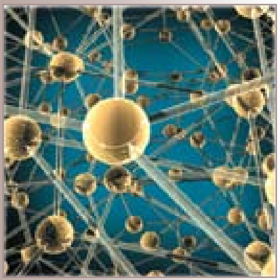

Semantic Email Addressing
The Semantic Web Killer App?

Email addresses, like telephone numbers, are opaque identifiers. They're often hard to remember, and, worse still, they change from time to time. Semantic email addressing (SEA) lets users send email to a semantically specified group of recipients. It provides all of the functionality of static email mailing lists, but because users can maintain their own profiles, they don't need to subscribe, unsubscribe, or change email addresses. Because of its targeted nature, SEA could help combat unintentional spam and preserve the privacy of email addresses and even individual identities.

Michael Kassoff, Charles Petrie, Lee-Ming Zen, and Michael Genesereth
Stanford University

Email addresses are a means to an end. The goal is usually not to send an email to a particular address, but to a particular person. You want to say hello to your friend Steve or send a message to the VP of marketing at Microsoft or to the head caterer for your wedding. Ideally, you could send a message to a person just by entering his or her name, position, or some other descriptive attribute. If a person's email address changes, the email system should send to the new address automatically. If the person matching a description differs over time, the email system should send to the person currently matching that description.

More generally, you should be able to send email messages to groups of people matching a particular set of attributes: all Stanford department chairs, all female customers living in Detroit, or all people in your organization who speak both English and French. Today, we use mailing lists to email predefined groups of people. However, because infinitely many ways to define a set of people exist (for example, "all people in the marketing department whose name starts with the letter 'M'"), you generally can't rely on such lists. Instead, you must be able to address your email to static mailing lists that are the best fit to your requirements, and you must know of their existence.

Semantic email addressing (SEA) is a simple but novel technology that lets you address email to a semanti-

 1089-7801/09/$25.00

Related Work in Semantic Email Addressing

Several other researchers have recognized the value in bringing semantics to email. For example, the Information Lens system[1] lets users send semistructured email messages and filter those messages using production rules. Users can send to a special mailbox called "anyone," and anyone can choose to receive messages from this mailbox based on production rules. This flips the nature of widely broadcast emails on its head. Instead of starting with receiving all emails and whittling them down based on filtering rules, the user starts with an empty inbox and pulls in email of interest. This is similar to the RSS subscription model. As RSS feeds contain more semantic information, the semantic subscription model exemplified by Information Lens might become more commonplace.

More recently, MailsMore[2] lets users annotate an email's content with Resource Description Framework (RDF) triples and automatically includes RDF triples based on standard email headers such as the "To," "From," "Subject," and body fields. This can be used for semantic filtering and filing of emails.

The Mangrove system[3–5] takes this idea further. It allows not only structured email content but also semantic email processes. Users can script email clients with declarative workflows that automatically aggregate information obtained from many email responses, automatically resend emails to people who haven't responded, or analyze the semantic content of incoming email messages and respond accordingly.

Most relevantly, Microsoft Exchange 2003 lets administrators create query-based distribution groups, which are essentially mailing lists whose recipients are based on a Lightweight Directory Access Protocol (LDAP) query run when the email is sent. This alleviates much of the administrative work required to maintain a mailing list. However, because only an administrator can create the mailing lists, users can't send SEA mail, and the information upon which the lists are based isn't under users' control. None of this application's functionality is available to users and very little to administrators. In fact, users can't see that a distribution group is query based: each query-based distribution group has a name, so to an outsider looks like a regular mailing list.

References

1. T. Malone et al., "Intelligent Information-Sharing Systems," *Comm. ACM*, vol. 30, no. 5, 1987, pp. 390–402.
2. A. Kalyanpur et al., *Representation Formalisms and Methods,* tech. report, Univ. of Maryland, 2001; www.mindswap.org/papers/SMORE.pdf.
3. O. Etzioni et al., "Semantic Email: Adding Lightweight Data Manipulation Capabilities to the Email Habitat," *Proc. 6th Int'l Workshop the Web and Databases*, ACM Press, 2003, pp. 12–13.
4. L. Mcdowell, O. Etzioni, and A. Halevy, "Semantic Email: Theory and Applications," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 2, no. 2, 2004, pp. 153–183.
5. L. McDowell et al., "Semantic Email," *Proc. 13th Int'l Conf. World Wide Web* (WWW 04), ACM Press, 2004, pp. 244–254.

cally defined set of entities (see the "Related Work in Semantic Email Addressing" sidebar for other research in this area). A SEA mail server computes the recipients of a semantically addressed email on the fly based on the address's semantic definition. SEA has other benefits as well. First, it doesn't require discovery. In traditional email systems, it's difficult for humans to discover email addresses and mailing lists, and automatic discovery by computers is almost impossible. In addition, SEA requires no maintenance. In traditional mailing lists, an administrator must create and maintain the list and individuals subscribe or unsubscribe to the list, and change their information with regard to the list. This can be particularly onerous when users must deal with many lists — for example, when their email addresses change. Instead, users can update their personal information, such as email address, once, in a single place, and the SEA mail server would automatically adapt.

SEA has application in both corporate intranets and the Internet. It also raises several issues, including security and privacy issues, errors, user adoption, and standardization.

Examples

To illustrate SEA, we consider two examples.

Corporate Example

Corporations and other organizations often have databases of information about personnel, projects, customers, and so on. Users can leverage this information to send emails based on properties of the people in a particular department.

Consider a moderate-sized company with several departments. The company has a centralized database containing the following information about its personnel: name, email address, department, group, position, project, and date of hire. Given this information, a user might define groups of people by querying the data — for example, for "all senior managers in the accounting department."

The company might also have the following information about its projects: project name, priority, leader, start date, and end date. With this

```
<foaf:name>Charles Petrie</foaf:name>
<foaf:workplaceHomepage rdf:resource="http://www.stanford.edu/"/>
<foaf:interest rdf:resource="http://en.wikipedia.org/wiki/BMW_motorcycles#R_series"/>
<foaf:interest rdf:resource="http://en.wikipedia.org/wiki/German_language"/>
<foaf:interest rdf:resource="http://en.wikipedia.org/wiki/Semantic_Research"/>
<foaf:mbox rdf:resource="mailto:petrie@stanford.edu"/>
  <foaf:knows>
    <foaf:Person>
    <foaf:name>Axel Polleres</foaf:name>
    <rdfs:seeAlso rdf:resource="http://www.polleres.net/foaf.rdf"/>
  </foaf:Person>
</foaf:knows>
```

Figure 1. Fragment of Charles Petrie's Friend-of-a-Friend (FOAF) profile. With semantic email addressing (SEA), you could use information from this and other FOAF profiles to send an email to a specific group of individuals, such as Stanford University employees interested in semantic research.

information, a user can define precise groups of people such as "all developers for current projects in the database group." Using SEA, the user can send an email to this group. Like a traditional mailing list, a recipient of a semantically addressed email can respond to the sender or to the group. Unlike a traditional mailing list, the recipient can respond to a particular subset of the group or forward the email to some other semantically defined group of people.

Using SEA, a computer could send an email to a specific person or group of persons. For example, you could program the company room-reservation system to automatically send an email to the building manager each time conference room 101 is reserved. As the employee serving as building manager changed over time, the room-reservation system would always send the email to the correct person, without your having to reprogram it.

Internet Example

Recently, the Friend-of-a-Friend (FOAF) Resource Description Framework (RDF) ontology has gained popularity. FOAF predicates a person's express properties, such as name, email address, group memberships, employer, gender, birthday, interests, projects, and acquaintances. As of 2004, more than 1.25 million FOAF documents were publicly available on the Internet, and that number has certainly grown since.[1]

By spidering the Semantic Web and collecting the information contained in FOAF files, you can build a large collection of data about people and their interests. You could use this information to email people with a given interest, who know people who know a particular person, and so on. Figure 1 shows a fragment of one of our FOAF files. As you can see, Charles Petrie is a Stanford employee interested in BMW R-series motorcycles, the German language, and semantic research, and he knows Axel Polleres. Using this information and similar information culled from various other FOAF files, you could address an email to all Stanford employees interested in semantic research, all people interested in R-series motorcycles, and so on.

Petrie's FOAF file is a single place in which he can control his personal information. If he were to change his email address, all semantically addressed email based on his FOAF profile would be automatically routed to his new address. If he left Stanford, he'd no longer receive emails addressed to Stanford people interested in semantic research. All of this is under his control, and he doesn't need to change his settings with each mailing list individually. The changes happen automatically.

In a sense, SEA is the opposite of spam. Although both SEA and spam might involve unsolicited emails, spam goes to everyone, whereas SEA is targeted toward those who've publicly announced their interest in the SEA mail topic. SEA is a marketer's dream come true. Moreover, unlike mailing lists, it requires no discovery for either the sender or the receiver.

Of course, we don't have to depend on FOAF descriptions alone. We can also integrate semantic information from various sources, such as RDF files or relational databases. For example, we could pull bibliographic information from a site such as DBLP and email everyone who has ever coauthored a paper with Petrie and whose email address is publicly available.

Using Semantic Email Addressing

Now that we've motivated SEA's myriad benefits, let's look at some practical details. How should it work? That is, how do you send semantically addressed email, and how would you reply to such an email?

To semantically specify a set of email addresses to which to send a message, a user needs three somewhat separable pieces of functionality:

- The user must be able to define the group of email addresses of interest. This might involve choosing from some predefined list of definitions; to attain SEA's full power, however, the user should be able to formulate new definitions.
- The definition must somehow be translated to a set of email addresses to which the email is sent.
- The SEA mail server must facilitate replying in a simple manner. This final piece is optional but sometimes useful.

An email client that allows SEA requires only some small extensions over a traditional email client. This might involve adding an extra button that lets a user specify a semantic email address. This functionality is similar to the address book functionality in modern email clients. When the user presses this button, a window pops up with an interface for defining a set of recipients, as in Figure 2.

Because a definition must be formulated in a particular ontology, the user must be able to specify a query in the ontology directly — say, via a textual editor — or use a tool that facilitates query formulation in that ontology. Generally, only power users will be able to formulate queries directly, so a graphical interface for query creation is useful here. The tool should be generic and able to generate query interfaces on the fly based on the ontology and some display metadata.

Figure 2 shows the interface to our prototype. The system automatically generates the interface based on the schema for a person and some metadata about how to display each field — for example, as a text box or a dropdown list. The interface lets the email sender define a set of people as opposed to a set of email addresses. Because each person is associated with at most

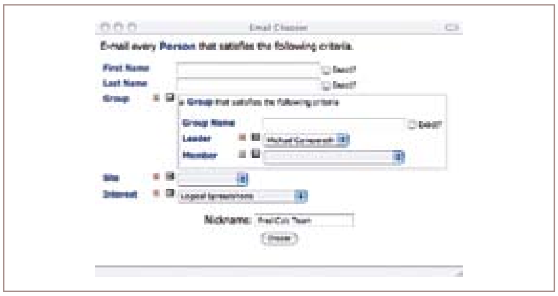

Figure 2. Interface for semantic email addressing. The interface lets users enter a specific name, group, location, or interest. Here, the user wants to send an email to members of the group led by Michael Genesereth interested in logical spreadsheets.

Figure 3. Confirmation page. After the user presses "Send" using the selection in Figure 2, the interface displays the list of people who will receive the email.

one email address, this also unambiguously defines the set of email addresses to send the message to.

The prototype interface lets users form embedded queries of arbitrary depth. For example, a user might form a query for all people at a site that's in a country where French is spoken. This cross-category search is particularly useful when a large amount of supporting information is available. In particular, if the Semantic Web becomes a reality, users could base their queries on arbitrary information on the Web.

Users could easily employ SEA to email large numbers of people. We therefore need safeguards to ensure that user errors won't result in a mass spamming. Feedback to the user regarding his or her query definition could take the form of a list of people to which the email will be sent, assuming the number of people is sufficiently small that such a display is feasible. Otherwise, displaying the number of people to which the email will be sent along with a sample of those people could be useful. Such feedback would let the user catch errors in the query definition and give confidence that the message won't accidentally go to a large number of people for whom it's not meant. Figure 3 shows a simple confirmation page displayed after the user presses "Send" for the set of people defined in Figure 2.

sea+.28.3F5.20AND.20.28PERSON.2EGROUP.20.3F5.20.3F6.29.20.28GROUP.2ELEADER.20.3F6.20MICHAEL.2EGENESERETH.29.20.28GROUP.2EINSTANCE.20.3F6.29.20.28PERSON.2EINTEREST.20.3F5.20PERSONALINTEREST.2E3364062876.29.20.28PERSON.2EINSTANCE.20.3F5.29.29@logic.stanford.edu

Figure 4. The semantic email address from the example in Figure 2. Ideally, these ungainly addresses will be hidden from the user.

In our prototype, semantic email addresses take the form query@domain. For example, if you send an email to a properly encoded "people in the logic group interested in logical spreadsheets"@logic.stanford.edu, the SEA system will send an email to the appropriate people (replacing the phrase in quotations with an appropriate computer-interpretable string). This format is useful for sending mail from email clients that don't support SEA, for adding semantic email addresses to your address book, and so on. However, in an ideal world — that is, a world in which all email clients support SEA — these semantic email addresses would be completely hidden from the user because they're long, unintelligible, and ugly, as the example in Figure 4 illustrates.

Instead of viewing such ungainly identifiers, we expect users to always view semantic email addresses through some graphical representation, such as the one in Figure 2, or a human-readable textual description, such as one written in a controlled subset of English. In our prototype, we also allow a short, human-readable nickname to be associated with a semantic email address using the same mechanism that associates usernames with normal email addresses. In Figure 2, the nickname "PrediCalc Team" represents the semantic email address in Figure 4 — that is, "people in the logic group interested in logical spreadsheets"@logic.stanford.edu.

Upon receiving a semantically addressed email, the receiver should be able to view both the semantic definition of the email's addressees and the actual people to whom the email was sent. The receiver can view the semantic email address through an appropriate GUI or natural language display (if the email client supports semantic email addressing) or by clicking a Web page link, which the SEA mail server sending agent automatically generates and places in the message's footer.

Viewing a list of the actual recipients is a bit trickier because the set of email addresses defined by a particular query can change over time. If the number of email addresses is small, the actual email addresses to which the email was sent might appear in the email's "To" or "Cc" fields. If the number is large, however, this becomes impractical. The sending agent must somehow keep track of either the set of addresses for each message or enough of the database's state at each point in time to reconstruct this information. Doing so in a space-efficient manner is an open problem.

Replying to a semantically addressed email raises similar issues. If the recipients' addresses are sent in the email's "To" or "Cc" fields, you can reply to the original message's recipients in the usual manner, or you can use SEA to reply. If the recipient list isn't included in the message, you must use SEA to reply. However, the set of people satisfying a condition can change over time, so replying can result in new people receiving the reply and old people not receiving it. Of course, this is nothing new; a traditional mailing list's subscribers also change over time.

Prototype SEA Module

Our prototype SEA module — the Infomaster Semantic Email Addresser (ISEA) — runs on top of the Infomaster information integration engine.[2] Infomaster lets you query multiple data sources on the Internet through a single mediated schema. The system can therefore pull information from many sources — not just about people but also useful supporting information about organizations, locations, and so forth.

Researchers at three Semantic Technologies Institutes — Digital Enterprise Research Institute (DERI) Galway, STI Innsbruck, and the Stanford Logic Group — tested ISEA over a period of one year. The requirements for an industrial-strength version were jointly developed and development is planned. Because STI is large and distributed, with members frequently coming and going, it's difficult for members to keep track of other members' locations and activities. It's therefore a natural application for SEA. The prototype lets members email people based on their site, group affiliations, name, interests, and other attributes (see Figure 2). We obtain this information from private databases and publicly available FOAF files.

An administrator sets the ontology SEA uses, so it's centrally controlled but is continually evolving.

Our prototype has four principal components:

- a Web-based user interface for constructing semantic email addresses;
- a database (Infomaster) for determining queries to determine the traditional email addresses of people matching a query;
- an email system (Postfix) for sending emails to (traditional) email addresses and receiving semantically addressed emails; and
- application code (ISEA) to glue the first three pieces together.

The system typically interacts with these components in the following order.

First, the user constructs a semantic email address using our semantic-address-creating interface. Optionally, the user can query the database through the interface to determine the list of recipients identified by the semantic email address he or she has constructed. The user then sends an email to this address using any email client.

Next, the ISEA Postfix application receives the email. This triggers an action that calls the ISEA application code, which translates the semantic email address into a database query. The database query results in a list of (traditional) email addresses. ISEA uses Postfix to forward the email to this list of email addresses, placing the addresses in the email's "Bcc" header and the semantic email address in the "To" header. (So, ISEA will receive another copy of the email, which can lead to an infinite loop. To avoid such a loop, we add an email header called X-ISEA, which we use to flag whether the message has been seen before and therefore shouldn't be processed.)

Finally, recipients receive the message using their email client of choice.

When replying to an email with a semantic email address, the user skips the first step (constructing a semantic email address).

Open Issues

SEA raises some important issues, both for its use in a corporate environment and on the open Internet.

Security and Privacy

People outside of an enterprise can use SEA to send email to members of the enterprise based on members' semantic descriptions. For example, you might want to send a message to the lead developer of a particular product at Yahoo, but you don't know the individual's email address, and Yahoo might not want you to know these details. However, the company might still want to let you contact the developer. By making a SEA form available, they can allow this in a simple way.

SEA is ideal for targeted email addressing, but it's also ripe for abuse. An individual or company could easily use SEA to send exceedingly untargeted emails. Within small- to medium-sized communities, this sort of behavior is often uncommon because violating community rules leads to social repercussions. Within larger communities, administrators might need to enact security policies to limit who can send to whom. On the public Internet, however, SEA abuse is much harder to control. Of course, this is true for any publicly available email address, whether or not SEA is used (consider, for example, the problem of webmaster@site.com addresses).

On the Internet, the more interesting issues are data ownership and authentication. Preventing users from changing their profiles to spoof membership in a restricted group on the Internet will require new solutions. Users requiring security will want to use a SEA system that authenticates profile properties against at least local permissions and similarly authenticates incoming messages.

In addition, on the Internet, SEA with subscription to public invisible properties becomes indistinguishable from semantic RSS, except for the authentication issue.

Dealing with Errors

One of SEA's most powerful aspects is that the set of recipients changes over time as the people and properties in the database change. Ideally, the database is always up to date, reflecting the world's changing realities. In practice, information in the database isn't always 100 percent accurate.

Inaccurate information can lead to people receiving email when they shouldn't and not receiving email when they should. System users should be able to ensure that their information is correct, either by editing it themselves or by having an authorized administrator change the information for them.

In our prototype system, users have password-protected access to their profiles via a Web interface. Individuals therefore control all of their own information and are expected to keep their profiles up to date. As you might expect in a large organization, not all information is always accurate. However, for frequently used semantic emailing addresses (for example, "all people interested in SEA"), the relevant information is more often kept up to date than addresses that haven't yet been used for SEA.

When an email bounces or is delayed, the system forwards an error message to the sender. However, in some cases the semantic email address is meant to anonymize the message recipients. In these cases, instead of forwarding the error message to the sender, the system sends a message that masks the recipients' identities. For example, the message might read, "Your email message has not been sent to the intended recipient. We apologize for the inconvenience." These problems and solutions are no different from those for traditional mailing lists.

User Adoption

One of the properties that could be selected in the prototype was "site," which had three values: Galway, Innsbruck, and Stanford. The number of people varied but was under 500. In total, the sites have 15 research groups. Each group has a leader and a set of members. Each person can change anything in the set of profiles, much as with a wiki. Anyone can add new groups and people. Each person has a profile of personal data, including their interests. Anyone can add a new interest. This openness is by design after consideration by a working design council. For the production version, we must modify the openness in accordance with the security and privacy considerations discussed earlier.

Even among technical people, we must carefully manage SEA's adoption. People often remark that they trust their existing email lists but not SEA virtual lists. When asked specifically about this trust (or lack thereof), they say that they know who is on the existing lists. At least at STI Innsbruck, however, this claim proves incorrect: only the administrator can see who is on the list. In contrast, any sender can easily check the intended recipients of a SEA message.

Similarly, people have remarked that they won't be able to use SEA messages like they use email from existing email lists. When asked for specifics, they say they store messages from specific email lists in specific folders and later use these emails to send to the email lists. Of course, they can use SEA virtual email lists in the same way. In fact, the ISEA nickname facilitates this methodology.

In general, SEA provides the same information and functionality as existing email lists, and more. Whatever worked with static email lists works with SEA virtual email lists, and with less effort. Thus it appears that there are initial psychological, rather than technical, barriers to SEA adoption, and we've seen that this conservatism dissipates with use.

Standardization

SEA doesn't require that all implementations use the same addressing scheme. The scheme can vary from organization to organization or even from server to server within an organization. However, we expect that one or more standards will emerge over time. With standardization, email clients could provide better built-in support for SEA, including SEA address editing, analysis, and search.

Future Work

In addition to work on security and privacy in SEA, we plan a wide range of improvements for our prototype. A parallel effort is seeking to develop SEA-specific email clients to solve the problem of the ugly semantic email address. However, we're directing our major development work toward a user interface to make SEA look more like existing popular email interfaces.

Letting external groups use SEA is much more problematic because we must address issues of distributed permissions and authentication. There's also the general semantic problem of linking to arbitrary external semantic sources on unifying the results. We solved much of this problem with Infomaster.[2] However, there remains the ontological engineering problem of constructing proper hierarchies of ontologies so that Petrie can use them to declare that he filters out email about BMW K-series motorcycles from all of those about BMW motorcycles. (Proper SEA design assumes that superclasses of expressed interest aren't of interest to the person profiled, but subclasses are.)

An interesting possibility to help control spam is semantic filtering and filing of email. A user could write semantic email rules based on not only standard email fields but also the semantic email address. For example, a user might classify as spam all email from someone not on a whitelist and without a FOAF that references someone within two degrees of separation who has no common interests. This semantic filtering and filing could be extremely effective, especially if combined with semantic email content. Other interesting problems are generating user interfaces that let users specify these sorts of filters and efficiently filtering large numbers of emails against a semantic email address.

SEA is a simple concept that has the potential to be easily implemented, leveraging existing semantics (such as FOAF) and significantly improving the functionality of email systems. We hope it will soon be incorporated into commercial email systems. Because it not only frees administrators from maintaining static lists but also gives people more control over their email based on local control of their semantic data, SEA could be the "killer app'' of the Semantic Web.

References

1. L. Ding et al., "How the Semantic Web Is Being Used: An Analysis of FOAF Documents," *Proc. 38th Ann. Hawaii Int'l Conf. System Sciences* (HICSS), IEEE CS Press, 2005, pp. 113–123.
2. M. Genesereth, A. Keller, and O. Duschka, "Infomaster: An Information Integration System," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, ACM Press, 1997, pp. 539–542.

Michael Kassoff is a PhD candidate in the Stanford Logic Group. His research interests include logical spreadsheets and semantic data capture methods on the World Wide Web. Kassoff has an MS in computer science from Stanford University. Contact him at mkassoff@stanford.edu.

Charles Petrie is a senior research scientist in the Stanford Logic Group. His research interests include concurrent engineering, enterprise management, and collective work. Petrie has a PhD in computer science from the University of Texas at Austin. He was the founding editor in chief of *IEEE Internet Computing*. Contact him at petrie@stanford.edu; www-cdr.stanford.edu/%7Epetrie/.

Lee-Ming Zen is an applied researcher at Microsoft adCenter Labs. His research interests include online user behavior, text mining, and advertising. He has an MS in computer science from Stanford University. Contact him at lzen@cs.stanford.edu.

Michael Genesereth is an associate professor in the computer science department at Stanford University, director of the Stanford Logic Group, and research director of CodeX (the Stanford Center for Computers and Law). His research interests include computational logic and its application in enterprise management and electronic commerce. Genesereth has a PhD in applied mathematics from Harvard University. Contact him at genesereth@stanford.edu.