。

APPLET 标记中的 width、height 和 align 属性直接映射到 OBJECT 标记中同样的属性。IE 用这些格式信息定位 applet 程序的显示。由于这些格式信息没有改变,所以用 Java Plug-in 和用 APPLET 标记的 applet 程序在浏览器中的显示是一样的。

并不是所有的 APPLET 标记的属性都能被映射到 OBJECT 标记的属性。例如 APPLET 标记的 code 和 codebase 属性不能映射到 OBJECT 标记的属性。按照 HTML 规范,OBJECT 标记不存在 code 属性,所以我们把 code 属性映射到 OBJECT 标记中的 PARAM 标记。PARAM 标记用于传递参数。同样地,对于 APPLET 标记的那些不能映射到 OBJECT 标记的属性,我们把它们映射到 OBJECT 标记中的 PARAM 标记的参数上。

另外有一个特殊的属性 codebase。在 APPLET 标记中,codebase 属性代表下载附加的类文件或 JAR 文件的

URL;而在 OBJECT 标记中,codebase 属性代表下载 Java Plug-in 的 URL(只有在本机上没有发现 Java Plug-in 时才下载 Java Plug-in)。由于 codebase 属性在 APPLET 标记和 OBJECT 标记中有不同的含义,所以规定必须把 APPLET 标记的 codebase 属性映射到 OBJECT 标记中的 PARAM 标记的参数上。

在上面的例子中,APPLET 标记的 code 和 codebase 属性映射到 OBJECT 标记中的 PARAM 标记上。PARAM 标记的 code 和 codebase 参数与 APPLET 标记的 code 和 codebase 属性取值相同。Java Plug-in 读取这些参数值,就能够下载 applet 或 JavaBeans。在 OBJECT 标记中必须有 type 参数,它不映射任何 APPLET 标记的属性。type 参数标识 Java 执行程序的类型,是 applet 还是 Bean,以便让 Java Plug-in 知道如何初始化 Java 执行程序的类型。以上三个参数的 code、codebase 和 type 在 APPLET 标记的 PARAM 标记中是不存在的,它们被 Java Plug-in 指定。除了 Java Plug-in 指定的三个参数外,其余的参数与 APPLET 标记的参数相同。OBJECT 标记中的 model 参数与 APPLET 标记中的 model 参数是相同的。

在 APPLET 标记中的文本 "No JDK1.1 support for APPLETT!" 映射到 OBJECT 标记中。原先,当浏览器不支持 Java 时显示此文本;映射到 OBJECT 标记后,当浏览器不支持 OBJECT 标记时,显示此文本。

5. Java Plug-in 在 Navigator 上使用

Navigator3/4 能运行在 Windows95/98、Windows NT 4.0 或 Solaris 平台上,在 Navigator 中使用 Java Plug-in 需要用 EMBED 标记。下面 HTML 代码 3 是将 HTML 代码 1 从 APPLET 标记到 Java Plug-in 的 EMBED 标记的映射。

```
< EMBED type = " application/x - java - applet; version =
1.1 " width = "200"
```

```
height = "200" align = "baseline" code = "XYZApp.
class"
```

```
codebase = "html/" model = "models/HyaluronicAcid.
xyz"
```

```
pluginspage = "http://java.sun.com/products/plugin/
1.1/plugin-install.html" >
```

```
< NOEMBED >
```

```
No JDK 1.1 support for APPLETT!
```

```
</NOEMBED >
```

```
</EMBED >
```

在上面的代码中, EMBED 标记包含了与 APPLET 标记相似的信息, 这些信息用于在 Navigator 中启动 Java Plug-in。EMBED 标记的 type 属性标识 Java 执行程序的类型, 是 applet 还是 Bean。Navigator 根据 type 属性加载 Java Plug-in, 并初始化 Java 执行程序。

APPLET 标记的 width、height 和 align 属性直接映射到 EMBED 标记中同样的属性, Navigator 用这些格式信息定位 applet 程序的显示, 由于这些格式信息没有改变, 所以用 Java Plug-in 和用 APPLET 标记的 applet 程序在浏览器中的显示是一样的。

与 OBJECT 标记不同, 所有的信息直接存放在 EMBED 标记中而不用 PARAM 标记。因此 APPLET 标记的所有属性和参数必须映射到 EMBED 标记的属性。在上面的例子中, APPLET 标记的 code 和 codebase 属性映射到 EMBED 标记的属性。code 属性代表 applet 程序, codebase 属性代表下载附加的类文件或 JAR 文件的 URL。APPLET 标记中的 model 参数映射到 EMBED 标记中的 model 属性。

与 OBJECT 标记的 codebase 属性一样, EMBED 标记的 pluginspage 属性指向下载 Java Plug-in 的 Web 站点, 只有在本机上未安装 Java Plug-in 时, Navigator 才会使用 pluginspage 属性。

6. 同时在 IE 和 Navigator 中使用 Java Plug-in

在 IE 中用 OBJECT 标记, 在 Navigator 中用 EMBED 标记能使 HTML 页面使用 Java Plug-in。

但是在 Internet/Intranet 环境中, 人们浏览 HTML 页面时, 很可能有的人用 IE, 有的人用 Navigator。所以要求当用 IE 和 Navigator 浏览同一 HTML 页面时, 都能激活 Java Plug-in。下面 HTML 代码 4 的写法可以做到这一点。

```
<OBJECT classid = "clsid:8AD9C840 - 044E - 11D1 -
B3E9 - 00805F499D93"
width = "200" height = "200" align = "baseline"
codebase = "http://java.sun.com/products/plugin/1.
1/jinstall-11-win32.cab#Version=1,1,0,0">
<PARAM NAME = "code" VALUE = "XYZApp.class">
<PARAM NAME = "codebase" VALUE = "html/">
<PARAM NAME = "type" VALUE = "application/x-java-
applet;version=1.1">
```

```
<PARAM NAME = "model" VALUE = "models/
HyaluronicAcid.xyz">
<COMMENT>
<EMBED type = "application/x-java-applet; ver-
sion=1.1" width = "200"
height = "200" align = "baseline" code = "XYZApp.
class"
codebase = "html/" model = "models/Hyaluroni-
cAcid.xyz"
pluginspage = "http://java.sun.com/products/plu-
gin/1.1/plugin-install.html">
<NOEMBED>
</COMMENT>
No JDK 1.1 support for APPLET!!
</NOEMBED></EMBED>
</OBJECT>
```

如果用户用 IE 浏览 HTML 代码 4, IE 能识别 OBJECT 标记, 所以 IE 会正常加载 Java Plug-in。COMMENT 标记是 IE 的专有标记, 它用于放置注释文本, 浏览器忽略 <COMMENT> 与 </COMMENT> 标记之间的内容, IE 中不识别 NOEMBED 和 EMBED 标记, 因此, </NOEMBED> </EMBED> 这一行被忽略。这样 HTML 代码 4 在 IE 中变得与 HTML 代码 2 一样。

如果用户用 Navigator 浏览 HTML 代码 4, Navigator 不识别 OBJECT 和 COMMENT 标记, 会忽略 OBJECT 和 COMMENT 标记, 经过忽略后, HTML 代码 4 与 HTML 代码 3 一致。

这个例子说明了如何组合 OBJECT 和 EMBED 标记可以达到无论用 IE 浏览器还是 Navigator 浏览器都能激活 Java Plug-in, 并保证属性和参数的设置正确。

另外, 使用 Sun 公司的 Java Plug-in HTML 转换器软件能自动地完成 HTML 代码 1 到 HTML 代码 4 的转换。

参考文献

- [1] Sun Microsystem Inc. Java Plug-in HTML Specification 1.0, 1998.

(来稿时间: 1998 年 12 月)