E-mail List Equations to Create Specific Audiences

Erik Blankinship
MIT Media Lab
20 Ames St.
Cambridge, MA 02139 USA
erikb@media.mit.edu

ABSTRACT

E-mail lists are useful for communicating with groups, but it is difficult to refine the intended audience. When planning a surprise party, for example, it is time consuming to delete the guest of honor's address from the multiple e-mail lists one wants to contact. We have built a tool that makes it easy to modify e-mail lists and, thus, contact specific audiences. The tool provides the computational mechanism for solving e-mail equations. An equation for the aforementioned example could be ((partyPeople@mit.edu & dancers@mit.edu) - joe@mit.edu). This equation would resolve to address all recipients of two e-mail lists except for joe@mit.edu. E-mail list addition, subtraction, intersection and xor can be computed with our tool. The tool already has been made available to a limited audience and overviews of some applications are presented in this paper.

Keywords

E-mail, E-mail lists, online communication

INTRODUCTION

The interests of people on a college campus are often varied; a combination of academics, residence life, and extra-curricular activities hold many students' attention. There is an audience for nearly all of these interests via e-mail lists. E-mail lists are self-organizing in that individuals add themselves to mailing lists consisting of people with shared interests, often at a career fair or using a student group's web site. Interests are varied from yoga-lovers to vegetarians-on-campus. Other e-mail lists are based upon location, such as a laboratory or residence hall.

E-mail addressed to a list reaches all of the subscribed members. It is easy to address multiple lists; one merely adds the lists' mailing addresses to the TO: field of their message. This technique works well when contacting large groups of people. A disadvantage is that these group e-mails are often regarded as spam. It is difficult to address a subset of a group's membership. In part, this difficulty stems from not knowing how to articulate the intended audience. Another difficulty is that the membership of lists can always change since they are on a network. In an attempt to address these issues, we created a tool that allows for the resolution of e-mail equations.

Treating e-mail lists as sets of addresses, our tool performs set algebra on equations in which e-mail lists are values. The results are refined lists of specific users' email addresses. Algebraic notation is used to create the equations, complete with clauses and order of operations. Examples of some equations and their application are presented later in the paper.

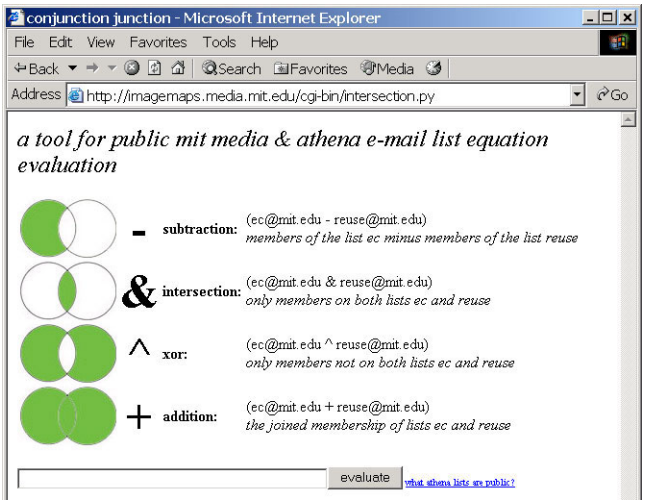

Figure 1. The web interface to the e-mail equation tool. A table provides reference to the set algebraic operators available. Venn diagrams graphically depict the set operations. A text field at the bottom of the page is for entering an e-mail equation, and pressing a form submit button results in a new web page with the solved for e-mail addresses.

RELATED WORK

Present e-mail applications do not fulfill the needs addressed by this tool. A freeware tool, E-Mail Manager [1], provides much of the same set algebraic functionality for local address lists. However, this tool does not help when addressing mailing lists on a network in which list membership can frequently change.

Another related project, Ephemeral Interest Groups [2], automatically creates branching discussion lists for furthering conversations not of interest to the entire original recipient list. The generation of Ephemeral Interest Groups does not address how to initially address a subset of mailing lists; rather it creates subset mailing lists for those who respond to an initial e-mail.

A web-tool called MediaLabbber [3] displays the membership of network e-mail lists, and also displays an

individual's memberships to multiple e-mail lists. This tool is useful for learning about a person's e-mail list affiliations, but does not help to automate audience creation.

TECHNICAL OVERVIEW

We set up our application as described below in our research lab. A python script runs as a CGI on an Apache web server. When a web user submits an e-mail equation, it first identifies all e-mail addresses in their query. The script contacts the appropriate list server to get list members. It is able to resolve list membership on public lists on the MIT Media Lab and MIT Athena servers using server side scripts. These scripts recursively explode any e-mail lists subscribed to other e-mail lists. Once each e-mail address is returned as a set of unique non-list e-mail addresses, it then evaluates the equation. If our server does not have permission to query another server for list members (i.e., a yahoo mailing list), that address is treated as a non-list e-mail address. The same is true for private lists withholding their membership.

The equation (coffeeDrinkers@mit.edu & operaSingers@mit.edu) reads as "coffeeDrinkers@mit.edu intersect operaSingers@mit.edu". We override the Python arithmetic operators + - / *, allowing set algebra of the lists. We map the & symbol to the / symbol to represent intersection. ^ is mapped to the * to represent xor. These changes were made so as to not confuse users who might think then can multiply or divide the sets. We also display Venn diagrams on the web page to provide an alternate representation and reminder of set algebra [Figure 1]. After submitting an equation through the web form, results are posted to the web page as a list of the solved for e-mail addresses. These can be copied and pasted into an e-mail. A MAILTO: link is also created, which when clicked creates an e-mail message in the user's e-mail client with the equation and all of the addresses added to the TO: field.

EXAMPLES

The e-mail equation tool, dubbed Conjunction Junction, was made available to members of our research group and an undergraduate dormitory floor for general use. Different applications for the tool have been observed (actual e-mail addresses have been changed for anonymity and legibility):

1. (dormFloor@mit.edu & currentResidents@mit.edu), has been used to address currently enrolled students who live on a dormitory floor. Since the floor's mailing list, dormFloor@mit.edu has many subscribed alumni, the Resident Assistant on the floor has made use of the tool to only reach residents who need to know current floor business.
2. ((dancers@mit.edu + partyPeople@mit.edu + coolDudes@mit.edu) - joe@mit.edu) has been used to address the membership of multiple lists, with possibly overlapping membership, with the removal of one person despite membership to any of the mailing lists. This is useful for planning surprise parties or leaving someone out of the loop.
3. (chemistry101@mit.edu & (dormFloor@mit.edu + otherDormFloor@mit.edu)) has been used to help solve the problem described as "who might understand the homework and is someone I know". The use of clauses as shown in this equation demonstrates order of operations: first the memberships of the two dorm floors are combined and then checked for intersection with the class list.

ISSUES

Private e-mail lists do not reveal their membership unless you are a member. This renders our tool incapable of returning the members of that list. To help with this problem, we provided a link to a page listing all of the public mailing lists available. A better solution might be to make the script available for users to run themselves, therefore providing all of their own permissions to various e-mail list memberships.

The xor operator has hardly, if ever, been used in e-mail equations so far. It might be that such an operator has little functionality in how people define audiences they want to reach. Some users are not familiar with set algebra and find the creation of equations difficult. However, we have found that despite this difficulty, people become excited when determining how to address the right group of people. There is an intrinsic motivation to think abstractly when the goal is communication with a specific audience.

ACKNOWLEDGMENTS

Thanks to Brian K Smith and Bakhtiar Mikhak for encouraging this work. Thanks to Walter Bender for contributing his perl code to the effort. Thanks also to the generous sponsors of the Media Lab's information: organized and Digital Life consortiums.

REFERENCES

1. Brothers, Laurence, Jim Hollan, Jakob Nielsen, Scott Stornetta, Steve Abney, George Furnas, and Michael Littman.1992. Supporting Informal Communication via Ephemeral Interest Groups. (eds), Proceedings of ACM CSCW'92 Conference on Computer-Supported Cooperative Work, 84-90.
2. Tarabrin, E. E-mail Manager v.1.0. Available at http://www.geocities.com/tarabrin2000/prog1/prog1.htm.
3. Van Dyke, N. Medialabber. Available at http://www.neilvandyke.org/medialabber