The Utility of RSS for the Classroom

Ronald J. Glotzbach
Purdue University
West Lafayette, IN, USA
rjglotzbach@purdue.edu

James L. Mohler
Purdue University
West Lafayette, IN, USA
jlmohler@purdue.edu

Abstract: Today’s post-secondary students are technologically savvy and they have come to expect faculty to use the myriad web technologies for different aspects of course delivery. Technologies such as RSS, blogs, web- and pod-casting are quickly being integrated into classroom materials to provide learning anytime, anywhere, and in multiple media modalities. Modern courses must not only be accessible (e.g., temporally, spatially, and technologically) but also accommodate different learning styles. This contribution outlines the novel use of RSS technology for a course announcement system. It begins by describing RSS technology, the application of RSS to create a real-time announcement system in a course, and the technical details of its creation and implementation.

What is RSS?

While the concept for RSS emerged in 1997 with the release of channels in Microsoft’s Internet Explorer 4.0 browser, the first version of RDF Site Summary (RSS) emerged in 1999. It is also identified by several other names including Real Simple Syndication, Rich Site Summary, Real-time Simple Syndication, and others.

While it has suffered from “the name game,” like many XML-based technologies, it has been comparatively slow to grab a practical foothold on the web. Although many news and advertising sites are now using RSS for real-time distribution, the base of consumers receiving RSS feeds remains mostly composed of early adopters. However, the generation who grew up on computer technology is now quickly adding to the base of people receiving RSS feeds. Additionally, the explosion of wireless technologies has helped to fuel this growth. According to BusinessWeek Online (2006), the number of RSS feeds grew from 307,000 in January of 2004 to over 13 million in August of 2005. This is more than 20 times the number of new feeds emerging during the prior period. A SlashDot survey (Hrastnik, 2005) predicts that RSS will continue to grow dramatically in the coming years based upon a survey that they conducted with their user base.

RSS allows content distributors to syndicate brief snippets of their content and post it as an RSS (XML) file on the web. Most RSS files include a title, brief description and a link where the user can follow-up to retrieve the “full-story.” Those who wish to receive RSS content use special applications called RSS aggregators to “subscribe” to RSS feeds. Once subscribed to a feed, when a new item is added to an RSS feed, the consumer is immediately notified in some manner. In this way, RSS feeds provide an active information mechanism on the web whereby consumers can know immediately of distributors’ information.

There are a variety of standalone RSS aggregators, and some browsers, such as Mozilla Firefox, include RSS functions within them. It should also be noted that RSS consumption is not limited to desktop applications alone – PDAs, cell phones and other wireless devices can be set up to receive RSS feeds. While the original goals of RSS may be loftier, in practical use organizations are using RSS feeds to create content summaries of their web sites to attract users to come and consume their content. One of the biggest limitations of any web site is the fact that it is basically passive – users must come there to see what’s new. RSS, one of many technologies that provide an active method for attracting traffic and getting individual consumer attention, is gaining popularity for this specific purpose.
Classroom Need

Like all web sites, course web sites suffer from the passive restrictions. Educators who use the web for content distribution would say that student attention and motivation is more difficult to obtain due to the hectic schedules many of them have. While most instructors do post announcements to their course web sites, it is questionable how many students actually come to the web site to read the announcements and how many of them actually read the announcements once there.

Realizing this problem, the authors decided to try implementing RSS technology into the courses that they teach. In the spring of 2006, the six courses under the authors’ purview were modified to include an RSS feed that was used for announcements for the course to determine the impact on student awareness. Anecdotal comments from students have suggested that many of the students have subscribed to the course feeds and found them to be worthwhile. Although not part of the initial implementation, in future semesters the authors’ intend on tracking usage data associated with the feeds to gather specific information rather than anecdotal data.

RSS Course Announcement System

The remainder of this contribution details the specifics of the RSS system created by the authors. The “system” includes both the interface components as viewed in the browser as well as the ASP scripts that were used to modify the RSS (XML) file.

The Course Announcement RSS Document

An RSS document is written in a consistent and repetitive format that aggregators and feed readers can easily parse into useful headlines. Listing 1 shows the frozen structure of a RSS 2.0 document.

```xml
<?xml version="1.0" ?>
<rss version="2.0">
  <channel>
    <title>CGT 141</title>
    <link>http://www2.tech.purdue.edu/cgt/courses/cgt141/</link>
    <description>Announcements for CGT 141</description>
    <language>en-US</language>
    <copyright>Copyright 2006 Purdue University</copyright>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <lastBuildDate>Sun, 29 Jan 2006 12:13:01 PM EST</lastBuildDate>
    <item>
      <title>Midterm Exam</title>
      <description>February 28, 7-8 PM in WTHR 200</description>
      <link>http://www.tech.purdue.edu/cgt/courses/cgt141/</link>
      <guid isPermaLink="false">8</guid>
      <pubDate>Wed, 11 Jan 2006 08:00:00 EST</pubDate>
    </item>
  </channel>
</rss>
```

Listing 1: An RSS 2.0 document from the course management system.

The root or document element of an RSS file is the `<rss>` element that includes the version attribute and associated “2.0” value. Nested inside the document element is one `<channel>` element that contains all necessary sub-elements, including the items that make up the feed headlines. The `<channel>` element is arguably unnecessary given that only one can exist within the document, providing the same affect (that of being a document element) that the `<rss>` element offers. As noted by Hammersley (2005) and Orchard (2005), three elements are required inside the `<channel>` element: `<title>`, `<link>`, and `<description>`.
Displaying Announcements on the Course Home Page

While a minor part of the system, we desired to be able to have the announcements that were in the RSS feed display in the browser as shown in Figure 1. To accomplish this, ASP code was included in the course home page as shown in Listing 2.

```
<%  
Set objXML = Server.CreateObject("Microsoft.XMLDOM")  
Set objLst = Server.CreateObject("Microsoft.XMLDOM")  
Set objAnn = Server.CreateObject("Microsoft.XMLDOM")  
objXML.async = False  
objXML.Load (Server.MapPath("141_RSS.xml"))  
%><ul><%
Set objLst = objXML.getElementsByTagName("item")  
nOfItems = objLst.length  
For i = 0 To (nOfItems - 1)  
Set objAnn = objLst.item(i)  
Response.Write("<li><strong>" & objAnn.childNodes(0).text & ":</strong> " &  
objAnn.childNodes(1).text)  
currentPage = "http://" & Request.ServerVariables("SERVER_NAME") &  
LCase(Request.ServerVariables("URL"))  
If objAnn.childNodes(2).text <> currentPage Then  
Response.Write("[ <a href='" & objAnn.childNodes(2).text & "> Link </a>  
] </li><br />")  
Else  
Response.Write("</li><br />")  
End If  
Next  
%></ul>
```

Listing 2: ASP Code to extract the RSS items and write it in the course home page.

Figure 1: Announcements are integrated from RSS feed into the course home page (http://www2.tech.purdue.edu/cgt/Courses/cgt141/).
UI for Editing the RSS Feed

To edit the RSS feed for each course, there were two HTML pages. We designed this “system” so that other faculty could use it and would not have to know how to write RSS (XML) or other web code. Figure 2 shows the initial interface screen for editing the RSS feed items. From this page the user can add, modify or delete items in the feed.

![Image of initial interface screen for editing the RSS feed items.]

If users choose to edit a feed item, they are presented a form such as is shown in Figure 3. If they choose to delete an existing feed item, they are prompted to ensure they wish to delete it. If they choose to add a new item, they are presented an empty form similar to that shown in Figure 3. Both of the UI screens shown in Figures 2 and 3 use the same ASP script to manipulate the data. The screen shown in Figure 2 uses code similar to that shown in listing 2 and thus will not be reviewed again. The code for taking the input from the form in figure 3 and writing it into the RSS feed is described in the next section.

![Image of an HTML form presented if the user chooses to edit a specific feed item.]

Figure 2: The user chooses to edit, delete or add a new feed item via an HTML page.

Figure 3: An HTML form is presented if the user chooses to edit a specific feed item.

Building the Feed with ASP

Active Server Pages (ASP) provides functionality for interacting with the XML Document Object Model (DOM) via instantiation of an instance of the Microsoft XML Core Services Msxml2.DOMDocument class. Using the XML DOM, Listing 3 demonstrates how an additional element can be appended as a child of the <rss> element with relative ease. The inherent issues present with this approach are twofold. First, this example appends the <item> element as a child to the document element of the RSS file, at the same level (sibling) as the <channel> element when it should be a sub-element (child) to <channel>, as noted by Hammersley (2005) and Orchard (2005) earlier. Second, this example does not retain the formatting of the XML document, stringing the appended text together on a single line scrolling off to the right. Attempts at appending to a child node of <rss> results in an error message stating that the operation cannot be performed with a Node of type Element. Instead, appending can only occur at the document element level. As a solution, the management application builds a memory-resident view of the RSS feed, appending the new item in sequence prior to the existing items, thus allowing aggregators to display the new item first in the list. Listing 5 shows the code that performs this. The resulting RSS feed from Listing 5, unlike the previous example, contains the new <item> inserted within the existing channel as the first item in the list (compare to Listing 1). Formatting and indentation from the XML file are also preserved to ensure the document remains easy to read as a plain text document (see Listing 4).

Listing 3: Performing a simple XML append using ASP.

```vbscript
'Create new RSS elements
Set oItem       = oXML.createElement("item")
Set oItemTitle  = oXML.createElement("title")
Set oItemDesc   = oXML.createElement("description")
Set oItemLink   = oXML.createElement("link")
'Initialize the new RSS elements with values
oItemTitle.text = "Example 1"
oItemDesc.text  = "Simple Append Without Formatting"
oItemLink.text  = "http://www.tech.purdue.edu/cgt/"
'Append the <item> element to the XML file
oXML.documentElement.AppendChild(oItem)
'Append the subelements to the <item> element
oItem.AppendChild(oItemTitle)
oItem.AppendChild(oItemDesc)
oItem.AppendChild(oItemLink)
```

Listing 4: Resulting RSS feed with new item appended.

```xml
<?xml version="1.0" ?>
<rss version="2.0">
  <channel>
    ...elements omitted for brevity...
    <item>
      <title>Example 2</title>
      <description>Appending a memory-resident channel</description>
      <link>http://www.tech.purdue.edu/cgt/</link>
      <guid isPermaLink="false">9</guid>
      <pubDate>Mon, 30 Jan 2006 12:31:45 EST</pubDate>
    </item>
    <item>
      <title>Midterm Exam</title>
      <description>February 28, 7-8 PM in WTHR 200</description>
      <link>http://www.tech.purdue.edu/cgt/courses/cgt141/</link>
      <guid isPermaLink="false">8</guid>
      <pubDate>Wed, 11 Jan 2006 08:00:00 EST</pubDate>
    </item>
  </channel>
</rss>
```
Summary

This contribution has provided a novel implementation of RSS for course improvement. As stated, anecdotal student comments have indicated that they found the RSS feed for course announcements useful. Future work in this area will focus on tracking usage data to determine the number of students using the feeds and how often they click through the RSS feed to the main course web site.

References


