# Content Syndication

## Web Architecture and Information Management [./] Spring 2009 — INFO 190-02 (CCN 42509)

Erik Wilde, UC Berkeley School of Information

**2009-04-06**

 [http://creativecommons.org/licenses/by/3.0/]

This work is licensed under a CC Attribution 3.0 Unported License [http://creativecommons.org/licenses/by/3.0/]





## Contents

# Abstract (2)

For many information sources on the Web, it is useful to have some standardized way of subscribing to information updates. Syndication formats such as RSS and Atom can be used by these information sources to publish a feed of updated information items. Feeds can be read directly in a browser, but in most cases they are read by specialized software; either a *feed reader* that allows users to subscribe to more than one feed and manage the information received through all these feeds, or some software module that reads feeds and embeds them for example in a Web page. This latter example is the classical usage of feeds; news feeds published by news agencies, and them embedded as news tickers into Web pages as a constantly updated source of information.





# Content Feeds (3)

- Early Web content was static or updated very infrequently
  - there was not yet the requirement to reuse content in different contexts
- Frequently updated Web content quickly became a very common scenario
  - as commercial interests took over the Web, users should have a reason to re-visit a site
  - presenting a steady stream of new content creates the image of a live Web site
- There are two major use cases where HTML is not sufficient
  1. users want an efficient way to get the updated content from a site
  2. sites want to aggregate updated content from other sites and re-publish it
- Syndication Formats [Syndication Formats (1)] are designed to support these two use cases
  - container formats for updated items
  - a small amount of metadata about these items for automated processing

# Syndication Formats

## RSS History (5)

- "The Myth of RSS Compatibility [http://diveintomark.org/archives/2004/02/04/incompatible-rss]" provides a good overview
- RSS is a schoolbook example for "why standards are a good thing"
  - RSS 0.9 was created for the *My Netscape* portal in March 1999
  - RSS 0.91 (a simplification) was introduced in July 1999 (as an interim solution)
  - the AOL/Netscape merger removed the format from the company's portal
  - RSS was without an owner, and different parties claimed/denied ownership
  - RSS 1.0 was created by an informal developer group
  - RSS 0.92 (and 0.93 and 0.94) were published without acknowledging RSS 1.0
  - finally, RSS 2.0 was released as a follow-up to the RSS 0.9x versions
- Using RSS has become an exercise in managing a menagerie of versions





## The Case for Content Management (6)

- RSS is very rarely produced by hand
  - by definition, RSS contains redundant information for a specific purpose
- If a *Content Management System (CMS)* is used, RSS can be generated
  - basic metadata can be generated by the CMS (title, author, date)
  - better tagging of content results in better tagging of feeds
  - well-tagged feeds are better foundations for large-scale reuse of feed items
- Blogging is simply a specialized case of a CMS
  - Web-based interface for controlling everything
  - strictly time-ordered sequenced of published items
  - navigation features primarily based on the time-specific facets of the blog (maybe tags)
  - all blogging tools include feed support

# Consuming RSS (7)

- RSS feeds often have quality problems
  - surprisingly often feeds do not even deliver well-formed XML
  - the use of embedded markup in RSS is not well-defined
- Writing an RSS reader from scratch is not a good idea
- There are three major tasks which RSS readers must do
  1. accept non-XML RSS feeds and fix them to be XML
  2. look at the feed contents and bring them into a unified form
  3. produce a unified view of feeds regardless of the RSS version





# Atom History (8)

- RSS's shortcomings were very apparent and could not be fixed
- In mid-2003, discussions started about an improved format
- It also became apparent that the format should have a protocol
- Atom 0.3 was released in December 2003 but had no formal home
- IETF was chosen as the new home with a working group in June 2004
- RFC 4287 [http://dret.net/rfc-index/reference/RFC4287] was published in December 2005
- *AtomPub* has been published as RFC 5032 [http://dret.net/rfc-index/reference/RFC5032] in October 2007

# Atom vs. RSS (9)

- Standardized by the IETF (well-defined process)
- Classification of entries (user-defined categories)
- More XML-like markup design (more nesting)
- Namespaces are used and supported as standard mechanism
- Atom feeds *must* be well-formed XML (there even is a schema [http://atompub.org/2005/08/17/atom.rnc])
- Interpretation of content is well-defined (various content types)
- Support for `xml:lang` and `xml:base`





# Atom Example (10)

```
<feed xmlns="http://www.w3.org/2005/Atom" xml:lang="en-us">
 <title>ongoing</title>
 <id>http://www.tbray.org/ongoing/</id>
 <link rel='self' href="http://www.tbray.org/ongoing/ongoing.atom"/>
 <updated>2007-04-11T12:55:09-07:00</updated>
 <author>
  <name>Tim Bray</name>
 </author>
 <subtitle>ongoing fragmented essay by Tim Bray</subtitle>
 <entry xml:base="When/200x/2007/04/02/">
  <title>Atom Publishing Protocol Interop!</title>
  <id>http://www.tbray.org/ongoing/When/200x/2007/04/02/APP-Interop</id>
  <published>2007-04-02T13:00:00-07:00</published>
  <updated>2007-04-10T14:24:00-07:00</updated>
  <category scheme="http://www.tbray.org/ongoing/What/"
term="Technology/Atom"/>
  <category scheme="http://www.tbray.org/ongoing/What/" term="Technology"/>
  <category scheme="http://www.tbray.org/ongoing/What/" term="Atom"/>
  <content type="xhtml">
   <div xmlns="http://www.w3.org/1999/xhtml">
    <p>Mark your calendar: <a href="http://www.intertwingly.net/wiki/pie
/April2007Interop">April 16-17 at Google</a>. <em>Everybody</em> is invited,
provided they bring along an APP implementation, client or server. This was
just announced a couple of days ago, and as I write this there are already
<s>six</s> twelve client and <s>seven</s> fourteen server implementations
signed up to be there and try to <a href="http://www.intertwingly.net/wiki/pie
/InteropGrid">fill in the grid</a>. Let's drop some names, in alphabetical
order: AOL, Flock, Google, IBM, Lotus, Microsoft, Oracle, O'Reilly, Six
Apart, Sun, WordPress. Um, have I mentioned that the APP is going to be
huge?</p>
   </div>
  </content>
 </entry>
</feed>
```

# Atom Content (11)

- RSS had no safe way of finding out what an entry's content is
  - this led to different implementations using "smart ways" of what the RSS author really wanted
  - one of Atom's main goals was to improve this in a well-defined way
  - Atom allows escaped markup (the only way to include non-XML HTML in an XML format)
- Each content element should have a type (the default is `text`)
- Atom's content interpretation algorithm (use first applicable rule):
  1. if type is `text`, no child elements are allowed (plain text content)
  2. if type is `html` then RSS's method of escaped markup is used
  3. if type is `xhtml` then there must be an div containing XHTML markup
  4. if type is an XML media type [Media Types] then the content should be treated as this type
  5. if type starts with `text/` then no child elements are allowed
  6. for all other values, the content must be an base64-encoded entity of the specified MIME type





# Atom Content Examples (12)

```
<content type="xhtml">
<div xmlns="http://www.w3.org/1999/xhtml">
One <strong>bold</strong> foot forward
</div>
</content>
```

[http://www.xml.com/lpt/a/1633]

```
<content>The "atom:content" element either contains or links to the content
of the entry. The content of atom:content is Language-Sensitive.</content>
```

[http://www.xml.com/lpt/a/1633]

```
<content type="html">The &lt;code>atom:content&lt;/code> element either
contains or links to the content of the entry. The content of
&lt;code>atom:content&lt;/code> is &lt;a href="http://www.ietf.org
/rfc/rfc3066.txt">Language-Sensitive&lt;/a>.</content>
```

[http://www.xml.com/lpt/a/1633]

```
<content type="image/png">
iVBORw0KGgoA … TAAAAAElFTkSuQmCC
</content>
```

[http://www.xml.com/lpt/a/1633]

```
<content src="image.png" type="image/png"/>
```

[http://www.xml.com/lpt/a/1633]

# Atom Categories (13)

- Atom allows to assign categories to entries
  - each category element must have a term attribute for the category
  - an optional scheme identifies the categorization scheme (ontology, taxonomy, …)
  - an optional label attribute provides a human-readable label for the category
- Three different cases of categorization can be distinguished
  1. use a well-known scheme (such as *Dublin Core*)
  2. use a private but well-designed scheme (which has a URI and can be reused reliably)
  3. use tags without schemes, which then are little more than content labels
- Widely-known tags are not easy to handle [http://www.tbray.org/ongoing/When/200x/2007/02/01/Tag-Scheme]
  - they are more than just privately assigned tags
  - there is no formal scheme for them, just an emerging consensus





# Switching from RSS to Atom (14)

- Generate both feeds but serve RSS with a HTTP redirect (301)
  - old subscribers with broken clients can still use the RSS feed
  - old subscribers with correct clients will use the Atom feed
- Atom exposes more information than RSS (category for tags)
  - the mapping of publishing info to the feed has to be changed/extended
  - for standard metadata use Atom's built-in metadata elements
  - for application-specific metadata consider reusing an existing metadata schema
- Atom can be used to publish snippets as well as full content
  - content allows any type of content to be used and may contain a complete entry
  - summary allows only text and should provide a condensed version of an entry
  - some Atom sources publish two feeds for summaries and content
- Generate good Atom and downgrade it to RSS 1.0 & 2.0

## Podcasts (15)

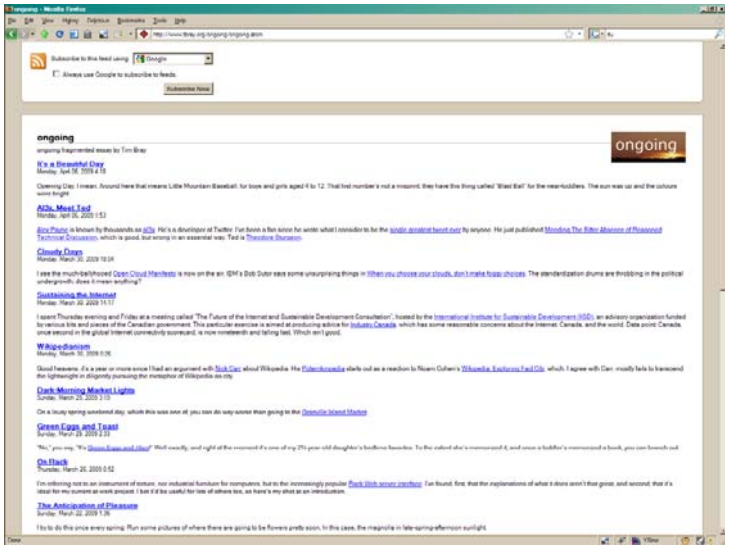
- *Podcasts* are simply *feeds with a number of additional elements*
  - the current "podcast specification" [http://www.apple.com/itunes/whatson/podcasts/specs.html] only allows RSS 2.0
  - in principle, there is no reason why podcasts cannot be Atom
- RSS 2.0's `enclosure` is used to point to the published item
  - *URL* points to the item itself so that it can be downloaded
  - *length* specifies the item's length in bytes
  - *type* specifies the items media type [Media Types] (video, audio, PDF)
- For business reasons, Apple wants podcasts to be "submitted to iTunes"
  - this ensures that the podcast can be found through iTunes
  - the iPhone currently only updates iTunes-published Podcasts



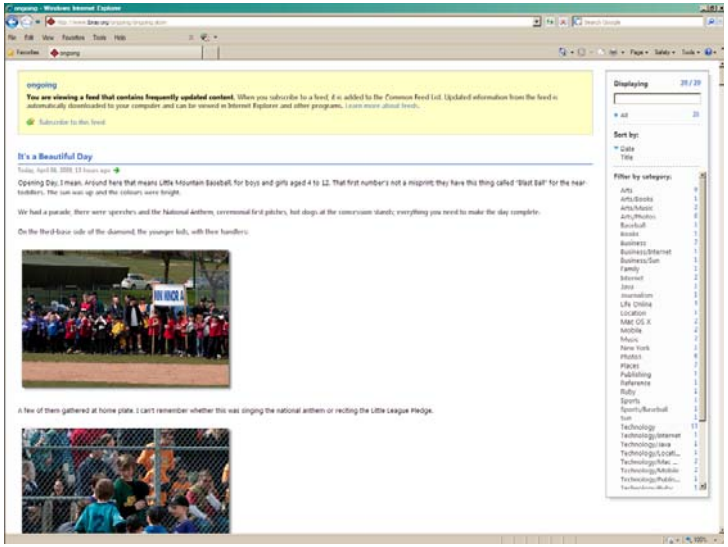
# Using Feeds

# Browser Handling

## Firefox (18)



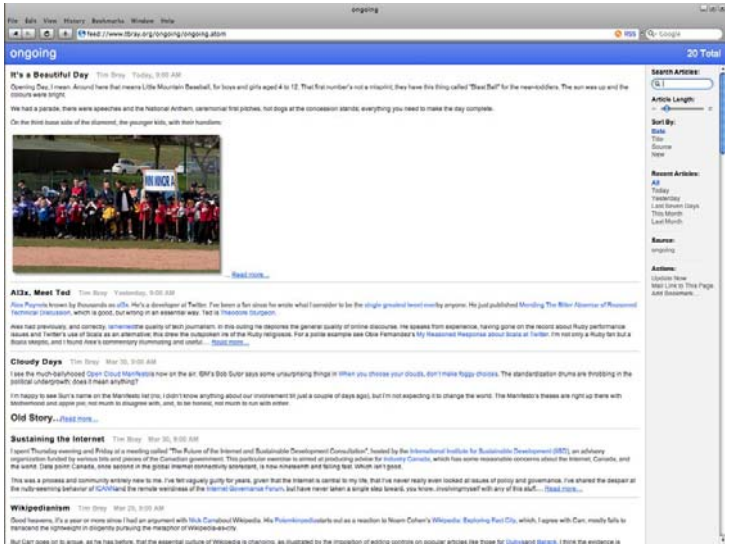
# Internet Explorer (19)



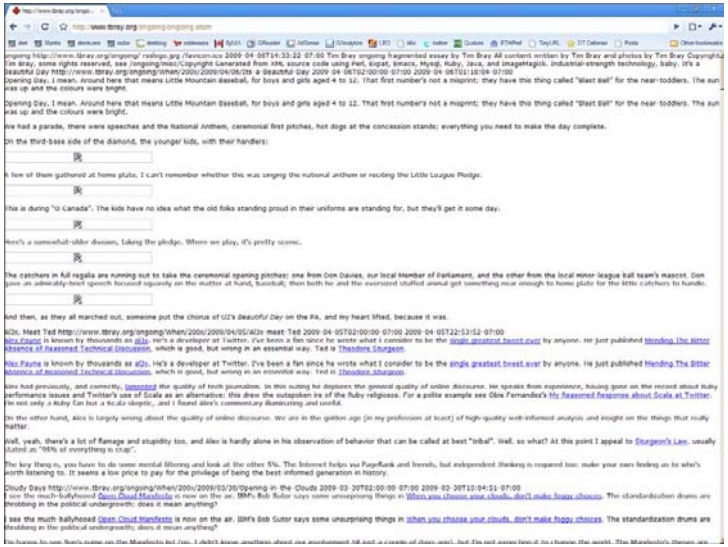





# Safari (20)



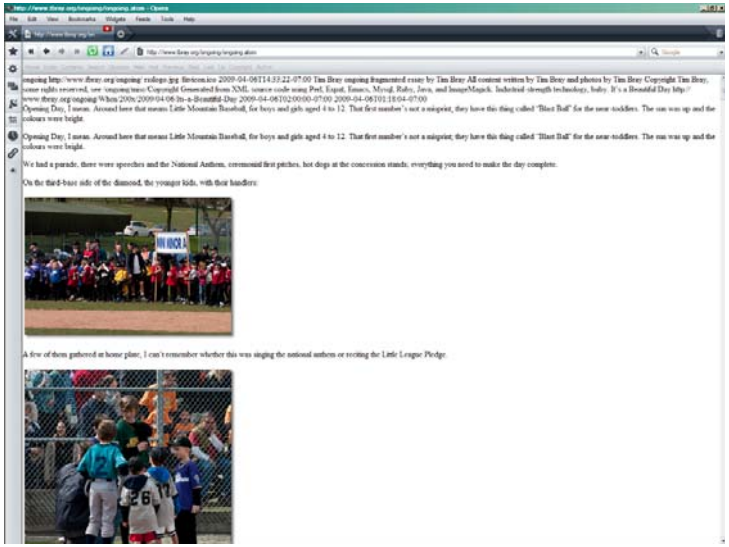
# Chrome (21)





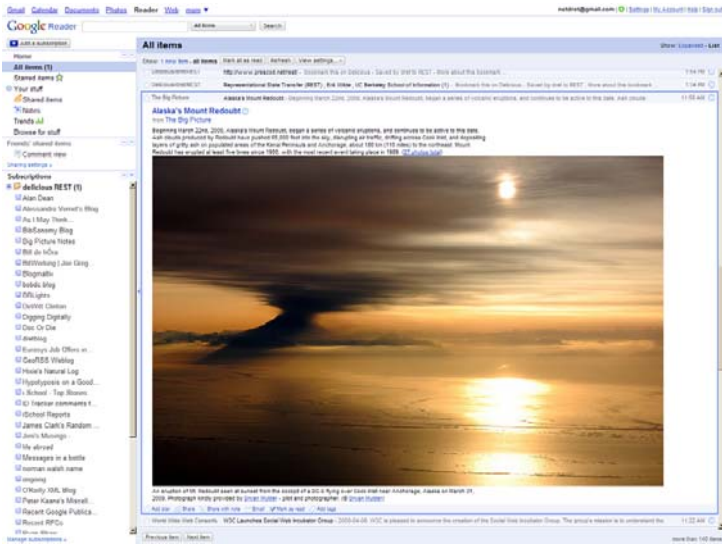


# Opera (22)



# Feed Readers

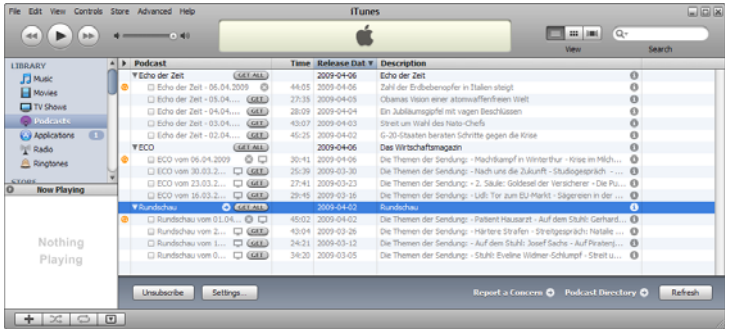
## Google Reader (24)








## iTunes Podcasts (25)



# Podcast Channel Information (26)

```
<rss xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd" version="2.0">
 <channel>
  <title>ECO</title>
  <link>http://www.eco.sf.tv</link>
  <language>ch-de</language>
  <copyright> &amp; &#xA9; 2007 SRG SSR idee suisse</copyright>
  <itunes:subtitle>Das Wirtschaftsmagazin</itunes:subtitle>
  <itunes:author>Schweizer Fernsehen</itunes:author>
  <itunes:summary>ECO zeigt auf, was die Wirtschaftswelt bewegt.
Zusammenh&#xE4;nge und Hintergr&#xFC;nde f&#xFC;r einmal nicht nur faktentreu
erz&#xE4;hlt, sondern Wirtschaftswissen
           mit Mehrwert f&#xFC;r jedermann. Aktuell und kritisch. Jeden
Montag um 22.20 auf SF1
       </itunes:summary>
  <description>ECO zeigt auf, was die Wirtschaftswelt bewegt.
Zusammenh&#xE4;nge und Hintergr&#xFC;nde f&#xFC;r einmal nicht nur faktentreu
erz&#xE4;hlt, sondern Wirtschaftswissen
           mit Mehrwert f&#xFC;r jedermann. Aktuell und kritisch. Jeden
Montag um 22.20 auf SF1.
       </description>
  <itunes:owner>
   <itunes:name>Schweizer Fernsehen</itunes:name>
   <itunes:email>eco@sf.tv</itunes:email>
  </itunes:owner>
  <itunes:image href="http://www.sf.tv/podcasts/data/eco_logo.jpg"/>
  <itunes:category text="Business">
   <itunes:category text="Business News"/>
  </itunes:category>
  <itunes:category text="TV &amp; Film"/>
  <itunes:explicit>no</itunes:explicit>
```





# Podcast Item Information (27)

```
  <item xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd">
   <title>ECO vom 06.04.2009</title>
   <itunes:author>Schweizer Fernsehen</itunes:author>
   <itunes:subtitle>Die Themen der Sendung: - Machtkampf in Winterthur -
Krise im Milchmarkt - Unternehmer mit Charakter</itunes:subtitle>
   <itunes:summary>&lt;p&gt;Die Themen der Sendung:&lt;/p&gt;&lt;p&gt;-
Machtkampf in Winterthur&lt;/p&gt;&lt;p&gt;- Krise im
Milchmarkt&lt;/p&gt;&lt;p&gt;- Unternehmer mit Charakter&lt;/p&gt;
</itunes:summary>
   <enclosure url="http://podcastsource.sf.tv/media/sf/podcast/eco/2009
/04/eco_20090406_222652_526k.m4v" length="116021491" type="video/x-m4v"/>
   <guid>http://podcastsource.sf.tv/media/sf/podcast/eco/2009
/04/eco_20090406_222652_526k.m4v</guid>
   <itunes:duration>00:30:41</itunes:duration>
   <pubDate>Mon, 06 Apr 2009 21:05:03 GMT</pubDate>
   <itunes:keywords>Wirtschaft, Schweiz, B&#xF6;rse, Finanzen,
&#xD6;konomie</itunes:keywords>
  </item>
```

# Simple Web Services (28)

- Syndication creates representations for universal concepts
- Atom adds some concepts to RSS's model
- Syndication revolves around the idea of interacting with items
- Atom-based interaction is one way of implementing REST
- For more semantics, Atom is only the foundation