

Semantic Attention Management

Charles Petrie • *Stanford University*



I stipulated last time that semantics, as a technology, will stand or fall based on its software engineering capabilities. This begs the question of a “killer app.” Certainly spreadsheet functions could have come about without Visicalc – we work in a Turing-complete space, after all – but they didn’t. Lesson learned: look for the application that people don’t yet know they want.

Many possibilities exist for semantic technologies. Indeed, using overloaded terms correctly in a given context is a major challenge that semantics is attempting to meet, thus making it the technology of globalization. But in this column, let me place some bets on a set of less obvious applications that will help us manage what we pay attention to: semantic attention management (SAM).

How do we focus on what we care about, excluding the barrage of others trying to get our attention? How do we reach others who are similarly excluding requests for attention? Part of the current answer is in what Chris Anderson termed “the long tail.”¹ You create a niche that only a few people care about, but you exactly target your audience – those who care about it a lot. (It all started with Star Trek.) The other part of the answer is in reaching that audience. The Internet makes it easy to offer content at a small price so that interested people can find it by searching and browsing. Collaborative filtering and social networking add a powerful feedback loop.

Semantic technologies (here, I focus on those based on the Resource

Description Framework [RDF; www.w3.org/rdf/]) offer to make our niches of interest even narrower, more dynamic, and easier to manage, including giving us more fine-grained control over who gets our attention with their Internet content – whether with products for sale (music, films, 1973 blenders, and so on) or for mind-share (blogs and email).

What’s Wrong with Mailing Lists

Let’s start to examine SAM by looking at technologies that might replace mailing lists. Today, people use list-servers such as Majordomo (www.myfirstserver.com/major_domo.htm) to subscribe to email lists of interest. The good news about mailing lists is that they’re likely to survive because they are a lowest common denominator: quick and easy to use, with a minimal technical commitment. Semantics isn’t yet popular and easy to use, but neither was email at one time. And mailing lists have issues that provide traction for semantics.

First, they require maintenance, as any list owner or subscriber knows. People’s interests and email addresses change, and the process of changing all the lists to which they subscribe is onerous, particularly if the old email addresses no longer work. That requires help from the list owners, who have to do manual pruning every so often anyway, in addition to simply being responsible to some institution for the lists.

Second, mailing lists aren’t easy to discover. I just heard that there’s a list

for our department to alert people (students) about left-over food, for example. Although I’ve been there for years, how was I to know? I suppose that if I’d thought such a list existed, I could’ve eventually uncovered it, but that search would take time I don’t have.

Third, existing lists might not exactly match our needs. I subscribe to one list that reaches only the owners of BMW R80ST motorcycles; only about 5,000 manufactured, and only in 1983 and 1984. That’s a fairly specific niche interest, but not specific enough. Surprisingly, the list generates several posts each day, although few interest me. Right now, I know everything I need to about this bike, except how to find good aftermarket parts, but I have to tune in to all of the other junk (interests other than mine) to catch the occasional hint. (The answer is an obscure company operating out of a garage in a small Bavarian village.)

Finally, mailing lists can fail to meet our needs at all. Suppose I wanted to send a message to all of the people in the travel department, for instance, to ask if anyone had seen a particular expense report, although I’m not in accounting and don’t know who is. Developing such lists would be difficult and inappropriate for a one-time use.

Worse yet, suppose that company policy calls for notifying the responsible administrator whenever reserving a “large” (defined in semantics or a list) conference room in any company building. Rather than having to research who the appropriate administrator is to reserve conference room

104 in building G1, I should be able to send an email to that *position*. Better yet, I should be able to simply reserve any room in any building and have the system automatically notify the proper parties. This scenario is a case for which mailing lists simply aren't appropriate.

Subscribing to mailing lists is a "macro" way to manage our interests and attention. Semantics offers a better way to solve all four problems.

Semantic Email Addressing

Imagine that we've all recorded descriptions of ourselves on not just the Web, but the Semantic Web (www.w3.org/2001/sw/). Many possibilities exist, but the simplest is Friend-of-a-Friend (FOAF; www.foaf-project.org). If I wanted to reach all BMW R80ST owners, my semantic email addressing (SEA) client would ask not for email addresses but rather for this description in some structured format. The SEA client would create a virtual list, dynamically, by searching through the public Semantic Web for people that matched our description. I could find and contact any BMW R80ST owners with FOAF files, even focusing on those who list brake discs among their interests. A SEA client lets you manage your interests and email addresses both through your public FOAF description; you don't have to join, much less manage or even discover, mailing lists.

What's more, we don't have to depend just on FOAF (and other RDF) descriptions; we can also use semantics to access and merge information from various sources, such as course-enrollment databases and public directories.

In the building-admin scenario, a simple SEA client would allow me to send email to someone who fit the semantics of `Administrator of MyCompany building G1`. An advanced (semantics-based) enterprise-management system would even let me reserve rooms and include a rule that "large conference rooms require approval by

the administrator of the building in which the room is located." Such a system would use SEA to send the email automatically when the room was reserved, alerting me that the reservation was contingent on approval.

SEA offers senders the advantage of precisely targeting the recipients by their individual interests, rather than broadcasting to a set of email addresses that might or might not be appropriate, or even valid. SEA messages go out to exactly the people who say their interests match, and to the email addresses they say are valid.

Oh, but won't this just generate more spam for the poor recipients? After all, email's inexpensiveness is why we get it from people we don't know offering to solve problems we don't have. Although SEA might generate more messages (as did the increased ease of email), it actually gives recipients better spam control.

Digital DERI SEA

SEA doesn't exist yet as a distributed (Internet) de facto standard because there's still no standard way to specify interests. For instance, my own FOAF can state only that I'm interested in BMW "R" series motorcycles (<http://xml.mfd-consult.dk/foaf/explorer/?foaf=http%3A%2F%2Fwww-cdr.stanford.edu%2F%2Fepetrie%2Ffoaf.rdf>). An effort is currently afoot to develop such an "interest" vocabulary (www.rdfweb.org/topic/ExpertFinder), but we're not waiting for that.

The Stanford Logic Group is part of an international research federation called the Digital Enterprise Research Institute (DERI; www.deri.org). To reserve rooms and describe internal events, resources, and people, DERI sites at Galway, Ireland; Innsbruck, Austria; and Stanford, California, are already using the centralized Digital DERI (DD) system, which is built on top of the Infomaster technology.² DD scoops up individual FOAF files and presents them all on a Web-based system that anyone with a DERI login can

modify, wiki-style. The system outputs RDF as needed to the DERI sites for internal usage, such as updating Typo3 database files (www.typo3.com) for generating Web pages.

People at the DERI sites can access and update their interests through the DD SEA, which is centralized but Web-based. The DD SEA client lets people search on sets of characteristics, including interests. If I wanted to reach just Reasonable Semantic Web Services (RSWS) research group members living in Austria, for instance, I could do so without knowing the email list name for that group (if any). The DD SEA client looks at the group tags and physical addresses of DERI members and returns the set of people to be sent email. When I send a message, the system will use my login information to fill in the `from` field and will hide the recipients' addresses in the `bcc` field. In addition, DD SEA then includes a non-standard field, called `Recipients-KIF`, which it populates with the set of conditions that selected the recipients, using the Knowledge Interchange Format (KIF; <http://logic.stanford.edu/kif/dpans.html>).

Users can receive DD SEA messages on any email client, but they must have a DD SEA client to perform a `reply-all`, which they can then do using the same set of conditions or just a subset of them. DD SEA will maintain a searchable log of sent emails, which anyone can use as long as they match the characteristics of someone who would have received such messages. After locating a given message, they can just use DD SEA to send a reply.

Of course, different individuals might now match those characteristics, and the new email might go to a somewhat new set of people – exactly what should and does happen in current mailing lists when sets of subscribers change between messages. (One existing mailing-list issue that SEA exacerbates is the new subscriber receiving a reply to a previously unseen message.)

Message specificity depends on the

specificity of the information that people declare about themselves. In an example of a positive feedback loop, SEA's value causes people to add information, which further increases value.

Beyond this simple example, we have the potential to do arbitrarily complex searches as people add information about themselves. For instance, we could send email to those living in Europe (including Galway and Innsbruck but not Stanford) and interested in Semantic Web services and soccer. Given that the emails are increasingly targeted to finer niches, and only by registered users, the emails are increasingly likely to be of real interest rather than spam that needs filtering.

But spam filtering is yet another possible SAM application. For instance, recipients could add SEA conditions to their filtering (or change their own semantic descriptions) and modify their filters with any arbitrary set of fine or general conditions they care to specify. For example, I could reject all email that comes from someone not in the white list and without a FOAF description that references someone within two degrees of separation from me with common interests.

Furthermore, it helps me solve my BMW motorcycle issue: I register a more general (still specific) interest such as "BMW R-series motorcycles," but I filter (delete or cache) all messages that aren't about aftermarket parts or, even more specifically, about disk brakes.

Semantic Email Workflow and Content

Semantic email can be incorporated into basic forms of workflow, such as scheduling meetings (including finding the right recipients), getting RSVPs, finding common dates, and general organization (www.cs.uga.edu/~rsp/project/proposal.htm). Some implementations of such semantic email workflow (SEW) systems are already available publicly (www.cs.washington.edu/research/semweb/email.html).

Though meeting scheduling seems like a killer application on this surface, its success isn't entirely clear. For one thing, combining structured knowledge with email to produce dynamic flexible workflow isn't new: such efforts in the '80s created Lotus Notes.³ But this never yielded an open email standard. Another issue is that online calendaring systems have historically struggled to be successful, current Web-based invitation systems (such as www.sendomatic.com or www.evite.com) could continue to be more popular than SEW systems.

Nonetheless, SEW has clear potential. In addition to research (and personal) interests, for instance, we could

phenomena tending to focus on narrow niche interests, they're natural applications of semantics. Similar to semantic email, we can have approaches that concentrate on content (like SEC) in blogs (<http://heory.csail.mit.edu/~dquan/iswc2004-blog.pdf>) and wikis (www.www2006.org/programme/files/xhtml/p171/pp171-oren/pp171-oren-xhtml.html and http://en.wikipedia.org/wiki/Semantic_Wiki). Many implementations of semantic wikis appear to exist already (http://wiki.ontoworld.org/index.php/Semantic_Wiki_State_of_the_Art).

Such efforts are also helping to solve the problem of generating standard semantic specifications. Right

Email lists are no longer needed; we can do better because we can reach exactly the people we intend to reach — and we no longer have to manage list subscriptions.

point to travel plans (with RDF) and other such activities in FOAF descriptions. We could then send a message to everyone interested in FOAF technologies who will be in Georgia in November 2006. My own travel plans are always on the Web, though only in HTML. (It's surprising that no one has yet proposed an RDF-based standard description for travel.)

Another possibility is to embed RDF directly in the email text. Semantic email content (SEC) would allow even more sophisticated processing, but it's unlikely to catch on because it would require special email clients. However, all of these possibilities — SEA/SEW, SEA/SEC — are compatible components of semantic email, which has potential to be a killer app for the Semantic Web.

Semantic Blogs and Wikis

Because blogs and wikis are long-tail

now, these hover between formal ontologies and informal tagging, but these efforts might converge into something like "folktologies" (http://novaspivack.typepad.com/nova_spivacks_weblog/2005/01/whats_after_fol.html) with the momentum of semantic wikis and blogs. Collaborative filtering, similar to Amazon recommendations, is another forcing factor in this space (www.rereviewed.com/roguesemiotics/index.php?p=137).

An alternate approach is to concentrate on the metadata (like SEA). Despite the momentum in content-based semantic blogs, more people will incorporate metadata semantics as doing so requires less work. RSS is already metadata in RDF, and people can already make precise queries over RSS indexes with current tools (www.xml.com/lpt/a/ws/2003/04/15/semanticblog.html). Of course, a big feature of RSS is that it "pushes" the

information to the consumer. However, the potential for semantics isn't just to find finer-grained RSS feeds. Rather, like SEA, it could let us do away with persistent subscriptions to blogs (like persistent mailing lists), instead letting us ask to be notified whenever any blog on the aggregator made a new post that fit our semantic criteria. For example, I want to know when anyone posts, anywhere, a reference to political news about anyone who served in a government post under a US administration that also employed David Addington in any capacity.

This suggests at least one interesting issue for SEA. We can, in DD, generate notifications based on semantic descriptions of individuals. For instance, I'd like to be notified whenever a visitor from another DERI site posts an event of "arrival at Stanford." When we subscribe to an email list, we're saying we want to be notified of any event (email) on that list. How can we do that with SEA? I've argued that such lists are too coarse-grained – that we want to receive lists only on more niche topics – but that could also make us miss some posts.

SEA might thus need an additional feature: a subscription function through which we could specify conditions for wanted emails. As with RSS, this would require email aggregators – in this instance, the DD SEA inbox collection. The system could automatically send the subscriber any email that fit the subscription conditions, regardless of whether it was intended for a person of the subscriber's description. Blogs and email would merge into similar functionality, especially as email becomes more Web and attachment-oriented.

Privacy argues against such an email subscription system. Right now, we allow those logged in to DD to see only emails that meet their semantic descriptions, rather than their search interests. It's easy to imagine the abus-

es that could arise from enabling anyone to tap into the email stream by selecting interests. Nonetheless, semantics makes such an email subscription function possible, not to mention many other functions that we can't even imagine right now. Semantics promises to make our future life on the Net even more interesting than it is. ☐

References

1. C. Anderson, "The Long Tail," *Wired*, Oct. 2004; www.wired.com/wired/archive/12.10/tail.html.
2. M.R. Genesereth, A.M. Keller, and O.M. Duschka, "Infomaster: An Information Integration System," *Proc. ACM SIGMOD Conf.*, ACM Press, 1997; <http://www-db.stanford.edu/pub/keller/1997/infomaster-sigmod97-demo.pdf>.
3. K.Y. Lai, T.W. Malone, and K.C. Yu, "Object Lens: A Spreadsheet for Cooperative Work," *ACM Trans. Office Information Systems*, vol. 6, no. 4, 1988, pp. 332–353.

Charles Petrie is a senior research scientist and consulting associate professor at Stanford University. His research interests include concurrent engineering, virtual enterprise management, and collective work. Petrie has a PhD in computer science from the University of Texas at Austin. He is EIC emeritus and a member of *IC*'s editorial board. Contact him at petrie@stanford.edu.