USING CONTENT SYNDICATION TECHNOLOGIES IN DISTRIBUTING AND PUBLISHING INFORMATION TO REACH ALL USERS

Serena Pastore
INAF - Astronomical Observatory of Padova, vicolo Osservatorio 5 – 35122 – Padova, Italy

Keywords: Content syndication, RSS specification, ATOM standards, XML technologies, web feeds, Ajax.

Abstract: Content syndication is a widely used method to distribute information as web feeds. It is an easy way of reaching the greatest number of end users requiring immediate access. Content syndication is essentially based on XML technology, is easily distributed and possesses a high level of interoperability across platforms. Both the website providing information with an up-to-date structure regardless of the different techniques to stored and manage content and the website consuming information benefit from such technology. Several different specifications and standards have been developed to support syndication, all used in every context. The paper describes how syndication technology has been used to distribute centrally located INAF information to each organizational entity's local Web site. It describes technological choices done both for producing and distributing the feed to each local website presenting it as a specific section of the home page. Feeds are produced according to different formats, and technologies and standards used are specific Web technologies collectively known as Web 2.0 applications.

1 INTRODUCTION

Content syndication technologies are essentially XML-based technologies (Moller A., et al. 2006) used to aggregate and distribute information, making it more accessible to users. Syndication differs only nominally from publishing methods using dynamic content management systems (CMS). It includes standardized protocols which permit the use of a site's data in other contexts such as other websites, browser plug-ins or a separate desktop application. It is an application of the kind collectively known as Web 2.0 applications (Murugesan, S. 2007). Web mashups are also Web 2.0 applications that combine information and services from several sources. Mashups could be used with syndication to create an improved user interface to view data. Information arrives to the user in a pull method rather then a push one in a specific format, called feed, which summarize different content giving only some items and usually a link to the place where to deepen into the topic. From the provider side, delivering feeds usually means providing information wrapped in an XML-based file. Such file is then shared and processed by many client applications such as feed readers and web aggregators as specific software that collects the feed and visualizes it inside a specific application or in a single Web page. However sometimes it is useful not only to design an application able to produce web content in a feed format, but also to provide a method to process it and visualize in an appealing way by using the interactive web technologies This paper describes how syndication and rich internet technologies have been used to distribute information, which, even if already aggregated in a system (Boccato, C., Pastore, S. 2006), needs to be interactively delivered to specific users. The Italian National Institute for Astrophysics (INAF, http://www.inaf.it) is composed of one headquarters and 19 satellite organizations mainly in Italy. Its Web site is basically designed to act as a reference source of astrophysical information for end users. However, a study has revealed that the majority of users prefer their local Web sites and do not take advantage either of the INAF Web site, which they visit only few times a week, or of other applications such as feed aggregators. It is therefore necessary to find a more effective information distribution method. Syndication and Web 2.0 technologies have thus been chosen as the way to collect, distribute and process updated information in order to be viewed

Pastore S. (2008).
USING CONTENT SYNDICATION TECHNOLOGIES IN DISTRIBUTING AND PUBLISHING INFORMATION TO REACH ALL USERS.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 228-231
DOI: 10.5220/0001518702280231

from each local Institute's Web site integrated on the home page. The main issues involved in this solution include: 1) the heterogeneous structure of information that derives from different logical and physical sources; 2) the existence of different syndication specifications and standards, each one having strengths and benefits and 3) the need to provide an application as a package to be easily included in each Web site independently of the technologies used by each webmaster. Standard web technologies (XHTML, CSS, Javascript, XML) and web programming languages which are the basis of the Web 2.0 technology paradigm, have been used to develop a solution for creating, processing and publishing information in the form of a feed.

2 PROBLEM DEFINITION

The initial problem is how best to distribute specific information that, although available on the INAF Web site, is not being accessed by potential users. The distribution approach could be divided into three phases: definition of publishing content, creation of the feed and its visualization.

2.1 Content Selection

The main data to be published (Figure 1) concerns different aspects of the Institute and astrophysics in general and could be practically divided according to their structure.

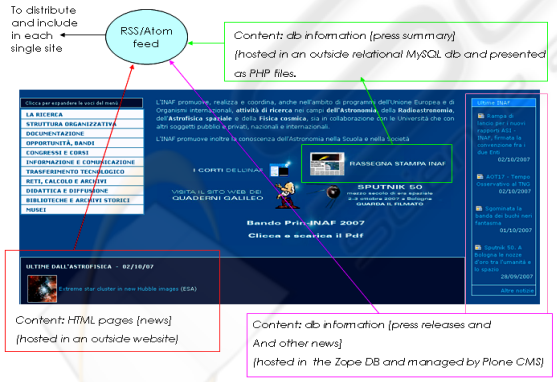

Figure 1: The main problem is how to aggregate and publish information coming from different sources and display them in a single web feed.

Content may be organized in a relational database and managed by a LAMP (Linux, PHP/MySQL) platform (Davis, M.E., Philips, J.A., 2007). Alternately, content could be organized in an object database and managed through a CMS such as the Plone/Zope/Python environment (Boccato, C, et al. 2006) or it could simply be stored as XHTML/HTML pages and thus organized hierarchically by tags. The goal is to create a unique feed containing items coming from various sources and distributing it to each institution to reside on their home pages. Every publishing system frequently makes available a library of automatic feed creation tools, and there are many scrapers which extract web page content and create feeds. In these ways, each produced feed may be merged with others to produce a solution, just as happens in many on-line web site aggregators.

Unfortunately such an approach involves incompatible feed formats and therefore requires further processing. The solution is thus to collect all the information to be published in a unique database from which data can be extracted to create the requested feed according to the format. This approach implies that information is twice published, but it provides more flexibility in successive feed processing.

2.2 Feed Creation: Syndication Specifications and Standards

Content syndication has various implementations and there is no ruling body. It is therefore necessary to choose the standard which suits the need. Each implementation however shares a common logical structure following an XML syntax. Content is organized into a so-called "channel", an entity to which refers as information provider. Each channel consists of single chunks of information (the so-called item), each one possessing attributes (title, description, a reference to the information, etc.). The main technology used is RSS. This consists of specifications developed by specific groups of interested people (as in the case of RSS version 1.0, http://web.resource.org/rss/1.0/) or by organizations (as in the case of RSS version 2.0, http://www.rssboard.org/). The Atom 1.0 syndication format (http://www.ietf.org/html.charters/atompub-charter.html) grew out of RSS and is a standard developed by the Internet Engineering Task Force (IETF). Moreover, Microsoft has introduced Simple Sharing Extensions (SSE) (http://msdn2. microsoft.com/ en-gb/xml/ bb510102.aspx), which extends the Atom 1.0 and RSS 2.0 specifications, while Javascript Object Notation (JSON, http://www.json.org) is a data-interchange format used by Google as another feed format. A comparison of the different attributes used by the main feed specifications is shown in figure 2.

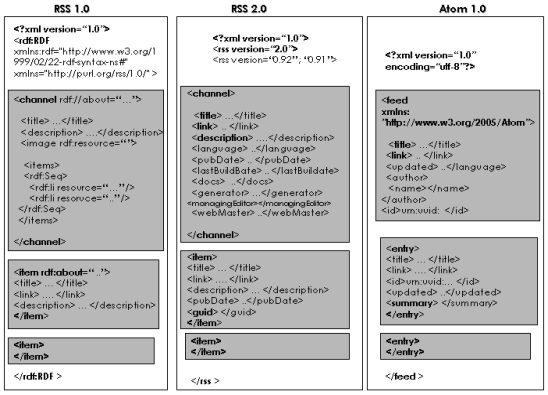

Figure 2: A comparison between RSS and Atom formats.

2.2.1 RSS 1.0, RSS 2.0 and ATOM 1

Despite having the same acronym, RSS 1.0 and RSS 2.0 are distinct and incompatible formats. RSS 1.0 stands for RDF Site Summary and incorporates the Resource Description Framework (RDF, http://www.w3.org/RDF/) and its tags and attributes to better describe resources. The basic structure of RSS 1.0 involves wrapping the entire feed in the `<rdf:RDF>` element which contains the definition, attributes and list of items of a `<channel>` (the source of information) and each item and its attributes specifically described in the `<item>`. Specification flexibility allows the use of metadata to attach information to the feed by integrating other standards (i.e. the Dublin Core, http://dublincore.org/) useful for semantic processing, even if they are a bit verbose. RSS 2.0, which follows on from various RSS 0.9x specifications, was developed by Netscape and later by Useland. It stands for Really Simple Syndication to emphasize its ease of use. According to this format, the feed is described inside the `<rss>` tag and includes a `<channel>` metadata with a set of attributes (which contain more information than in the previous format) and then the list of items and their attributes (i.e. standard as link, title and description metadata and other facilities like enclosure which allows attachments to be automatically downloaded, or a `<guid>` element that identifies the item uniquely). Finally Atom, as defined by IETF in the last 1.0 version, is a standard which defines both a feed representation format (the Atom Syndication Format, RFC 4287, http://www.ietf.org/rfc/rfc4287.txt) and an interaction protocol (the Atom Syndication Format an internet drafts, http://www.ietf.org/internet-drafts/draft-ietf-atompub-protocol-17.txt) with enhanced interoperability. In the Atom format, the feed is specified by the `<feed>` metadata that initially describes the channel (even if it does not associate it with a specific tag) and its attributes and then specifies each item inside the `<entry>` tag. Most client feed applications deal with each format. A web application which creates syntactically corrected and validated feeds following the different formats, may however guarantee a spread information delivering.

2.3 Feed Processing

Despite having different standards, feed formats are XML files and may be managed and processed by many libraries and tools developed using different programming languages (i.e. PHP MagPie RSS, http://magpierss.sourceforge.net/, the Java ROME https://rome.dev.java.net/, or Python RSS.py (http://www.mnot.net/python/RSS.py). Many are distributed as on-line tools (for example, a lot of scraping tools are used as web aggregators like xpath2rss, http://freshmeat.net/projects/xpath2rss/) despite the fact they do not provide a packaged solution to be delivered to each website. However, the common underlying concept is the extraction of information, its formatting according to XML syntax and the processing and parsing of the visualization inside a Web page or other application. Focusing on the visualization inside a Web page, the simplest way is to include an external feed by pointing to a RSS parser developed with every language which processes it and then presents the content according a specific style through CSS technologies (Schmitt, C., 2006). The choice of the language and the platform is subjective.

3 THE DEVELOPED SOLUTION

After the analysis of constraints and issues, the developed solution has followed the rule of easy implementation and requires the design of the content database, the development of a Web application for producing the feeds and the establishment of simple visualization procedure as means of a set of scripts and CSS templates. The LAMP platform has been chosen for the first two phases, while other more interactive technologies, collectively known as the Ajax paradigm (Gross, C., 2006), have been adopted for the visualization phase. The MySQL schema design requires more work to include all the attributes related to each feed specification needed for successive validation. The

Web application for producing and publishing the feed has been developed as a Web form user interface (Figure 3) used by authors to insert content. Thus the application, taking advantage of the FeedCreator.class.php tool (http://www.bitfolge.de/rsscreator-en.html), creates the three different validated feeds by extracting the needed information from the database, according to RSS 1.0, RSS 2.0 and Atom 1.0 formats and publishes them in a specific directory of the Web server. Then the feeds may be distributed as they are to several Web site to be integrated into the publishing systems or into another application. The feeds may also be directly integrated into Web pages using the <link> tag (inside the <head> section of the HTML page) and the type specification within the type attribute (i.e. `type="application/atom+xml" href="file"`).
At this point the methods are different for visualizing the feeds inside each home page (Figure 3) because they make use of specific produced stylesheets to tailor the presentation.

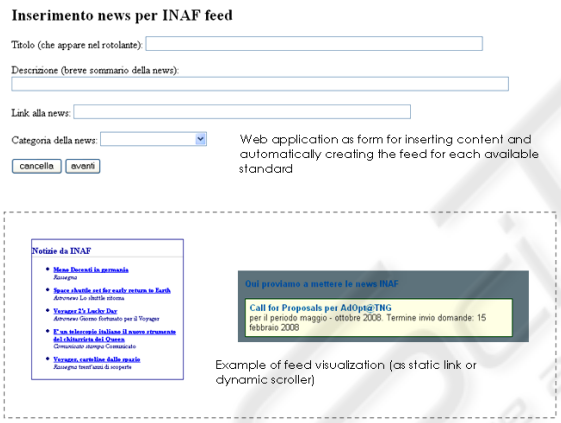

Figure 3: Interface of the Web application which creates the feed and examples of feed visualization.

A first implementation uses a static solution of a simple list by including a PHP RSS parser inside the webpage and thus simple HTML/PHP code even if it requires that the target web server supports PHP language. Other interactive techniques use Javascript languages both in a synchronous (client-side) or asynchronous (server-side), allowing scrolling of the content (Figure 3). Ajax technology in particular enhances interactivity and usability, since information is represented, processed and dynamically displayed according the Document Object Model (DOM, http://ww.w3.org/DOM) and JavaScript with a fast message exchange with the server. An example of this technique is the Google's Ajax feed API (http://code.google.com/apis/ajaxfeeds/), a library which manipulates Atom or RSS feeds, through an Ajax interface. Its application requires the creation of a key usable within all URLs of a site directory where content is stored and the API invocation is simply included inside a script tag in the page:

```
<script type="text/javascript"
src="http://www.google.com/jsapi?key=AA
A"></script>
<script type="text/javascript">
  google.load("feeds", "1");
</script>
```

However, the followed solution gives more freedom to each local webmaster which could decide to choice the feed format, a visualization solution or even use his publishing system.

4 CONCLUSIONS

It is necessity that information to be published and distributed in a simple, straightforward way to reach as many end users as possible. This need has led INAF developers to study methods to display specific information as feeds in the local Web sites of its organizational entities. Approaching different specifications nowadays used to describe syndication, a specific web application has been developed which creates web feeds from several content sources following several formats and allows their dynamic visualization by adopting interactivity technologies like Ajax used in the web 2.0 paradigm. Moreover other web 2.0 applications such as web mashups could be integrated with feeds to guarantee a better user experience.

REFERENCES

Moller A., and Schwartzbach, M., 2006. *An Introduction to XML and Web technologies*, Pearson Education.

Murugesan S., 2007. Understanding Web 2.0. *IT Professional*, Vol. 9, Issue 4, July-Aug. 07, pp. 34-41.

Boccato, C., Pastore, S., 2006. The Web Information System of the INAF: different actors contributing to disseminate information, In *Current Research in Information Sciences and Tech. Multidisciplinary approaches to global information systems*, Volume I, Open Institute of Knowledge, pp. 507-511.

Davis, M.E., Philips J.A., 2007. *Learning PHP & Mysql*. O'Reilly

Schmitt, C., 2006. *CSS Cookbook*. O'Reilly

Gross, C. *Ajax Patterns and Best practices*. Apress. 2006.