

基于 RSS 的信息推送系统的设计和实现^①

Design and Implementation of RSS – based Information Push System

杨俊¹ 兰宏勇² (1. 昆明理工大学 信息工程与自动化学院 云南 昆明 650051;
2. 云南师范大学 数学学院 云南 昆明 650092)

摘要: RSS 信息推送技术属于 Web2.0 时代兴起的新技术之一,它的出现提供了个性化获取信息的一种新方式。本文通过对该技术的研究,在 Java 类库 ROME(RSS and Atom Utilities)的基础上,设计实现了 RSS 提要的产生方案,并为达到高效的目的进行了分析和改进。最后将 RSS 技术嫁接到手机终端,实现了基于 KXML 的 J2ME 版的 RSS 阅读器。本系统的使用将减轻建立各类信息网站的复杂性,提供了一个方便高效的内容实时发布平台,具有良好的应用和发展前景。

关键词: RSS 信息推送 ROME 缓存 KXML J2ME

1 引言

互联网的迅猛发展,使用户面临着“找信息”和“更新信息”两大难题。像 Google 和百度这样的搜索引擎解决了第一个问题,而第二个问题的解决方法就要依赖 RSS 技术了。RSS 是一种用于共享新闻和其他 Web 内容的数据交换规范。它与传统获取信息的方式不同,RSS 提供了将信息“推”向用户的方式。目前,越来越多的网站开始在自己的 Web 页面上提供 XML 格式的 RSS 提要,允许用户通过 RSS 阅读器进行阅读,使用户不再需要不厌其烦地辗转于各个博客站点和逐一访问感兴趣的新闻网站,只需在软件上订阅这些网站的 RSS 频道。然而,随着这些 RSS Feed 的流行和广泛订阅,如何产生高效的 RSS Feed 并减轻服务器的负荷和带宽的占用成为大家要面对的重要问题。同时,基于手机客户端的 RSS 阅读器能够突破时间和空间的限制,具有灵活、便捷、高效的优点,真正实现了在任何时间、任何地点获取网络服务的目标。使用户及时获取自己定制的最新信息,并有选择地进行高效阅读,从而给用户节约了大量宝贵的时间。

2 RSS 信息推送技术

RSS 到底是什么?它可以是“Rich Site Summary

(丰富站点摘要)”、“RDF Site Summary(RDF 站点摘要)”还可以是“Really Simple Syndication(真正简易聚合)”。这主要是因为该技术有不同的源头,不同的技术团体对其有不同的解释。目前,RSS 版本有 RSS0.9x/2.0 和 RSS1.0。

信息推送技术是一种基于客户服务器的机制,由服务器主动将信息发往客户端的技术。它能让信息主动地寻找用户,具有主动性和及时性。使用 RSS 来实现信息推送有两个步骤:首先是 RSS 的联合,发布一个 RSS 文件(一般称为 RSS Feed,即一个具有固定 URL 的 XML 文档)后,这个 RSS Feed 中包含的信息就能直接被其他站点调用,而且由于这些数据都是标准的 XML 格式,所以也能在其他的终端和服务中使用。然后是 RSS 的聚合,网络用户可以在客户端借助于支持 RSS 的新闻聚合工具软件,在不打开网站内容页面的情况下,可以收集和定制网络新闻,阅读支持 RSS 输出的网站内容^[1]。图 1 是推送过程的示意图^[2]。

3 生成有效的 RSS Feed

3.1 如何有效的生成 RSS Feed

为了更有效地生成 RSS Feed,本文采用动态生成

^① 基金项目:国家自然科学基金资助项目(10772158);云南省引进高层次人才工作经费(2003);云南省自然科学基金项目(2005A0026M)资助。

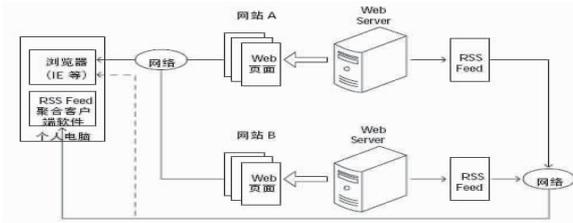


图 1 RSS 示意图

RSS Feed 的方法,即由运行在 Web 服务器上的应用程序代码来生成 RSS Feed,它的优点是可以根据 Web 请求参数或者路径信息产生不同的内容,也就是说动态的 RSS Feed 能够被参数化。同时我们选择 Java 下的 ROME 技术来生成 RSS Feed,因为 ROME 易于使用,而且它可以根据需要生成任何格式的 RSS Feed,并且 ROME 只依赖 Java 和 JDOM 解析库,使用 ROME 时只需要使用 ROME jar 和 JDOM jar 两个 jar 文件即可。

3.2 系统设计实现

RSS Feed 的产生和解析遵循 RSS 标准规范的编码、解码过程,见图 2。

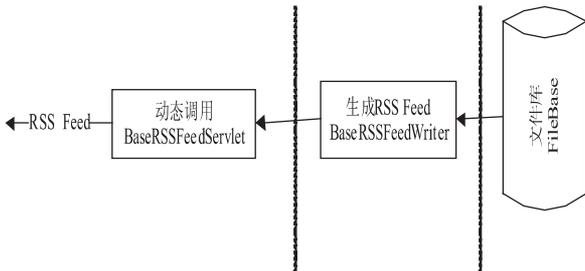


图 2 产生 RSS Feed 的原理图

下面主要介绍如何用 ROME 建立一个 RSS Feed 生成器来生成任何 RSS 甚至是 Atom Feed。在 ROME 中, RSS Feed 对象是通过 SyndFeedImpl 类创建的。

```
SyndFeed feed = new SyndFeedImpl();
```

ROME 中使用 SyndEntry 类对应 < item > 元素,因为在一个 < channel > 中包含多个 < item >,所以需要使用列表 List 对象将每个 SyndEntry 对象收集起来,一次提交给 RSS Feed 对象。在 BaseRSSFeedWriter.java 这个类中的 write() 方法实现了上面的操作,主要代码如下^[3]:

```
public void write( Writer writer, String baseUrl, String format ) throws Exception
{
```

```
SyndFeed feed = new SyndFeedImpl();
//告诉 ROME 要生成的 RSS Feed 格式,设置属性
feed.setFeedType( format );
feed.setLanguage( "en-us" );
...//略
//为了支持 Atom 格式,添加一个链接到本身的链接
```

```
SyndLink selfLink = new SyndLinkImpl();
selfLink.setHref( feed.getUri() );
selfLink.setRel( "self" );
feed.setLinks( Collections.singletonList( selfLink ) );
ArrayList entries = new ArrayList();
Iterator files = depot.getFile( ).iterator();
//遍历文件库中返回的文件
While( files.hasNext() )
{
    SyndEntry entry = new SyndEntryImpl();
    String url = baseUrl + file.getName();
    entry.setLink( url );
    ...//略
    entries.add( entry );
}
WireFeedOutput output = new WireFeedOutput();
WireFeed wireFeed = feed.createWireFeed();
output.output( wireFeed, writer );
}
```

下面写一个简单的 Java Servlet 来调用上面的 write() 方法,实现动态生成 RSS Feed。

```
protected void doGet( HttpServletRequest request,
    HttpServletResponse response ) throws
    ServletException, IOException
{
    ServletContext application =
    this.getServletContext(); //获取上下文环境
    //找文件库,若没有,则创建一个
    ...
    try
    {
        //查找格式参数
```

```

...
String url =
request.getRequestURL( ).toString( );
String baseUrl =
url.substring( 0 ,url.lastIndexOf( "/" ) );
BaseRSSFeedWriter baseWriter = new
BaseRSSFeedWriter( response.getWriter( ),
baseUrl ,format );
baseWriter.write( response.getWriter( ),
depotUrl ,format );
}
catch( Exception ex )
{
...//略
}
}

```

3.3 技术改进

随着 RSS Feed 的广泛流行,订阅者的不断增加,势必会导致服务器不堪重负,而缓存将是解决问题的最好出路。所以对 BaseRSSFeedServlet 类进行改进。

(1) 通过 HTTP 条件 GET 来支持客户端缓存:

```

Date sinceDate = new Date(
request.getDateHeader( " If - Modified - Since " ) );
if( sinceDate != null )
{
    if( base.getLastUpdateDate( ).compareTo(
        sinceDate ) < = 0 )
    {
        response.setStatus( HttpServletResponse.
SC_NOT_MODIFIED );
        response.flushBuffer( );
        return ;
    }
}

```

(2) 通过 HTTP 缓存控制头来支持代理缓存:

```

response.setDateHeader( " application/rss + xml ;
charset = utf - 8 );
base.getLastUpdateDate( ).getTime( );
response.setHeader( " Cache - Control " ; max -
age = 5400 ,must - revalidate" );

```

(3) 使用简单的 LRU 缓存来支持服务器缓存:

```

LRUCache cache = new LRUCache( 5 ,5400 );//保
存 5 条记录,记录的过期时间为 1.5 小时

```

通过上面的改进,即可降低对计算机、网络和宽带的占用,从而达到高效的目的。

4 基于手机客户端的 RSS 阅读器的实现

生成 RSS Feed,经过校验有效并发布后,即可被 RSS 阅读器接收使用。应用在手机端的 RSS 阅读软件比较少见。主要是因为各种手机设备千差万别,不同型号手机的设计开发一般来说也不同,从而大大增加了开发的工作量。Java 手机及 J2ME 跨平台技术的出现,在一定程度上缓解了上述问题。

因为手机内存的限制,本系统采用 KXML 技术来解析 RSS Feed,这种解析方式,可以做到一边后台扫描文档,一边前台将解析到的新闻 Title 显示在用户界面上,不影响用户阅读新闻列表^[4]。

下面分析 KXML 如何做一个 Pull 语法分析程序。

首先创建一个 XmlParser,把它传到输入流中,并且跳过 RSS 根节点。

主要代码如下:

```

public void parseRssFeedXml( InputStream reader )
throws IOException , XmlPullParserException
{
    KXmlParser parser = new KXmlParser( );
    parser.setInput( reader , null );
    parser.nextTag( );
    parser.require( parser.START_DOCUMENT , null ,
null );
}

```

然后寻找 RSS 中代表每一个新闻的“item”节点下的三个子节点“title”、“link”、“description”。

主要代码如下:

```

for( int eventType = parser.getEventType( );
eventType != XmlPullParser.END_DOCUMENT ;
eventType = parser.next( ) )
{
    if( eventType == XmlPullParser.START_TAG && "
item" . equals( parser.getName( ) ) )
    {

```

```

eventType = parser.next( );
while( true )
{
    tmpText = parser.getName( );
    if ( eventType == XmlPullParser. START_
TAG && " title" . equals( tmpText ) )
    {
        parser.next( );
        headlineTitle =
            parser.getText( ). trim( );
    }
    else if( eventType ==
XmlPullParser. START_TAG&&
" description" . equals( tmpText ) )
    {
        parser.next( );
        headlineDescription =
            parser.getText( ). trim( );
    }
    else if( eventType ==
XmlPullParser. START_TAG &&
" link" . equals( tmpText ) )
    {
        parser.next( );
        newsLink = parser.getText( ). trim( );
    }
    eventType = parser.next( );
    if ( eventType == XmlPullParser. END_TAG
&& " item" . equals( parser.getName( ) ) )
    {
        titleList. newTitle( headlineTitle ,
            headlineDescription ) ;
        break ;
    }
}
}
}

```

其中 parser.next() 将分析器推进到下一个事件, 这样就可以遍历整个文档。混淆打包后, 通过手机模拟器运行, 图 3 ~ 图 6 演示了该程序读取 RSS Feed 源

的运行效果图。



图 3 RSS Feed 列表



图 4 等待解析界面



图 5 标题列表



图 6 文章摘要

5 结语

本系统与已有的信息推送系统相比在以下方面做了一些创新探索。

第一方面是结合内容库动态生成 RSS Feed。网站可以利用这个功能自动提供内容供用户进行订阅, 从而实现内容的推送服务。而用户用一种客户端工具就能够访问和更新用户所定制的所有内容, 再也不需要频繁打开所有的网站去寻找自己需要的东西。

第二方面是分析随着 RSS Feed 的广泛流行, 订阅者不断增加, 势必会导致服务器不堪重负, 由此提出了解决问题的方案。

第三方面是打破了桌面 RSS(下转第 31 页)

(上接第 67 页)

阅读器的常规,开发了一个基于手机终端的 RSS 阅读器,突破了时间和空间的限制,达到了灵活、便捷、高效的目的,真正实现了在任何时间、任何地点获取网络服务的目标,使用户及时获取自己定制的最新信息,从而给用户节约了大量宝贵的时间。

本系统的使用将大大减轻建立各类信息网站的复杂性,提供了一个方便高效的内容实时发布平台,从“以用户为中心”的角度出发,RSS 快速、简要、个性化定制的特点更适合移动媒体^[5]。从长远来看,RSS 更需要拓展思路,与新兴的手机终端结合,从而带来更为广阔的市场。

参考文献

1 蔡巍.“推送”(Push)技术简介.中国信息报道,

1999,21(1):2-5.

2 Winer D. RSS2.0 Specification. <http://blogs.law.harvard.edu.tech/rss>.

3 Johnson D. RSS AND Atom IN ACTION. 1st ed., New York: Manning Publications, 2007. 187-201.

4 Yuan M J. J2ME 移动应用程序开发. 1st ed., 北京:清华大学出版社, 2004. 93-95.

5 陈圣俭,程小梅.基于 J2ME 的 RSS 阅读器的开发与实现.现代通信,2007,1(2):116-118.