

The Return of Pointcast: Why the US Military should Immediately Implement RSS for
Information Dissemination
by: LT John Stafford, USNR

Local operations off Hawaii. USS Ramage (DDG-61).

Senior Chief Wilson just wished the promotion results for this year's Chief board would come out. That way people would quit asking -- Communications LPO was a popular person this time of year. Of course, the Executive Officer and Commanding Officer would get the news first -- but that didn't stop their interest either. Internet access on a nondeployed destroyer was almost nonexistent, about two hours twice a week (if ship's heading allowed). It wasn't even enough to clear the steadily mounting email queue. Tonight when the link was established they would need to download the list for the entire Navy (over a thousand people). Even if they'd asked squadron to look up ship's company, that was the only way to guarantee the training team and helicopter detachment onboard would be covered. Was there a way to ensure this high priority information was at the top of the queue once the internet connection was established?. How could bandwidth be conserved without ignoring the valuable members of Team Ramage whom personnel classified as part of their home units?

A small town in Iraq. 115th Military Police Company. Rhode Island National Guard.

1st Lt Barrett was glad to finally be settling down. The first few weeks in Iraq had been one dusty village after another as the carefully laid deployment maps had been adjusted to face reality on the ground. A quiet town, the town council had once been a model of cooperation. American troops had even entirely ignored it for several months as attention was focused in a more difficult area 100 miles away. But last month, ambushes and sabotage suddenly began on a small scale. The docile and supportive town council had become frustrated and angry pointing to missed deadlines for training and reconstruction efforts. Some of their demands were clearly beyond anything that American forces would have agreed to. But some seemed reasonable. How could 1stLt Barrett access the message traffic documenting decisions agreed to four or five predecessors before and a year ago? How could she build on the logistics and intelligence lessons of the past year? Access to message archives would be key, but it seemed too difficult to trace all the units who had been in the town. And most unit message archives were now in the United States, long ago purged by the communications centers in theater.

Indian Ocean. USS George Washington (CVN-73)

LT Fernandez was stumped. As Information Technology Officer, he frequently received complaints about internet access. Even the CO, with the highest access priority complained about long delays every time he clicked on a webpage link. Web access to sailors E6 and below was entirely blocked but it seemed to have little effect.

Like all aircraft carriers, the ship had a web proxy server installed. This proxy server served as an intermediary for all web site requests and attempted to enforce the priority tiers. It cached copies of recently downloaded pages and would respond with those if sufficiently fresh. But the statistics weren't good. Too many websites were not proxy-friendly, or people would refresh their screens anyway, assuming the information was out of date.

Looking at the logs, the pattern was clear. Much of the data demand at the unclassified level was predictable: supply information, personnel data, hometown news, sports (NASCAR was a favorite). At the secret level, it was even more clear: briefs, messages and policies from various fleet commanders, and intelligence data.

How could LT Fernandez end the situation where the World Wide Web had become the World Wide Wait? Could access to the data already downloaded to the ship be made available to the entire crew?

A common thread. A possible solution.

The common thread in each of these three problems is information. In each situation, publicly available (at least inside the military) information has difficulty making it to the front lines that need it most. Valuable information pipes are clogged with interactive data requests replete with timeouts and lengthy delays when information is received, instead of being filled with a steady stream of information useful to the warfighter.

The traditional response to this is knowledge management. In theory, information systems should be carefully designed and architected to maximize data throughput and minimize any overlap of data. This works well in a narrow limited range such as a missile communicating with a fire control system. The parameters are well defined, security and reliability are paramount, and years of engineering and testing go into a system such as this. In most cases, it works extremely well once the system is fielded.

But the Internet violates almost all of the tenets of information management. By all rights, it should be absolutely useless. If it had been developed as a military system, it would have never passed operational testing -- ill-defined pockets of information with little standardization which waste bandwidth while providing poor reliability in low-bandwidth military environments.

Instead of being useless, it is a key force transforming the military. Systems commands perform remote troubleshooting, with land-based engineers looking at digital pictures and equipment charts over the shoulder of a technician 3000 miles away. Daily status messages are replaced by powerpoints which synchronize across the battlegroup, yet even these powerpoints are being steadily replaced with web-based systems like KWeb which provide a "living brief" which can be continually updated by information owners.

The World Wide Web, and the Internet technologies it rides upon, is chaotic and frustrating. But its existence as a standardized, open platform has let a million flowers bloom,

including thousands of government sites. No longer does an information owner need to worry about the consumer. They still should -- frequently, assumed bandwidth and connectivity do not exist and thus the continuation of the stream of CD-ROMs that existed before would be most appreciated. But they trust that somehow, someday, the soldier on the front line will access the website and get his W2, or his billet list, or even the number of the base hobby shop. And thus both soldier and content provider press the providers of communications such as LT Fernandez and Senior Chief Wilson for just a bit more access to that precious, hoarded commodity.

It is clear that there is a level still to come beyond KWeb in which supply, tactical, intelligence, and command systems will directly feed a common synthesizing interface without user intervention. But this is a hard, difficult problem requiring difficult integration. The future for KWeb is business intelligence engines, and Oracle backends, and integration to legacy databases, and automated analysis and processing of message streams. A primary difficulty is that systems are designed to requirements, but it is very difficult to require compatibility with "the future". Progress is inevitable, yet inevitably different than we envision.

While we can envision such a system using the resources (bandwidth and connectivity) of the web, we cannot say that this future application will be of the web. It was the preexisting resources of the web such as HTML editors, web servers, and web browsers that allowed KWeb to go from wargame to deployment in less than a year. The next level cannot use these low cost, readily available, commercially developed pieces making the future both more expensive and more risky.

In fact, the future looks much like the past in time and scope, just with faster updates, prettier graphics, and more linkages. Such a program will be far too complex (and fragile) to allow riders to tinker away at the Admiral's suggestions during predeployment training. So

hopefully the Admiral two steps before who approved the capabilities document was right. It is clear that the web cannot meet all of our requirements. It might make 50% or 70%. But it might enable them in 1/3 of the time. And three or four years from now commercial web-based technologies may have advanced enough to solve another 20% without requiring a complete rewrite like most proprietary systems. By which time, the future will have moved forward yet again.

There is an alternative approach. Between the sheer anarchy of the web and the configuration control boards and ten year plans of the not-web, an alternative exists. That alternative is the semantic web.

The First Step to the Semantic Web

*The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.*¹

The semantic web demands that the data of a website be turned into information represented in a standardized format. This transformation is invisible to the end user but allows computer programs to take meaningful action based on the site's content. For example, a doctor's calendar on one web site may contain "tags" such as <date> allowing an automated appointment engine to

¹ Berners-Lee, Tim, James Hendler, and Ora Lassila. "The Semantic Web." Scientific American. May 2001. p.2
<http://www.scientificamerican.com/linktous.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>

distinguish numbers which are dates from the numbers in the street address of the office.

The semantic web has had a difficult time in adoption, primarily due to the social issues. The technology is relatively simple, just human-readable tags (which can be added by any text editor) attached to content. However, a consensus has to be reached on each type of information to determine the required elements of information and tag names. One nascent standard is FOAF² standing for Friend of a Friend which is used to represent personal contact information and linkages between people (with appropriate encryption and security). Instead of using a proprietary service such as Friendster or Tribes, which requires separate data entry for each service with no synchronization or updating between competing services, a person writes up a small file and publishes it to their website. Then an external FOAF search engine (of which a multitude could exist with specialized feature sets and purposes) gathers these files to provide networking contacts or to ensure a personal address book is up to date. But since the standard and files are open to the Internet, they may be used for new and innovative purposes at any point.

But the sluggish adoption of FOAF (as of February 2004, there are less than 30,000 FOAF files compared to over 2,000,000 registered users of Friendster -- a traditional social networking site) is a clear problem. And it is one of the most popular of the semantic web standards. The difficulties are twofold: first, the semantic web must be integrated into content generation applications. There is little incentive for social networking sites to generate open standards based FOAF files since their business model is based on getting people to participate in a closed environment with their advertising and premium services.

²<http://xmlns.com/foaf/0.1/>

Yet the semantic web has generated one extremely important standard in the field of knowledge management and dissemination: RSS.

RSS has stood for several things in its brief life: RDF Site Summary (RDF being the core format of the semantic web), Rich Site Summary, and today Really Simple Syndication.³ Its original purpose was extremely simple. It provided news headlines which could be displayed inside of a small window in a web portal. The web portal would retrieve a file which was automatically populated by new news items. The user would see a list of headlines, and could click on any one they found interesting to read the entire article.

From Pointcast to Weblogs

This was controversial because of a service called Pointcast. Pointcast was a very popular application from 1995 to 1999. It wasn't web-based -- instead it downloaded content similar to web pages into a proprietary application on the user's machine. The content primarily consisted of articles from newspapers and magazines such as the New York Times (most articles from the print version of the Times were provided). It also had comprehensive stock market data including graphs of stock prices and financial wire service news for each stock.

Once the content was downloaded on the local machine, it could be read in a very rapid manner. This ability to rapidly read content especially appealed to users without high-speed internet connections (which were unavailable to most users outside of large cities or universities at

³ Reagle, Joseph. "Web (RSS) Syndication History."
<http://goatee.net/2003/rss-history.html>.

the time) since information that was downloaded in a constant stream for 10-15 minutes could be rapidly scanned in 2-3 minutes. Even waiting for the constant stream to finish was much quicker than the normal cycle of waiting 15-20 seconds for a new page to load after each click.

A consistent user demand was to add new sources of content rather than the proprietary ones that Pointcast provided. Microsoft and Pointcast teamed to create the Content Definition Format in March 1997, a proprietary standard, to provide this capability -- a list of headlines which could be downloaded into Pointcast which each corresponded to a web page. The W3C, the standards body responsible for the core standards of the World Wide Web, published nearly at the same time a draft of the RDF standard which is the core element of the semantic web. Netscape, having an intense rivalry with Microsoft, extended the RDF standard to create the RDF Site Summary standard rather than using the existing Content Definition Format standard in March 1999 for its new portal.

Development continued in 1999 and 2000 (though Netscape lost interest in using the format) responding to user demands to go beyond headlines and directly package graphics and text as part of the format. In August 2002, the RSS 2.0 standard⁴ was published.

An RSS document is written using XML, the eXtensible Markup Language. XML is a format that allows documents both human and machine readable to be defined and presented.

The basic element of an RSS document is the channel. A channel can be equivalent to a website or a portion of a website. The required elements are a title, a link to the corresponding website, and a description. Optional elements allow information on the editor, webmaster, and copyright details to be stored.

A channel contains one or more items. These are the actual bundles of information that the

⁴Winer, Dave. "RSS 2.0 Specification." <http://blogs.law.harvard.edu/tech/rss>

end user desires. Each item can be as simple or as complex as a user desires, however the usual practice is for each news article or message to be a single item which allows further sorting and searching to the minimum unit of information.

The key elements of an item are the title, the description (which despite the name may contain the entire message), and the link which points to the full item on the web. A key element is the category which allows one or more taxonomy elements to be identified which apply to the information. This helps identification of the item in future searches.

Another element of the item is the guid which is a unique string identifying the document. This supports multiple levels of distribution or archiving of a single document without unnecessary duplication. By convention, it points to the authoritative source of the document, although the copy received may have come from a more convenient location.

A key element is the enclosure. This allows rich media such as audio or video to be included as part of the information bundle. The file size and file format of the enclosure are included, since these may be factors in the decision whether the enclosure will be automatically downloaded or whether downloads will wait for a user request.

A sample RSS 2.0 file is as follows:

```
<?xml version="1.0" ?>
<rss version="2.0" >
<channel>
<title>John Stafford's Website</title>
<link>http://www.thenextamerica.com/</link>
<description>My personal weblog</description>
<language>en-us</language>
<managingEditor>stafford@fastmail.fm</managingEditor>
<webMaster>stafford@fastmail.fm</webMaster>
<ttl>40</ttl>
<item>
<description><b>RSS</b> distribution of information can revolutionize knowledge management on the front lines. For more information see the specification <a href="http://blogs.law.harvard.edu/tech/rss">
```

```
here</a></description>
<pubDate>Mon, 23 Feb 2004 17:56:02 GMT</pubDate>
<guid>http://www.thenextamerica.com/MilitaryRSSSample</guid>
<enclosure url="http://www.scripting.com/mp3s/weatherReportSuite.mp3" length="12216320" type="audio/mpeg"
/>
</item>
```

While complicated to the uninitiated reader, the sample clearly shows how a simple item of information can be wrapped in useful metadata. This would be extremely difficult if it had to be done by hand every time. Fortunately, the metadata is normally added automatically by the software that is used to publish the data on the web, in much the same way programs like MTF are used to add the header fields to a military message.

XML (and thus RSS) relies on the concept of a schema, which are the rules that an XML document must follow to be correct. Thus, as long as a document can be validated using the schema, any compliant application should be able to read it. One of the principles of XML and RSS is extensibility which allows specific applications or user groups to add their own elements for their own purposes. These new fields should be ignored by any program that doesn't understand them. To use these new fields a unique namespace is defined (part of which is an owned domain name to prevent conflict) so that different groups may add elements of the same name with possibly different behaviors. If an addition is widely accepted, and of a general nature, it may be incorporated in a future version of the specification and its accompanying schema.

The infrastructure required to publish documents using RSS is minimal. In fact, it uses http, the hypertext transfer protocol, which underpins the world wide web. Thus, any existing web server can host an RSS document and become an RSS Host. The harder part is integrating RSS creation into the existing system for creating web content.

The most common way is by using weblog software. Often shortened to "blog", a weblog

is normally viewed as a single page of short entries in reverse chronological order. Depending on volume, the page may cover a week or only a single day. Archives of previous articles that have been removed from the front page as they age are also available. Weblog software handles everything required in constructing the pages. Once a color scheme and format have been established, only the text of a new entry must be added (normally using a simple web-based form) for the entire site to be modified reflecting the new information. A weblog can be setup by a novice user in less than 10 minutes. Over 1.7 million weblogs exist as of February 2004.⁵ Common topics are politics, technology, or merely items of a personal nature. The impact ranges anywhere from a daily news publication with similar reach to many traditional newspapers (<http://www.andrewsullivan.com/>) to modern teenage diaries. Due to extremely low cost and ease of use, they are popular among young people and technical experts alike.

Weblogs may average between 1 to 20 separate postings each day. Or a weblog could go several weeks without any activity. Most people who have active weblogs also read other weblogs and share any unique finds or put their own perspective on a piece of news. It can be slow and tedious to check each individual website of interest on a daily basis for new content. Thus, bloggers like Robert Scoble (an extreme case who reads over 1200 weblogs daily) have quickly turned to RSS aggregators for help.

An RSS aggregator is a special piece of software which manages RSS channels. Once the list of channels has been entered, the software will automatically check each site for information, normally every few hours. If it finds new information it downloads it until the user is ready to read it. It may be a stand alone program or it may integrate directly into a user's email program such as Microsoft Outlook. A user can then rapidly scan the headlines of each channel, reading the items

⁵ <http://www.technorati.com/>

of interest. RSS is essential to an aggregator because it standardizes each weblog. When visited with a web browser, there are literally thousands of different formats and color schemes for weblogs with new ones being created every day. RSS allows the format of a website to be as personal as a person wants, while still being accessible to power users. There are only a few popular brands of weblog software and all but one currently support RSS.

It is unfortunate that many portions of the military (including the entire Navy) have gone to locked down desktop environments where software such as RSS aggregators cannot be evaluated or demonstrated. Widespread support at the grassroots level (similar to that originally shown for Netscape in 1994 and 1995) has been essential to RSS becoming part of the corporate information strategies in major companies. Without the support of power users familiar and experienced with the technology, there is little chance of it reaching many desktops until Microsoft includes it two generations beyond Windows XP (Microsoft has no plans to put an aggregator into Avalon). Thus, if current deployment cycles for operating systems continue it could be 2012 or later until many employees can test the effectiveness of a client RSS aggregator.

Fortunately, an RSS aggregator does not have to be a piece of software loaded onto a client. In fact, one of the most basic (and most useful) aggregators is an RSS to email gateway. This is a web server where a user can input a channel that they want to monitor and an email address. Whenever new information appears, it is automatically emailed to the person by the server-based aggregator. A server-based aggregator can combine various channels into a single channel such as all weblogs about a certain political candidate. It can also apply filters, such as creating a customized channel that only contains items with a keyword.

Now that RSS has been described, how could it help each of our three scenarios?

The first scenario is the shipboard download of promotion board results. It would help Senior Chief Wilson if an RSS channel existed for promotion board results. Instead of needing to navigate through the BUPERS website, an RSS aggregator would perform a simple check of whether there was new content in the channel to answer the question and would automatically download the message upon release.

His bandwidth concerns could also be mitigated in an alternative fashion but it would require the promotion board result format to be adjusted. It would need to be broken up into individual entries by person, or a separate entry per command (by unit identification code). Senior Chief could predefine a server-based filter on the channel that would only transmit the new channel unit identification codes of interest to the ship. 95% of the download would be eliminated.

In the second scenario, 1stLt Barrett wants to access old messages from her predecessors in a portion of Iraq. Currently, communication centers discard all messages after as little as 30 days. While this is primarily due to the massive flow of information, it can also be traced to lack of interest in these historical messages. Currently, they can only be accessed by addressees, who already have a local copy, and require significant manual intervention. Even fleet commanders may only have a two or three year record of messages received. Until the end of Vietnam, a copy of every naval message was retained by the Naval Historical Center. Today, they are ephemera with most messages quickly lost to history.

An alternative would be the establishment of regional message archives. These could contain all messages (and receive them as part of message processing) or be a special addressee (such as MESSAGE ARCHIVE OPERATION IRAQI FREEDOM). These archives would be open to anyone in the government with a need to know and appropriate security clearance and

would not contain PERSONAL FOR or other messages requiring special handling. Once established, RSS feeds could then be used to analyze and disseminate the information. For example, a "Lessons Learned" feed could automatically email all messages with those words in the subject to interested units, in or out of theater. This would allow units to self-synchronize since they could immediately learn from each other rather than waiting for analysis by higher authority and later propagation. Peer to peer propagation of messages in this fashion is contrary to current practice, but the custom of isolating messages to a small group is a relic of previous technology. Today, there is little reason why any message should not be releasable to the entire government. A similar search capability could be performed by custom software on existing message formats, but conversion to RSS (relatively easy since many fields correspond) allows the use of readily available commercial tools.

The final scenario of limited access with a continuing bandwidth crisis aboard an aircraft carrier requires new technology. But this is where RSS really begins to shine. What could be called an "RSS Proxy" should be installed onboard to manage RSS traffic.

The RSS Proxy would be invisible to users as a web proxy is. It would save bandwidth immediately, since information received via an RSS feed normally has unnecessary graphics and advertisements removed. When an RSS channel is accessed the proxy would retain a copy of the information as well as annotating when the RSS channel was last checked. It would also record the user who accessed the information.

The next time a user requested the information, it would provide the information locally without using any bandwidth. The administrator could examine what channels were used, how many users were accessing a channel, and then determine how frequently the channels will be updated. Daily would be appropriate for channels in the United States; hourly or more quickly for

channels of battlegroup interest. The administrator could also prioritize channels (channels the CO monitors have top priority; NASCAR has medium priority on weekends but lowest priority during the week). Unlike most web pages, the modification times and dates on RSS channels are kept with extreme accuracy and the channel may contain specific information about how frequently it should be checked. Using RSS the information downloaded remains useful for days or weeks, and bandwidth is never wasted on old information. Due to the nature of the web, a regular webpage must be purged from the cache after 6 hours or so (a day at most) and be entirely downloaded upon the next access. A web cache has no way to distinguish between information with long-term value and short-lived graphics and advertisements. Due to this there are often broken links and missing graphics on complex websites in the shipboard environment.

With the RSS proxy installed, anyone onboard could access RSS information. If no one with "internet" access had requested it, the channel would simply be unavailable to lower ranking personnel. However, once downloaded all channel information would be available to everyone onboard. Instead of each person waiting through interminable delays trying to access offship data, RSS would maximize the throughput of the internet connection by providing either complete information or nothing at all. The enclosure feature could be used for high-bandwidth attachments such as pictures or video to signify that rich media is available and then to let the administrator manually decide if the attachment should be downloaded (or to refuse to download it altogether). RSS traffic can also be easily distinguished from standard web traffic so it can be both prioritized against itself (determining RSS channel is more important) and against other IP traffic such as email and interactive web traffic.

An RSS proxy would give an aircraft carrier (or any bandwidth-limited end user group) rapid access to downloaded data. It finally allows the extremely valuable internet link to be tuned

to maximize throughput and to give priority access to information important to the ship's mission. It would also help 1stLt Barrett in the second scenario since requested portions of the message archive could be loaded directly into the proxy using DVD-ROM or tape for use and searching by the entire unit while they are in the area. While software of this nature currently does not exist (due to lack of utility when high-speed internet links are permanently available) research and development in this area could benefit the military and be a unique technological advancement in the field to contribute back to the civilian technical community.

Another area in which the RSS proxy could help is in outgoing messages. Most equipment casualty reports and logistics requests have a large number of "carbon copies". These carbon copies could be eliminated by posting these messages to a regional message archive. Then rather than have yet another email which is eventually sent to thousands of people, the specialist providers could work directly from the searchable, sortable archive using customized RSS channels based on ship type, homeport, and their area of expertise.

Advanced RSS

"Trackback" is another useful concept related to RSS. Any weblog posting that references a previous posting can issue a trackback to the server hosting the previous posting. Upon receiving the trackback, the server automatically (if desired) puts a link and a short summary of the new message underneath the old. If applied to equipment casualty reports, each time the original report is referenced (using the "REF" line in a traditional message), viewers of the original report would now see which other messages exist addressing the report. This could potentially eliminate much of the current flood of messages since anyone interested in the equipment casualty

could easily track the entire response effort from that single page.

There may also be applications in the multi-level security area. It has proven surprisingly difficult to provide comprehensive timely access to unclassified material to the Secret-level networks on the SIPRNET which dominate desktops at regional staffs and fleet commanders. The web is almost entirely unsuited to multi-level provision due to the numerous elements in almost every page and the breaking of pages upon transfer. Movement of information from the SIPRNET to the unclassified side is nearly impossible. RSS, on the other hand, neatly bundles the graphics and text of a "page" into a single element of human-readable data (rather than the security problems of proprietary formats such as .PDF or Microsoft Office files). This bundle, even transferred using CD-ROM or other air gap method, is immediately useful on the other side once loaded on an RSS host. The metadata could also be extended to include classification and declassification information to provide an auditable trail of the security history of a piece of information.

To enable the military to use the full capability of RSS, there are several extensions that should be made to the standard prior to adoption.

PKI: Authors should be able to authenticate their items by signing them using PKI. Signatures using a unit key may be a possible requirement for dissemination or entry into a message archive. Also, encryption using PKI may be adequate to allow storage of messages requiring special handling. The portion of the item signed (which may range from just the <description> element to the entire message other than the signature) should be clearly marked to allow the addition of additional metadata and signatures at a later date.

Message Priority, Security Classification: Minor additions to the elements of an item to match current practice.

Geographic Location: Another minor addition though the data format would have to be very strictly defined both in format and in levels of precision. Example:

<geolocation>56.04.XX.XN42.21.XX.XW</geolocation>. Xs are used instead of 0s to indicate that the information is unknown. Server-based aggregators could usefully provide a capability to pull messages from within a certain number of miles from a given point. Geographic location could also be valuable to an automatic security classification engine as a key part of the ruleset. Geographic location should refer to where an event occurred or is occurring, not the location of the message writer.

Operation Name, Operational Commander, Administrative Commander, Country: This information is expected to be useful when creating server-based channel filters. It allows easy partitioning by types of unit without having to directly list the units (which is a frequently shifting and difficult to maintain list). It also is valuable for message retention for historical purposes.

Conclusion

RSS and RSS aggregators together form an emerging technology whose adoption could make a significant impact on the military's concerns about knowledge management and bandwidth to isolated units. The military should immediately leverage this technology by developing an RSS proxy while simultaneously supporting experiments to distribute military information using RSS.

RSS is a clear example of an open standard with considerable commercial support which could improve information flow to warfighters.