Dynamic Online Communities: DynaMail

Technical Report Number: CS03-21-00

Aleksandar Manojlovic
Department of Computer Science
University of Cape Town
amanojlo@cs.uct.ac.za

Ryan Slade
Department of Computer Science
University of Cape Town
rslade@cs.uct.ac.za

Hussein Suleman
Department of Computer Science
University of Cape Town
hussein@cs.uct.ac.za

ABSTRACT
DynaMail is a Mailing List Manager which automatically creates mailing lists based on user attributes and a history of their message interactions. A DynaMail Mailing List Manager creates dynamic mailing lists by providing a single email address to which messages are sent. From there, messages are distributed to subscribers belonging to different mailing lists. In a DynaMail system there are no actual mailing lists stored on the system, the user attributes define the mailing lists which are computed and generated dynamically. These attributes are entered into the system using a Web-based system which allows users and administrators to configure the system to their needs.

Categories and Subject Descriptors
H.4.3 [**Communications Applications**]: Electronic Mail

General Terms
Design, Human Factors

Keywords
Online Communities, Dynamic Mailing Lists

1. INTRODUCTION
An Electronic Mailing List is a community-building tool that connects people via email messages. There is one central address to which everyone sends messages for the group, and from there the email message is sent out to all the subscribers. A person receiving the email will usually have the choice of responding to the sender individually, or to the whole list.

This paper describes the design and evaluation of a new type of a Mailing List Manager. The Mailing List Manager named DynaMail allows the creation of dynamic mailing lists based on user attributes and a history of their message interactions.

Traditionally mailing lists are static, meaning that once a list is set up, if a user decides to send a message to the mailing list address, the message will be delivered to all of the members within that list only. Sending an email message to multiple mailing lists requires the user to manually send the message to each list. Since mailing lists consist of a group of people who are linked together through similar interests or personal connections, we can view a mailing list as a community of people who use email as a communications medium.

If we define a group as users who share a common interest, then we can view a collection of groups as a single online community- each group by itself represents a subset of the online community. A dynamic Mailing List Manager tries to model a single community as a set of smaller sub communities. By dividing the members of a mailing list into a number of sub communities and giving each user a descriptive attribute, a dynamic mailing list tries to make better sense of the connections that the different mailing list members have among one another. A dynamic mailing list attempts automatic selection of users who should receive a message sent to the mailing list (i.e., the users who form a part of a community). The ability to automatically select the members who will form a part of the community adds a dynamic element to the original static mailing list. By dynamically creating mailing lists, the actual creation of a mailing list will occur only once a message has been sent out to the mailing list by a member.

2. BACKGROUND AND MOTIVATION

2.1 Online Communities

A community can be defined as a group of people who share common interests (e.g., film community, music community). An online community is a community that uses Internet technology for members to interact with one another and share similar interests.

Online communities usually have a dual use. Firstly they can act as a community builder, or secondly, a community maintainer. As a community building tool, online communities can be used to draw in people with similar interests and build up a community which did not exist in the first place. An online community as a community maintainer could be used to support an already existing physical community. If a physical community already exists, online communities can act as a facilitator for this type of community to interact. Whether the Internet technology is used to support a community with or without a physical existence, once the community gains online presence, the functionality required by both will be very similar. McNamara deriving from his case study [8] suggests avoiding temptation to use the most sophisticated online tools, unless they are specifically suited to the purpose. Even though the tools used to support online communities keep evolving, McNamara suggests that a simple mailing list can sometimes be the best tool of all. While a mailing

list can often be used to support online communities, there is a need to moderate a mailing list in order to maintain the focus of the discussion. If Web presence is required for a community using a mailing list, a Web site could be provided to the members, expanding the current mailing list functionality. A Web site which supports a mailing list is usually used to archive the discussion occurring within the community and present it in a searchable format.

2.2 Current Solutions

The following are some of the more popular implementations of static mailing list solutions.

Majordomo (http://www.greatcircle.com/majordomo) is a program used to automate administration and management of mailing lists. In the Majordomo world model [9] there are three types of people: users, mailing list owners and the owner of the Majordomo server itself. Upon setting up a Majordomo mailing list, most of the operations can be performed remotely by sending an email message to the Majordomo server. The type of operations offered via email allows users to subscribe, unsubscribe or retrieve the mailing lists offered by the server. The users are also able to obtain information about these lists by sending email requests to Majordomo.

Unlike Majordomo, the DynaMail Mailing List Manager provides a Web page which allows the user to perform all of the administration functions. The DynaMail Web interface allows the user to view the mailing lists offered by the system, and subscribe or unsubscribe to the different groups. In this respect, DynaMail administration is similar to the Mailman Mailing List Manager.

Mailman (http://www.gnu.org/software/mailman/mailman.html) is another program used for management of mailing lists, similar to Majordomo. One of the differences between Mailman and Majordomo is that Mailman provides a Web page for each mailing list. Instead of sending messages to the mailing list in order to subscribe to a mailing list, a Web page allows users to perform the required operations through the page. Mailman offers a large number of features to its subscribers, some of the more significant being: message archiving, Spam prevention, email-based admin commands and support for virtual domains. Another interesting feature of Mailman is its automatic mail bouncing [8]. If a delivery error occurs, Mailman is able to use pattern matching in order to determine the undeliverable email addresses. Once an address has been found to be undeliverable, the address becomes disabled. Mailman also contains several Spam prevention mechanisms which attempt to reduce the number of Spam messages sent out to list members.

The DynaMail Mailing List Manager attempts to prevent Spam messages by only allowing registered members to send email messages to a DynaMail mailing list. All of the information about registered members is kept in a database. One of the current Mailing List Managers which uses a database to store the users belonging to a mailing list is the Sympa Mailing List Manager.

Sympa (http://www.sympa.org) is a Mailing List Manager which automates mailing list management and administration. While Mailman and Majordomo both store the list of users in a flat file structure, Sympa is able to read a list of users from a database. Sympa also allows several ways of user authentication. The method of authentication can be based on a password, an SMTP "From" header or an S/MIME signature. Much like Mailman, Sympa performs archiving of email messages and Spam prevention. Archiving and displaying of messages in Sympa is performed by the MHonArc mail archiving program. MHonArc converts email messages into HTML format and provides a way to thread link the converted messages.

Similar to Sympa, DynaMail stores users in a database. The difference between the two systems, however, is that DynaMail attempts to make connections between the users to decide who should receive mails.

3. METHOD

3.1 DynaMail Overview

The DynaMail Mailing List Manager consists of two parts, namely the Web interface and the Mailing List Manager server.

Each user of the DynaMail system is required to register with the system. The registration process is performed through the Web interface. During the registration process, a user is required to select certain attributes which determine the sub communities that the user will form a part of. Once a user sends a message to the DynaMail server, a mailing list is created by querying the database and extracting all users who share common attributes. Users set up their "common interests" by subscribing to subgroups of the list using the Web interface. The groups themselves are added to the system by an administrator, also using the Web interface.

DynaMail attempts to model an online community as a single mailing list which consists of a number of sub lists. Unlike the process of each user manually creating a list consisting of sub lists, DynaMail tries to move this function away from the user and on to the Mailing List Manager. In this way DynaMail allows automatic distribution of email messages among the different sub communities.

3.2 DynaMail Server

DynaMail mailing list server is designed to accept messages from a Mail Transport Agent (MTA) and, based on the queries performed on the database (MySQL), generate a mailing list. The MTA used for the implementation of DynaMail is Sendmail.

DynaMail sends and receives messages in XML format. The XML format was implemented since it enables interoperability between the possible entities which could be used to communicate with DynaMail. In order to provide XML support, DynaMail's design is based on a Three-Tier Architecture, where a middle tier converts messages to XML format and back from XML into the original message format.

By converting email messages into an XML format, the original format of the message is abstracted. Since Sendmail operates with messages in a non XML format, a translation mechanism was implemented in order to translate a message into XML format.

The operating process begins with the user sending an email message to the DynaMail server (e.g. DynaMail@cs.uct.ac.za) as described in Figure 1. The message sent by a user is then received by Sendmail Mail Transport Agent (step 1). Sendmail forwards the message to the XML converter (step 2) which translates the message from the standard email message format (RFC 822) into XML format by adding additional tags. The XML converter's sole purpose is to convert the mail messages into XML format since the DynaMail Server only accepts messages in this format. Sending the messages in XML format would allow the DynaMail Server to accept messages from a variety of sources, as long as the XML tag format is adhered to.

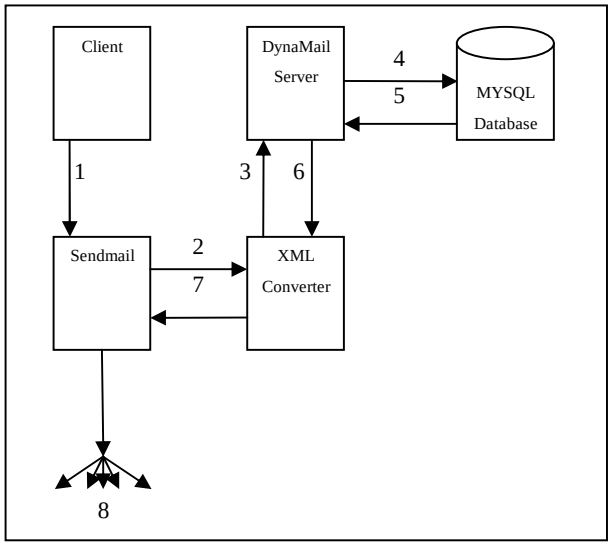

Figure 1 DynaMail Operation Diagram

Once the mail message has been converted to the XML format, the message is forwarded to the DynaMail server (step 3).

The originating e-mail address is used to extract the required group members from the database. These are the members that have the same group properties (e.g., belong to the same sub communities) as the person who has originally sent the message. User authentication is based on a user's email address. The message address verification is done by querying the database with an email address to see whether the query would yield a list of results (step 4). Once this query has been submitted a list of addresses to which the original message should be delivered to is generated (step 5). If no list is generated, no further processing occurs and the message is discarded. This would mean that either the user doesn't belong to the system or that there are no users belonging to the system with the same sub group properties. If a mailing list does get generated, this proves that the user is a member of the system. Upon generation of the mailing list the message ID and the email address of the member are saved in the database. The message ID and the address are used later when users reply to messages. For each user in the newly created list, a separate message is created (in XML format), and sent back to the XML converter (step 6). The XML converter strips the XML tags from the messages and forwards the plain messages back to the Sendmail program (step 7). Once Sendmail receives the messages from the XML converter, Sendmail will be responsible for distributing them to the specified clients.

The process described above is performed if a new message is sent to the DynaMail server. Once a recipient of a message replies to the original message, the message ID of the message is checked against other messages stored in the system. If a message is found to be a reply to a message previously sent to the DynaMail server, this reply message should only be received by the users who belong to the groups that contain both the original sender and the person replying to the message. The new mailing list is obtained by intersecting the groups of these two users.

3.3 DynaMail Web interface

In order for users and administrators to make use of the system, there needs to be a method for them to interact with the system so that users and groups can be added. The choice was made to implement this using an open ended three tier design. The reason for this is that the actual back-end of the system is created independently to the way that users will interact with the system. The back-end will receive XML requests as its input, and will respond using XML. This means that anyone communicating with the third tier only needs to know how to send the XML requests and interpret XML responses. It is then their job to translate this XML into their own desired format and output it. Of course, in order to show how this would work in practice, the first and second tiers of the system were also designed and implemented to show a working system.

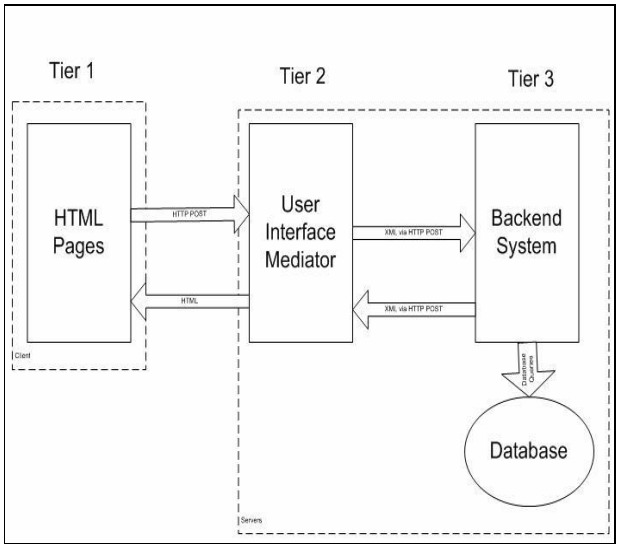

Figure 2 DynaMail Operation Diagram

User input was implemented using HTML forms and output was in the form of HTML Web-pages. The first tier is a combination of standard HTML pages as well as PHP scripts where necessary, which will pass HTTP requests to the second tier. The second tier will then convert these to the required XML format, which will be

forwarded to the third tier. The third tier will then perform the task required, updating or retrieving data from the database as necessary. When the second tier receives a response from the third tier, it will convert the XML back into HTML which it will send back to the first tier to display.

4. RESULTS

4.1 User Testing

The user tests for the DynaMail mailing list server were all performed on a local network. The user tests were performed using seven computer science students. Upon using the system the users were presented with a description of the system and a questionnaire which tried to examine the users' understanding of the system.

During the user testing phase, users were required to sign up to different sub groups offered by the DynaMail system. The sign up process was performed via the Web interface. One of the main problems that users had with the different sub groups offered was that each sub group was viewed as a separate static mailing list. Even though the DynaMail operation was explained to the users, a dynamic list was still compared to traditional static mailing lists by the users. The DynaMail system does not allow a user to send a message to a specific group of their choice, since the groups are created dynamically. Because of this, the users were not comfortable with the fact that a certain level of control over sending a message was taken away from them. In addition to having dynamically created mailing lists, the users would have preferred to be able to select the different subgroups offered instead of letting the system select the groups automatically for the users.

In general the users did not have problems understanding the process behind creating dynamic mailing lists. Even though the users were not completely satisfied with the restricted functionality offered by DynaMail, certain users thought that a dynamic mailing list would be useful for collaboration purposes.

Another aspect of the system examined from the user's point of view was whether the users required knowing the recipient of a sent message. The users in general were not concerned with the fact that they did not know who the receiver of a sent message was, since the receiver is assumed to share the same interests as the sender of the email message. However, users were concerned that they should never receive messages from someone who they did not know.

4.2 Performance

The DynaMail Mailing List Manager performance depends on two different technologies for correct operation. A MySQL database is used to retrieve information about the users and a Mail Transport Agent (Sendmail) is used to perform delivery of email messages. From the user's perspective, the performance of the email sending process remains unchanged. This is because sending an email message is a connectionless operation. An email message sent to a DynaMail mailing list is first received by the Mail Transport Agent (MTA), which in DynaMail's case is Sendmail. Instead of Sendmail delivering the email message directly to the recipient, the message is forwarded to the DynaMail system. The only extra time required to deliver an email message will be incurred by performing queries to a MySQL database. Once the user was registered with the DynaMail system through the Web interface, they were required to send an email message to another member of the dynamic mailing list. The first message sent by the user was performed via the DynaMail server, and a second message was sent directly to the recipient without DynaMail's intervention. In sending and receiving an email message, the users could not detect a difference between performing these tasks through the DynaMail server and sending a message directly to a user on a local network.

4.3 DynaMail Design Evaluation

One of the goals behind the design of the DynaMail Mailing List Manager was to make the system transparent to the user, by allowing them to use normal e-mail programs to send messages. By providing a single point of access, a user sending a message to one mailing list address is able to send messages to mailing list subscribers who could belong to a number of different sub lists. While a user is able to become a part of a number of sub communities within a single community, if a user becomes a part of multiple communities, DynaMail will not be able to keep the system transparent to the user and still be able to deliver a message to the correct community. This is because DynaMail will not be able to distinguish which community the user intended to send the message to. If a user requires being a part of more than one community, the user would now have to indicate, on the message sent, which community the message needs to be delivered to. In essence, being able to indicate which community a message needs to be delivered to, provides the same functionality as a static mailing list. This feature was not implemented by DynaMail since the main focus was on creating dynamic mailing lists, and preserving transparency to the user.

The Web interface design proved to be successful since it allowed testing of the third tier to be done independently of the other two tiers. This ensured that before the interface itself was implemented, the back-end was known to work. From this point, the other two tiers could be implemented, knowing that if any errors occurred, they could be narrowed down to the first two tiers. The design also makes it possible for the system to be integrated into other systems that use a different method of displaying data to the user.

5. CONCLUSION

Because of the way DynaMail's dynamic mailing lists are created, for the connections between users to seem sensible, a DynaMail user will have to belong to a sub community which forms a part of a larger community, with the same general interests. This is because the system does not know the context of messages sent, and therefore relies on the way that the groups are created to compute the list of users that messages will be sent to.

Since all of the user information is stored on a DynaMail server, the users are no longer required to remember all of the mailing lists which they are members of, or select a mailing list which a message should go to. A dynamic mailing list attempts to move this function away from the user and on to the Mailing List Manager.

Even though user information is stored on the DynaMail server, the transparency of the system from the users' point of view can only be preserved to a certain degree. If the functionality of a static mailing list is required by a user, the user will have to provide some extra information when sending the message.

Dynamic mailing lists do not attempt to replace the functionality of a static mailing list, but rather complement their functionality. It was found that there are specific cases where DynaMail could be successfully deployed, such as a single community made up of sub communities that share a common interest. In these scenarios, a Dynamic Mailing List could provide a more sensible way of distributing messages over a traditional static mailing list. Even though there are a number of situations to which dynamic mailing lists can be applied, in a more general situation it would appear that static mailing lists provide a better solution to user needs. This is because they offer more control to the user over dynamic mailing lists.

6. REFERENCES

1. Aumont, S. Salaun O. Wolfhugel C. Sympa Mailing Lists Management Software. (2003). Available http://www.sympa.org/doc/sympa.pdf.
2. Buckman, J. History of List Servers. (2003). Available http://www.lyris.com/about/company/whitepapers/listserver_history.pdf.
3. Chapman, D. Majordomo: How I Manage 17 Mailing Lists Without Answering "– request" Mail. (2003). Available http://www.greatcircle.com/majordomo/majordomo.lisa6.pdf
4. Cowan R King, M.. Tips on facilitating a social change email list. (2003) Available http://democracygroups.org/mailinglisthowto.html.
5. Fernback, J. Thompson B. Virtual communities: Abort, Retry, Failure. (2003). Available http://www.well.com/user/hlr/texts/VCcivil.html.
6. Kim, J. Community Building on the Web. (2003). Available http://www.naima.com/community/.
7. Manheimer K. Viega, J. Warsaw B. Mailman: The GNU Mailing List Manager Availabe. (2003). http://www.usenix.org/publications/library/proceedings/bsdcon02/cfp/viega.pdf.
8. McNamarra, K.S. Case Study: Building an Online Community. (2003). Available http://www.worldbank.org/devforum/case_eleven.html.
9. Morrow, K. Building Online communities. (2003). Available http://www.itrainonline.org/itrainonline/english/communication.shtml.
10. Shirky, C. Social Software and the politics of groups. (2003). Available http://shirky.com/writings/group_politics.html.
11. White, N. How some folks have tried to describe a community. (2003). Available http://www.fullcirc.com/community/definingcommunity.htm