Semantic Email Addressing: Sending Email to People, Not Strings

Michael Kassoff, Charles Petrie, Lee-Ming Zen and **Michael Genesereth**
Logic Group, Department of Computer Science
Stanford University
{mkassoff, petrie, leezen, genesereth}@stanford.edu

Abstract

In this document, we explore how semantic technologies can be brought to email addressing. We introduce the notion of *semantic email addressing (SEA)*. SEA allows emails to be sent to a semantically specified recipient of group of recipients, which may be dynamically changing over time. We give some applications of SEA and describe our prototype implementation.

Introduction

Email addresses, like telephone numbers, are opaque identifiers. They are often hard for a person to remember, and worse still, people's email addresses and phone numbers change from time to time.

Email addresses are a means to an end. One's goal is usually not to send an email to a particular address, but instead to a particular *person*. One wants to say hello to our friend Steve, or send a message to the VP of marketing at Microsoft, or to the head caterer for one's wedding. Ideally, one could send a message to a person just by entering his name, his position, or some other descriptive attributes. If the email address of a person changes, then the email system should send to the new email address, automatically. If the person matching a description differs over time, the email system should always send to the person currently matching that description.

More generally, one should be able to send emails to groups of people matching a particular set of attributes: all chairs of departments at Stanford, or all female customers living in Detroit, or all people in our organization who speak both English and French.

Today, mailing lists are used to email predefined groups of people. However, as there are infinitely many ways to define a set of people (e.g."all people in the marketing department whose name starts with the letter 'M' "), one cannot in general rely on such predefined lists. Instead, one must have the ability to address our email arbitrary groups of people.

We use the term *semantic email addressing* (SEA) to refer to emails that are addressed to a semantically defined set of entities. The recipients to a semantically addressed email are computed on the fly based on the semantic definition of the address.

SEA has other benefits as well:

No discovery required. Email addresses and mailing lists can be difficult to discover, even for a human. Worse, since email addresses and mailing list names have no semantics, automatic discovery of email addresses and mailing lists by a computer is an extremely difficult task, if not impossible. With SEA, discovery is completely obviated.

No maintenance required. Traditional mailing lists require manual labor to maintain. A mailing list administrator must go through the process of creating the list. The list must then be maintained by the administrator himself and the individuals who would like to subscribe to the list, unsubscribe to the list, and change their information with regards to the list. This can be particularly onerous when the user must deal with many lists, for example when his email address changes. Ideally, the user could update his personal information such as email address once, in a single place, and have email addressers automatically adapt. SEA makes this possible.

The rest of this paper is organized as follows. We first describe some applications of SEA. Next, we explain how one can send and reply to messages using SEA, and discuss our prototype implementation called ISEA. We then touch upon security and privacy issues raised by SEA. Finally, we describe related and future work.

Examples

To illustrate semantic email addressing, we consider two examples. The first illustrates SEA in a controlled, corporate environment. The second illustrates the use of SEA using public information gathered from the Internet.

Corporate Example

Corporations and other organizations often have databases of information about personnel, projects, customers, and so on. This information can be leveraged to send emails based on properties of the people in the department.

As an example, consider a moderate size company with several departments. The company has a centralized object-oriented database containing the following information about company personnel:

```
<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 FOAF profile

```
Name
Email Address
Department
Group
Position
Project
Date of Hire
```

Given this information, one may define groups of people by querying the data, for example for "all senior managers in the accounting department." If the company also has the following information about projects:

```
Project Name
Priority
Leader
Start Date
End Date
```

One may then define precise groups of people such as "all developers for current projects in the Database group." Using semantic email addressing, one may then send an email to this group. Like a traditional mailing list, a recipient of a semantic email address may respond to the sender or to the group itself. Unlike a traditional mailing list, the recipient may respond to a particular subset of the group, or forward the email to some other semantically defined group of people.

Using semantic email addressing, it would be easy for a computer to send an email to a specific person or group of persons. For example, the company room reservation system could straightforwardly be programmed to automatically send an email to the building manager each time conference room 101 was reserved, without knowing who the building manager is. As the employee serving as building manager changed over time, the email would always be sent to the correct person, without having to reprogram the room reservation system.

Internet Example

Recently, a RDF ontology known as FOAF, or Friend-of-a-Friend, has gained in popularity. This ontology contains predicates for expressing properties of a person such as their name, email address, group memberships, employer, gender, birthday, interests, projects and acquaintances. As of 2004, over 1.25 million FOAF documents were publicly available on the Internet, and that number has certainly grown since (Ding *et al.* 2005).

By spidering the Semantic Web and collecting the information contained in these FOAF files, one can build up a large collection of data about people and their interests. This information can then be used to email people with a given interest, or who know people who know a particular person, and so on. To illustrate, a fragment of one of the authors' FOAF file is shown in Figure 1. As you can see, Charles is a Stanford employee interested in BMW R series motorcycles, German, and semantic research, and knows Axel Polleres. Note that using this information and similar information culled from various other FOAF files, one could address an email to all Stanford employees interested in semantic research, or all people interested in R series motorcycles, and so on.

Note how his FOAF file is a single place in which Charles can control his personal information - if Charles were to change his email address, then all semantically addressed email based on his FOAF profile would be automatically routed to his new address. If Charles were to leave Stanford, then he would no longer receive emails addressed to Stanford people interested in semantic research. All of this is under Charles' control, without the need to change his settings with each mailing list individually. It all happens automatically.

In a sense, SEA is the opposite of spam. While both SEA and spam may involve unsolicited emails, spam is sent to everyone, while SEA is targeted towards those who have publicly announced they are interested in it. SEA is a marketer's dream come true.

Note also that, unlike mailing lists, there is no discovery required for either the sender or the receiver. It just works.

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

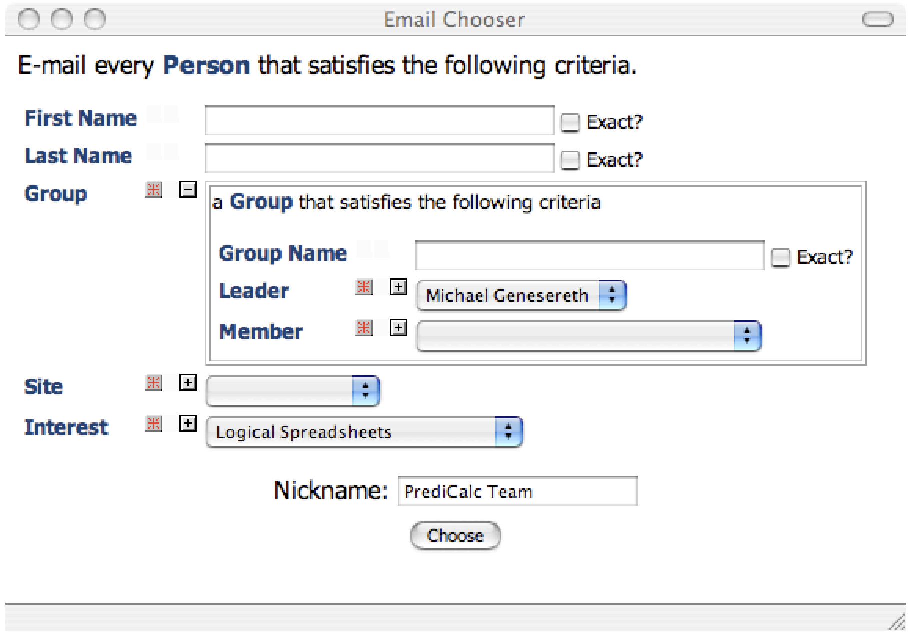

Figure 2: Sending email to members of the group lead by Michael Genesereth interested in logical spreadsheets

Prototype

We have built a prototype SEA module on top of Infomaster (Genesereth *et al.* 1997) called ISEA ("Infomaster Semantic Email Addresser"). Infomaster is an information integration engine, and can be used to query multiple sources of data on the Internet through a single mediated schema. This is useful for SEA because it allows information to be pulled in from many sources - not just about people, but also useful supporting information about organizations and locations and so forth. The prototype is currently being used in a system used by members of the Digital Enterprise Research Institute in four locations - Stanford, Galway, Innsbruck and Korea. Since the institute is large and distributed, and members are frequently coming and going, it is hard for a member of the institute to keep track of who is where and doing what. Thus, it is a natural application for SEA. The prototype allows people to be emailed based on their site, group affiliations, name, interests, and other attributes (see Figure 2). This information is obtained from private databases and publicly available FOAF files.

Using Semantic Email Addressing

Now that we have motivated the myriad benefits of SEA, we delve into some practical details. How should it work? We first talk about sending semantically addressed email, then we talk about how one might reply to a semantically addressed email.

To allow the user to semantically specify a set the email addresses to which to send to, there are three somewhat separable pieces of functionality required. First, the user must be able to define the group of email addresses of interest. This might involve choosing from some predefined list of definitions; though to attain the full power of semantic email addressing, the user should be able to formulate new definitions. The second piece of functionality is that the definition must somehow be translated to a set of email addresses to which the email must be sent. The final piece of functionality is to facilitate replying in a simple manner. This final piece is optional but sometimes useful.

An email client that allows for SEA requires only some small extensions over a traditional email client. In our prototype, the interface for composing a message looks just like a traditional interface, with To, Cc, Bcc, Subject and Body fields. The only difference is an extra button, which allows a semantic email address to be specified wherever a traditional email address can be placed. This functionality is similar to the address book functionality found in modern email clients. Upon pressing the button, a window pops up with an interface for defining a set of recipients.

Since a definition must be formulated in a particular ontology, either the user either be able to specify a query in the ontology directly, say via a textual editor, or he must use some sort of tool to that facilitates query formulation in that ontology. As only power users can be expected to formulate queries directly, a graphical interface for query creation is useful here. To support the possibility of using various ontologies, the tool should be generic and generate query interfaces on the fly based on the ontology and some display metadata. The interface used by our prototype is shown in Figure 2. The interface is automatically generated based on the schema for a person and some metadata about how to display each field, for example as a text box or a drop down list. Here, an email is being sent to members

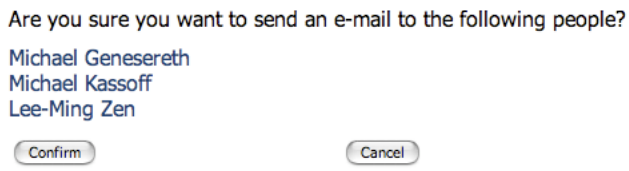

Figure 3: Confirmation page shown after pressing send

of the group lead by Michael Genesereth interested in logical spreadsheets. Note that it allows the emailer to define a set of people as opposed to a set of email addresses. As each person is associated with at most one email address, this also unambiguously defines the set of email addresses to send the message to.

Note how the interface of ISEA allows embedded queries of arbitrary depth to be formed. For example, one might query for all people at a site that is in a country where French is spoken. This cross-category search is particularly useful when large amounts of supporting information are available. In particular, consider that if the Semantic Web becomes a reality, then one will have the possibility of basing his query on arbitrary information on the Web.

Semantic email addressing is a powerful tool, which could easily be used to email large numbers of people. Safeguards must be put in place to make sure that user errors do not result in a mass spamming. It is useful to give the user feedback on his query definition by displaying a list of people to which the email will be sent, in the case that the number of people is sufficiently small that this can be done. Otherwise, it is useful to display the number of people to which the email will be sent, along with a sample of those people. This allows the user to catch errors in his query definition, and give him confidence that he is not accidentally sending a large number of people a message not meant for them. Figure 3 shows a simple confirmation page displayed in ISEA after the send button is pressed after choosing the set of people defined in Figure 2.

As mentioned earlier, ISEA creates actual *semantic email addresses* that are essentially of the form query@domain. For example, in our prototype system, by sending an email to a properly encoded "People in the Logic Group interested in logical spreadsheets"@logic.stanford.edu, an email will be sent to the appropriate people (where the quoted part is replaced by an appropriate computer-interpretable string). This is useful for sending mail from email clients that do not support semantic email addressing, for adding semantic email addresses to one's address book, etc. However, in an ideal world in which all email clients supported semantic email addressing, these semantic email addresses would be completely hidden from the user, as they are long, unintelligible, and ugly.[1] Instead of viewing such ungainly identifiers, we expect users to always view semantic email addresses though some graphical representation like the one shown 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 user names are associated with normal email addresses. In Figure 2, the nickname "PrediCalc Team" is given to the aforementioned semantic email address.

Upon receiving a semantically addressed email, it is sometimes useful to give the receiver the ability to view both the semantic definition of who the email was sent to and the actual people to whom the email was sent. The former can be accomplished by viewing the semantic email address through an appropriate GUI or natural language display, in the case that the email client supports semantic email addressing, or by clicking a Web page link automatically generated by the system and placed in the footer of the message by the sending agent. The latter is a bit more tricky, for the reason that the set of email addresses defined by a particular query may change over time. In the case that the number of email addresses is small, the actual email addresses to which the email was sent might appear in the To or Cc fields of the email, but if the number of addresses is large, this becomes impractical. In this case, the sending agent must somehow keep track of either the set of addresses sent to by each message or enough of the state of the database 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. In the case that the addresses of the recipients are sent in the To or CC fields of the message, one can reply in the usual manner to the recipients of the original message, or they can use semantic email addressing to reply. In the case that the recipient list is not included in the message, semantic email addressing must be used to reply. However, note that the set of people satisfying a condition may change over time, so replying can result in new people receiving the reply and old people not receiving the reply, which may or may not be the desired behavior. Of course, this is nothing new - the subscriber list to a traditional mailing list changes over time as well.

Security and Privacy

SEA is ideal for targeted email addressing, essentially the opposite of spam. However, SEA is also ripe for abuse - it can easily be used to send exceedingly untargeted emails if one desires. Within small to medium sized communities, this sort of behavior is often uncommon since social repercussions occur when a community member violates community rules. Within larger communities, security policies may need to be enacted to limit who is allowed to send to whom. On the public Internet, however, SEA abuse is much harder to control. Of course, this is true for any publicly available

[1] In our prototype, the semantic email address of Figure 2 is represented as 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. We make no claims as to the optimality of such an encoding, but present it as an illustration of why such addresses should be hidden from users.

email address, whether or not SEA is used.

An interesting possibility is for people outside an organization to use SEA to send email to members of an organization based on their semantic description, without knowing who they are. For example, I might want to send a message to the lead developer of a particular product at Yahoo!, but I do not know his or her name or email address, and Yahoo! might not want me to know his or her name or email address. However, they still might want to allow me to contact him or her. By exposing a semantic email addressing form to me, they can allow this is a simple way.

Related Work

Several other authors have seen the value in bringing semantics to emails, though all in a somewhat different way. For example, the Information Lens system (Malone *et al.* 1987) allowed users to send semistructured email messages and to filter those messages using production rules. An interesting feature of this system was that it allowed users to send to a special mailbox called "Anyone" from which anyone could subsequently choose to receive messages based on production rules. This flips on its head the nature of widely broadcast emails - instead of starting with receiving all emails and whittling them down based on filtering rules, the user instead starts with an empty inbox and pulls in emails of interest based on rules. This is quite similar to the subscription model of RSS, the syndication format popular today. As RSS feeds contain more and more semantic information, the semantic subscription model exemplified by Information Lens may become more commonplace.

More recently, *MailSMORE* (Kalyanpur *et al.*) allows the user to annotate the content of an email with RDF triples, and automatically also includes RDF triples generated based on the standard headers of an email such as the To, From, Subject and Body fields. This can be used for semantic filtering and filing of emails.

The MANGROVE system (Mcdowell *et al.* 2004a; McDowell *et al.* 2004b; Etzioni *et al.* 2003) takes this idea further but allowing not just structured email content, but also semantic email processes which allow email clients to be scripted with declarative workflow that can be used to automatically aggregate information obtained from many email responses, automatically resend emails to people who have not responded, or to analyze the semantic content of incoming email messages and respond accordingly.

Most relevantly, Microsoft Exchange 2003 allows administrators to create query-based distribution groups, which are essentially mailing lists whose recipients are based on an LDAP query run at the time of the email sending. This alleviates much of the administrative work required to maintain a mailing list. However, because the mailing lists may be created by an administrator only, they do not allow the full power of SEA. In fact, users are completely shielded from the fact that a distribution group is query based: each query-based distribution group is given a name, and thus looks just like a regular mailing list to an outsider.

Future Research

An interesting possibility to help control spam is semantic filtering and filing of emails. One could write semantic email rules based on not just the standard email fields, but also the semantic email address. For example, one might classify all email as spam that comes from someone not in a whitelist and without a FOAF that references someone within two degrees of separation and no common interest. This semantic filtering and filing could be extremely effective, especially if combined with semantic email content. How to generate user interfaces that allow these sort of filters to be specified and how to efficiently filter large numbers of emails against a semantic email address are interesting questions.

Conclusion

In this paper, we have introduced the notion of semantic email addressing. The targeted nature of a semantically addressed email is powerful and could be used to combat unintentional spam. Furthermore, since it can facilitate the contacting of individuals based on their characteristics, it can be used to preserve the privacy of email addresses and even individual identities. SEA is a simple concept, and one which we hope will soon be incorporated into commercial email systems.

References

L. Ding, L. Zhou, T. Finin, and A. Joshi. How the semantic web is being used: An analysis of foaf documents. *HICSS*, 2005.

O. Etzioni, A. Halevy, H. Levy, and L. McDowell. Semantic email: Adding lightweight data manipulation capabilities to the email habitat, 2003.

M. Genesereth, A. Keller, and O. Duschka. Infomaster: An information integration system. In J. Peckham, editor, *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, pages 539–542. ACM Press, 1997.

A. Kalyanpur, B. Parsia, J. Hendler, and J. Golbeck. Smore - semantic markup, ontology, and rdf editor. *http://www.mindswap.org/papers/SMORE.pdf*.

T. Malone, K. Grant, F. Turbak, S. Brobst, and M. Cohen. Intelligent information-sharing systems. *Commun. ACM*, 30(5):390–402, 1987.

L. Mcdowell, O. Etzioni, and A. Halevy. Semantic email: Theory and applications. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(2):153–183, December 2004.

L. McDowell, O. Etzioni, A. Halevy, and H. Levy. Semantic email. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 244–254, New York, NY, USA, 2004. ACM Press.