INF 385q — Knowledge Management Systems

*Web Syndication as a Tool for Knowledge Management*

December 13, 2005

Caleb Ames
Web Syndication as a Tool for Knowledge Management
December 13, 2005
Caleb Ames

Web syndication, particularly as embodied by XML applications such as serialized resource descriptive framework RDF feeds, offers an attractive way to manage information and build communities whose interactions will serve as an increasingly important wellspring of knowledge. This paper will discuss what web syndication is, outline its history, and explain some of the technology that it is based on before turning to the topic of how it may serve as a tool for knowledge management.

What is syndication?

Web syndication is something of a misnomer that arose during the commercial development of the Web. It is simply what is done by syndicates, a word that, if it doesn’t conjure images of criminal cartels regulating their respective markets, perhaps evokes the ubiquity of Associated Press articles or the world of television where, say, Twentieth Television markets “That ‘70s Show” as an off-network sitcom to whomever has the money to pay for the license to broadcast it. Syndication is a kind of control—specifically, in our era, of information. It is also a kind of distribution. Its profitability comes from the ability to monetize other peoples’ use of protected content.

It is no surprise, given the profits syndicates promise, that in attempting to commercialize the Web, an effort was made to introduce syndication to the Internet. Unfortunately for those early pioneers, if control, content, and distributions are the linchpins of syndication then customers are the engine. Those early efforts to bring syndication to the digital realm failed partly because of the bugbears of control and customers (or the lack thereof) and partly because
of mismanagement and bad luck, so the content dried up and what was left in the empty husk of syndication were tools for the distribution of information.

Those tools have taken on a life of their own, and having been abandoned by the conglomerates with large capital holdings, have since been repurposed to the needs of individual publishers disseminating their own work. These small-scale writers – the journal scribblers and bloggers of every stripe, ranting and otherwise – have brought web syndication to a level of prominence that has, a decade along now, attracted the attention of business again. This time not as a tool for monetizing content, but as a way of maintaining relationships with their customers by building their own on-line communities and attempting to speak with a human voice.

*Push*

Of course it didn’t start that way. It began with the attempt to mold the new medium of the Web into something more like the traditional media of print or television: powerful content owners and a passive, paying audience. The technology for this model of unidirectional communication came to be known as “webcasting” – a name derived from the more staid “broadcasting” – or more generally as “push technology.”

The company that came to embody “push” was “PointCast,” founded in 1996. Pointcast began by streaming personalized information over its proprietary network software to corporate offices. Soon the company teamed up with Netscape to offer a Web-based service that would push information and advertisements to users’ screen savers. It is difficult now to imagine the vertiginous mania this software inspired - the CTO of Sun Microsystems, Eric Schmidt, declared that the software has the potential “to be as huge a hit with consumers as Web browsers are now” (Rigdon, 1996). Wired Magazine went one better, “the Web browser itself is about to
croak. And good riddance.” In its place they envision a world where information is everpresent and insistent on being noticed, a realm of relentlessly prodding content providers imagined as an information paradise. (Kelly & Wolf, 1997)

The excitement has a number of causes. The most prosaic was the appeal of familiar models of funding. The Wall Street Journal, for example, saw the Web as a space desperately in need of advertisers and declared that the Internet “has been a medium in search of a viable business model. Now it has found one: television.” (Bank, 1996) PointCast could provide a verifiable audience to advertisers, a guaranteed set of receptive “eyeballs.” Another was the belief that operating a browser was too difficult, and a simpler, TV-like interface would appeal to the masses. Some, however, pointed out that browsing web pages is a burdensome activity. This is the “hunt and gather” problem, in which each user must cultivate their own collection of useful websites which must be periodically visited in order to harvest new information. This can take evolve into its own ritual – a daily walking through a handful of favored websites combined, perhaps, with checking one’s email, done with a blank sort of tooth-brushing regularity. Just as too many emails can flood an inbox, however, so too can one accumulate too many sites, particularly when those sites update frequently. Eventually you run out of time to check them all, and their informational content sifts past like sand through a sieve. Alternatively, you might not have enough sites, or the right kinds of sites in your personal collection, and a prepackaged bundle of information in the form of a “channel” would be an appreciated guide.

It wasn’t long before a host of imitators began following PointCast’s lead, the two most prominent being Netscape and Microsoft who used “push” software as an extension of their fight for supremacy in the browser market. Netscape announced “Netcaster,” push software that would
automatically deliver channels of information to the Netscape Communicator client. Microsoft shot back with the Channel Definition Framework (CDF) which worked with the Active Desktop software released with Internet Explorer 4.0 and Windows 98 to deliver “push” media straight to the desktop without needing to open a browser window. PointCast began to struggle, however, from a combination of technological and management missteps, and as quickly as it began, the mania faded. PointCast died an ignominious, whimpering kind of undeath; it was bought out by venture capitalists who merged it with another firm in 1999 and PointCast sent out its last broadcast in 2000. (Junnarkar, 2000) With “push” left without a champion or hype, the two software giants quietly let it go. Microsoft allowed CDF to disappear into the dark bowels of its developer website unseen by common users and Netscape abandoned Netcaster outright.

Underneath the clash of software companies, however, a tangle of technologies were cross-pollinating to form a more enduring basis for syndication than the proprietary “push” formats. The most important of them is XML, or Extensible Markup Language, a text format for data exchange developed near the end of 1996 by a W3C working group led by James Clark with Tim Bray and Michael Sperberg-McQueen serving as the specification’s editors. (W3C, 1996) XML serves as a relatively simple toolkit for constructing languages that describe structured documents. Its openness and flexibility as a format encouraged further innovation, and in 1997 XML was used as the basis for the Resource Description Framework (RDF) and the Scripting News format. (W3C, 2005 and Winer, 1999)

RDF is another W3C maintained project focused on using XML to describe an ontological system, or, more simply, as a language for metadata. Scripting News was the first of the XML formats developed for syndicating content over the web and was designed by Dave
Winer of Userland Software. These two technologies have danced with each other intermittently over the past eight years. They first brushed together when Netscape developed and released the RDF Site Summary (RSS) 0.9 format in 1999 as a way of adding channels to their portal, the My Netscape Network. (Marsden, 2001) RSS 0.9 was intended to be a metadata format and in that spirit included the barest of RDF headers. Under the direction of Dan Libby, however, it evolved towards straight content syndication and by RSS version 0.91 the RDF references were removed (and the specification re-dubbed “Rich Site Summary”) and a number of scriptingNews tags were included, bringing the two formats closer together. (Libby, 1999) In 1999 Netscape abandoned RSS and it was picked up and championed by Userland. Meanwhile, in between July and November 2000 a group calling itself the RSS-dev working group led by Rael Dornfest forked RSS by creating an entirely new format that they named RSS 1.0 (a controversial name that resurrected the acronym “RDF Site Summary”) which again included RDF headers. Besides bringing back RDF, another part of the specific appeal of RSS 1.0 was that it now included XML namespaces which opened up the possibility of expanding the specification through modules. On the other side of the fork, Userland developed RSS 0.91 up to version 0.94 and in 2003 Dave Winer, now operating on his own, released another RDF-free syndication format based on RSS 0.92 called RSS 2.0 – which Winer declared stood for “Really Simple Syndication.” (“RSS Specifications”, 2004) This specification is now copyrighted by Harvard and is frozen so that no further work will be done on it. Finally, the development of yet another syndication format, Atom, began in 2003. By 2005 Atom 1.0 was published. It also does not make use of RDF, but it does support more of the features of XML and has somewhat better defined elements. For example, Atom feeds reside in an XML namespace, include a RelaxNG
schema, and define elements for full or partial content data. Despite this welter of incompatible specifications, syndication feeds have found increasing popularity. According to Lee and Goodwin the number of RDF pages, which include RSS 1.0 feeds, have increased 220% from 2003 to 2004 and have attracted a number of traditional publishers such as the Wall Street Journal, National Public Radio, and Reuters. (Lee & Goodwin, 2005)

Before the traditional publishers had gotten into the game, however, came the bloggers. The proliferous passions of the RSS developers kept their various formats from suffering the same fate as the forgotten CDF, and the bloggers made the formats relevant by using them. Mainly they used them to track other bloggers, as a solution to the hunt-and-gather problem. By eschewing commercial content providers and their portals the bloggers avoided one of Push’s weakness – the early resistance that users had to letting expensive bandwidth be used for pushing ads – and emphasized web syndication’s function as a distribution mechanism for aggregating content or headlines. Syndication no longer seems like an appropriate term for the kind of time-based content streams that RSS represents, but linguistic inertia ensures that the term, somewhat mutated in meaning, remains in place for now. As companies return to the idea of syndication, however, that terminology will come under pressure to represent the needs of the corporations that use it. For example, Microsoft promises to release an RSS implementation next year that they will call “web feeds.” (Clarke, 2005)

Applications

Regardless of what they are called, web feeds have three defining characteristics. They are chronologically ordered, they contain metadata, and they are a “pull” media. Out of these three traits a great many applications may be built. The most prominent RSS application is the
news aggregator. Win Treese, in enumerating the benefits of a traditional newspaper’s “user interface” writes that it is “easy to navigate, it’s easy to ignore what you don’t want, and it’s fairly easy to find what you do want. It’s also very fast—it doesn’t take long to move to the next article.” (Treese, 2003) Consuming information on the web lacks all of these virtues. It takes time to find the information, it takes time to locate the navigational hyperlinks to move to new information, it takes time for the server and client to communicate with each other, and because of all of the preceding there is a significant transaction cost imposed for retrieving an article whether the user chooses to ignore its content or not. News aggregators facilitate information discovery by mitigating some of its costs, and they do so by relying on RSS’ role as metadata and by the fact that RSS transactions are user-initiated. What was called “pull” earlier is now called “subscribing” in the world of RSS feeds, and means that RSS-aware software will request and retrieve the feed on a predetermined schedule so that the user need not wait for the information to be fetched. Feeds also contain metadata about the articles that are being browsed, usually in the form of a title, author, date, and a short fragment of the content. This metadata provides a useful information surrogate for the articles and combined with the no-wait aspect of pre-fetching feeds means that news aggregators have the potential to compete with the speed and convenience of the newspaper’s user interface.

Besides informing people, RSS can be used to inform machines. Because web feeds conform to a predictable XML format, applications may be built that “read” the metadata contained in them and use it to make decisions. An example is provided by Liu, Maly, Zubair and Nelson in their discussion of methods for synchronizing an Open Archives Initiative (OAI) repository with a metadata service provider. (Liu et al., 2003) The problem they faced is that as
the document repository changes the service provider must periodically “re-harvest” it for metadata in order to have a fresh record of the repository’s holdings, yet harvesting a repository that hasn’t changed is a waste of time, bandwidth, and computing cycles. One possible solution that they offered is to have the data provider respond to queries with an RSS feed that describes the repository’s update schedule, providing the service provider a hint for how frequently it should harvest it. This is perfectly analogous to the human being who uses an RSS news reader to discover how often they should stop by their favorite blog but demonstrates how even semi-structured metadata can be a boon for a variety of automated processes.

RSS can not only deliver information but can also be a source of information. Jarrett and Dennis have developed a prototype application called “BuzzMaps” for “collecting news items, capturing social activity, and enabling social translucence.” (Jarrett & Dennis, 2003) To that end, it records information on users’ participation levels and popularity in a shared space and displays it in an unambiguous, visual fashion. In this framework, RSS subscriptions are a kind of attention or “buzz” that can elevate the status of the community members who are their target. This is an intuitive measure of popularity because, although one might visit a web page out of idle curiosity or as a mistake, or perhaps bookmark it to find it later, an RSS subscription represents an invitation into one’s information space. And once the feed has been invited onto a news reader, it will stay and continually pop up new items until it has been unsubscribed. The act of remaining subscribed, then, indicates a level of endorsement for the feed’s content that is otherwise difficult to capture and because of the “pull” nature of feeds the content author receives a continual stream of feed requests that represent a kind of vote for how well received his or her content is.
Finally, there are some novel applications based on the fact that RSS feeds are time-ordered, which means also that they are expected to change over time. Dave, Bogen, Karadkar, Francisco-Revilla, Furuta and Shipman have written about using RSS feeds as the foundation of a dynamically growing hypertext collection. (Dave et al., 2004) They are suited to this purpose because they “de-hypertextualize” the Web in order to fit it to a linear metastructure amenable to collection development; “Seen from the vantage point of a feed the complexity of the underlying link structure is subsumed for the benefit of the newly provided context.” This context is the feed author’s “path” through the web. Using a system called “Walden’s Paths,” a user may layer their own paths over that of the feed authors by using information filters to extract feed items that are of topical significance to the user, or that meet their criteria for age or novelty. These items become “stops” in the user’s new custom-defined path and the filters may be preserved so that as the source feeds add new items those items that pass through the filter appear as additional stops in the user’s path. The overlapping of various users’ paths “re-hypertextualizes” the underlying Web material, providing a new context for understanding that information as a new, coherent collection.

**Knowledge management**

Knowledge thrives in the interface between people and the world. It is the applied side of information and fuses “how” things are done with memory, judgment, and prediction. Of the RSS tools already mentioned, three have knowledge management applications.

Walden’s Paths uses feeds in order to capture the judgment that put them together. A feed from slashdot.com, for example, is imported because the system managers trust Slashdot’s
editorial judgment in releasing items that share some topical similarity—that the items will be about technology or tech culture.

BuzzMaps are explicitly targeted at enhancing predictive utility which is a particular kind of “how to” knowledge, specifically, how to predict which items will be the most relevant to the user before they are selected. Buzzmaps rely on the notion of “socially translucent systems,” a poetic turn of phrase used by Erickson, Smith, Wendy, Laff, Richards and Bradner (1999) to describe systems that make social cues visible. The benefits of such systems are that, by introducing accountability to Web-based interactions, they bring to bear the pressure of social norms which in turn have a civilizing influence on discourse. Web syndication can participate in such systems in a limited fashion as a kind of social proxy—because they emanate from a particular Web site they can act as a stand-in for their author, and because they must be refreshed periodically they can provide snapshots of the amount of attention being paid to an author. These traits provide some basis for certain social behaviors, such as ignoring feeds from certain domains, and thus, certain people, or gauging popularity by tracking feed retrievals.

By far the most common use of web syndication in knowledge management, however, is as an aid to conversation, as in the news aggregator mentioned above. Web feeds can be used to speed up communications and make visible comments that would otherwise not be read. The increase in speed is due to receiving timely notification of new items from the feed reader, and the increase in visibility is due to the fact that feeds make it easier to keep track of the postings of more people which may increase the audience for those posts. In particular, in a business environment feeds— in combination with Web logs—can liberate the knowledge of lower-tier staff by providing them a mechanism for following the work of their fellow employees and to publish
an up-to-date feed of their own activities. Within the right culture of knowledge sharing, this kind of grass-roots communication has the potential to drive knowledge transfer between departments and units that would otherwise not talk to one another.

In the end, then, this will be the greatest impact of Web syndication: the network of connections between people that it will help to weave together. Where it has failed as a means to sell content along multiple channels, it has succeeded as a way for communities to keep track of each other and, in combination with newer formats such as Trackback, to communicate with each other while still maintaining an individual sense of place.
References


