グループウェアと
ネットワークサービス 43－6
（２００２．３．２２）

柔軟な宛先アドレスシステム fms(flexible email system) の設計

間宮 章彦† 山根 信二‡ 村山 優子‡

†岩手県立大学大学院 ソフトウェア情報学研究科
‡岩手県立大学 ソフトウェア情報学部

電子メールでグループ通信を行う方法として，メーリングリストがある．メーリングリストにおける購読者リストの保守は，管理者が手作業で行うか，自動メーリングリスト管理システムを使用して行われる．自動メーリングリスト管理システムは様々な購読管理処理を自動化するが，メッセージ毎に，一時的にメンバの変更を行うことはできない． その対策として, 本稿では宛先アドレスに集合演算を導入するシステムの設計について報告する. さらに，メーリングリストに複数のドメインが含まれる場合には他のメールサーバと協調して動作する必要がある．この協調方法について考察を行った.

The design of a flexible email system

Akihiko Mamiya† Shinji Yamane‡ Yuko Murayama‡
†Graduate School of Software and Information Science
Iwate Prefectural University
‡Faculty of Software and Information Science
Iwate Prefectural University

Mailing lists are used as a means of group communication on with email. The members of a mailing list are managed manually, alternatively by making a use of a mailing list management system. Although a mailing list management system automates various management processes, it is quite difficult to change members of a list only temporarily. By incorporating set-theory operations to email destination addresses, we have dealt with this problem. This paper reports the design of such a mail system. Moreover, we raise such issue that when two or more domains are included in a mailing list, it is necessary to cooperate with other mail servers.

1 はじめに

本稿では宛先アドレスに集合演算を用いることにより，メッセージ毎に一時的にメンバの変更を行い，動的で柔軟な宛先指定を実現するシステム，flexible email system (fms) の機能と設計について述べる.

まず，従来のメーリングリストについて述べ，本システムの概要及び設計，そして実装にあたっての検討すべき課題を示す．電子メールはコンピュータネットワーク上の古典的なアプリケーションである．これまでは電子メールでグループ通信を行う場合，電子メールの To,Cc,Bcc フィールドにメンバを羅列することにより，同じメッセージを多くの人への同報が可能である．この方法では，グループのメンバが増えたり，減ったりしたとき，グループ内の全てのメンバが宛先を更新しなければならない.

また，アドレスが増えると宛先アドレスの入力ミスなどの可能性が大きくなる．さらに，宛先にメールアドレスが羅列されたメールを受け取った受信者はそのメールがどのようなグループに送信されたかという，グループの意味合いが伝わりにくい．

電子メールでグループ通信を行う他の方法として，メーリングリストの利用がある．メッセージを１つのアドレス（リストアドレス）に送信するだけで，そのメッセージがグループの全員に配送できる．メーリングリストのアドレスは通常，そのメーリングリストのグループの意味が分かる名前にされることから，受信者はその宛先を見ることにより，送信されたメールがどのグループに送信されたかを認識することができる．メーリングリストを用いる場合，送信する宛先は１つだけですみ，宛先アドレスの入力ミスが起こりにくいという利点もある．

メーリングリストにおける購読者リストの保守は，管理者が手作業で行うか，自動メーリングリスト管理システム（majordomo[9] など）を使用して行われる．自動メーリングリスト管理システムは様々な購読管理処理を自動化するが，投稿毎に，一時的にメンバの変更を行うことはできない．

2　システム概要

2.1　メールアドレスのグループ管理

メーリングリストは１つのリストアドレスを，グループ全員のエイリアス（別名）として作成される．

エイリアスは電子メールのアドレスを別の名前で登録する機能で，sendmail[6] では別名（/etc/mail/aliases など）ファイルに書き込むことにより設定される．

例えば

```
saru:  taro
```

と書かれていれば，saru 宛なら sendmail が taro に宛先を書き換える．エイリアスは複数の電子メールアドレスに対しても別名が可能である．

```
animal: saru,inu,kiji
```

が最も単純な書き方で，animal 宛のメッセージを saru と inu と kiji に送るものである．宛先を以下のように，ファイルに書き込んでおくこともできる．

```
animal::include:/var/mail-lists/animal-list
```

上の例では，animal という宛先アドレスに対して，そのメンバを animal-list というファイルに指定している．

2.2　集合演算の導入

```
To:  tennis@example.org
```

上記の例は，ある組織のテニス仲間のグループに宛てる場合である．

このリストアドレスのグループはあらかじめ固定されており，メンバの変更を行う場合は，あらかじめ管理者が変更を行うか，メンバ自身が自動メーリングリスト管理システムを使用して変更する必要がある．

本システムは，宛先アドレスに演算子を組み込むことにより，メッセージ毎に一時的にメンバの変更を実現するシステムである．

利用できる演算子は以下のものである．

和　$A + B = \{x | x \in A \vee x \in B\}$

差　$A - B = \{x | x \in A \wedge x \notin B, B \subset A\}$

AND　$A \& B = \{x | x \in A \wedge x \in B\}$

括弧　演算の優先

例えば以下のような宛先指定が可能となる．

```
tennis@example.org
                - martina@example.net
```

この例ではテニス仲間のメールリストから martina@example.net というメンバーを除いた宛先を指定している．

2.3　集合演算の導入部位

演算子を用いた宛先アドレスの研究として舛本の研究がある [3]．舛本は演算子を含む宛先

アドレスを，独自のメールヘッダ (XTO:フィールド) に格納した．また，受信者に届くメールは，To:フィールドと Cc:フィールドに演算結果のメンバのメールアドレスが羅列されている．

この手法では，送信者がヘッダを編集できない MUA(Mail User Agent) を使用している場合，独自のメールヘッダの挿入することができないので，演算子を含む宛先アドレスを指定できない．受信者には宛先にメールアドレスが羅列されたメールが届く．このようなメールは受け取った受信者に「そのメールがどのようなグループに送信されたか」という，グループの意味合いが伝わりづらい．

本システムでは，従来から電子メールの宛先として用いられてきた To:フィールドのコメント部分に，演算子を含む宛先アドレスを導入することにより，より多くの MUA で利用できるように設計する．

3 システム設計

3.1 利便性の高い宛先記述方式についての考察

To:フィールドにコメントを挿入する方法は以下の 3 種類が定義されている [2]．以下の To:フィールドはいずれも `foo@example.org` 宛のメールで，コメントとして comments が記載される．

- `To: comments:foo@example.org;`
- `To: comments <foo@example.org>`
- `To: foo@example.org (comments)`[1]

複数の MUA で調査した結果，これらのコメントを用いた宛先アドレスで送信できないものがあった．これは RFC2822[2] にそった実装がなされている MUA が少ないためである．実装されていない仕様を採用することは避けるべきであるとの判断から，本システムでは，調査した全ての MUA で実装されていた，

- `To: comments <foo@example.org>`

というコメント方法を用いる．

コメント (comments) 部分には宛先アドレスにアットマーク (@) などの特殊文字が含まれることが予想されることからダブルクオート (") でコメント (comments) を囲む必要がある．

メールアドレスにはハイフン (-) が使われることがあるので，本システムでは演算子のマイナス (-) は (\-) にする．

例えば，前述したテニス仲間のメーリングリストから martina@example.net というメンバを除いた手先を指定する場合は以下のような To:フィールドとなる．

```
To: "tennis-players@example.org
   \- martina@example.net"
   <program@example.org>
```

3.2 システムの内部動作

上記の例では，テニス仲間のメーリングリストから `martina@example.net` というメンバを除いた集合演算を用いた宛先指定をし，fms プログラムへのエイリアスである `program@example.org` に宛先を指定している．

`program@example.org` に送信されたメールは UNIX のパイプで fms に渡される．メールを受け取った fms は集合演算を処理し，処理結果をメールのエンベロープ部分に書き込んで，MTA(Mail Transfer Agent) に渡され，演算結果の受信者に配送される．図 1 にシステムの動作を示す．

4 システムの問題点

このシステムで集合演算を解決するには，メーリングリストや，ほとんどの MTA が備えている組み込み機能を使うメールエイリアスを，個々のアドレスに分解し，演算をできる状態にしなくてはならない．

これは，メーリングリストのメンバに別のメールサーバで管理されているメーリングリストのメールアドレスが含まれる場合には，他のメールサーバで管理されているメーリングリス

[1] RFC822 では定義されているが，RFC2822 からは廃止された．

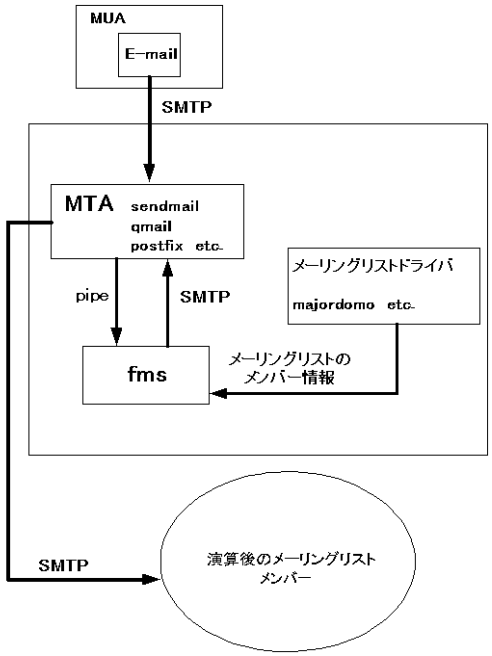

図 1: システムの構成

トのメンバ情報が必要になるということである. この問題には以下のような解決法が考えられる.

1. メールの集合演算のための中央集中処理型のサーバを用意し, そのサーバ宛に集合演算を含む電子メールを送り, そこでサーバに登録されたエイリアスだけを解決する (図 2).

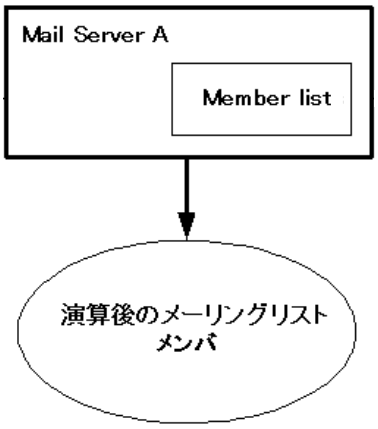

図 2: 他のサーバと協調しないときのモデル

2. 解析機能は送信ユーザ側のメールサーバに置き, グループのアドレスのメンバー分解などをすべて送信側で行う. 異ドメインのアドレスについては, そのドメインのメールサーバに問い合わせる (図 3).

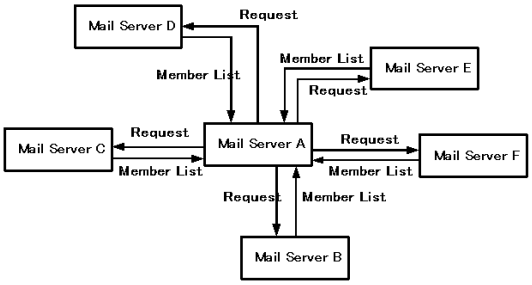

図 3: 他のサーバからメンバ情報を取得し，アドレス解決するモデル

3. 解析機能を各ドメインのメールサーバに置き, 各サーバは自分のドメイン内のアドレスや別名についての解決だけを担当し, 解決できないものいついては当該ドメインのメールサーバへ転送する. 最終的にすべてのアドレスについて別名検査がおこなわれた段階で集合演算を施す (図 4).

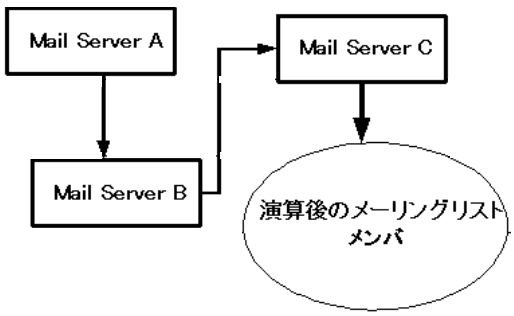

図 4: 他のサーバにカスケードでアドレス解決をするモデル

1 の手法は，メーリングリストのメンバ情報をあらかじめ構築しておく必要があり，メーリングリストの内容が変わった場合には，同期させる必要がある.

2 の手法は，SMTP(Simple Mail Transfer Protocol) の EXPN(expand:展開) コマンド [1] を利用する方法が考えられる.

EXPN コマンドはエイリアスを解決するためのコマンドである. しかし，EXPN コマンドは SMTP の必須コマンドではなく拡張コマ

ンドなので，サポートしていないサーバがあるという問題がある．一般的によく使用されていると思われる MTA で Sendmail[6], Postfix[7], qmail[8] について調査した結果，Sendmail は実装されているが，管理者によって使用不可に設定されていることがある．さらに，Postfix や qmail では実装さえなされていない．

また，メーリングリストの登録者数がある程度以上，多くなると，管理コストを減らすために，メーリングリスト管理ソフトを利用していることが多い．この場合は，MTA ではなく，メーリングリスト管理ソフトがメーリングリストのメンバ情報を持っているので，EXPN コマンドではメンバ情報を引き出せない．

3 の手法は，最終的にすべてのアドレスについて別名検査がおこなわれた段階で集合演算を施し，メールを配送するために，これらの仕組みを組み込んだメールサーバは第三者のメールを不特定多数に送信することができる．つまり，第三者中継 (Third-Party Mail Relay)[4, 5] を許してしまうという問題点がある．

例えば，以下のような脅威が考えられる．

1. A というメールサーバから B というメールサーバに「私のところでアドレスを分解できるだけ分解した結果，100 万通のメールは別名検査をしたが，B の管理しているメールアドレスが含まれていたので，このメールアドレスの別名検査を行ってください」という要求がくる．

2. B というサーバは B の管理しているメールアドレスの別名検査を行った結果，それはメーリングリストで，3 つのアドレスに分解できた．

3. B のメールサーバは演算結果の 100 万 3 通のメールを送信することになる．

5 今後の課題

5.1 メールサーバ (ドメイン) 間の信頼関係を設定する必要性

上記の 2 の手法ではメンバ情報を交換するプログラムを全てのメールサーバにインストールする方法も考えられる．このプログラムの役割はメールアドレスの別名検査を行い，その結果を要求のあったメールサーバに返すことである．しかし，第三者にメーリングリストのメンバ情報が漏洩してしまうという問題がある．3 の手法でも第三者中継のおそれれがあるという問題がある．これらの問題を解決するためにはメールサーバ (ドメイン) 間で信頼関係を設定する必要がある．

5.2 メールサーバ間の信頼の輪

4 章の 3 の例では，図 5 のようにメールサーバシステム同士がお互いに信頼しているグループ α とお互いに信頼しているグループ β で共通のホスト B がある場合や，図 6 のように B と F が個別に信頼関係がある場合にアドレス解決能力が高くなる． また，信頼関係を多く持っているメールサーバはアドレス解決能力の高いメールサーバとなり，利便性が高くなることが予想される．

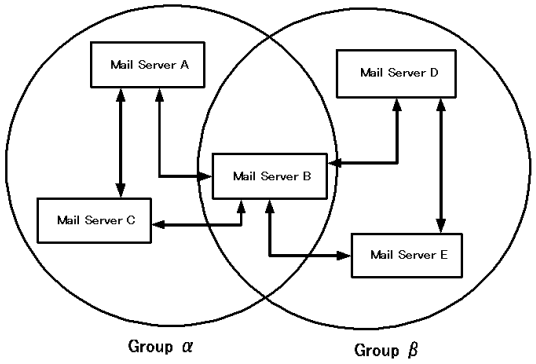

図 5: 一部，重なるグループ間の信頼関係

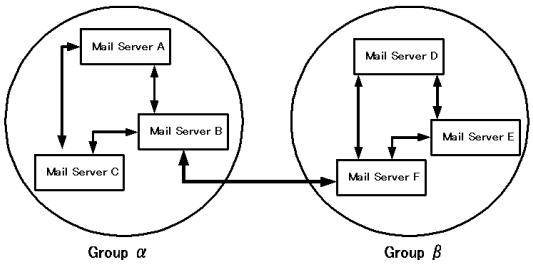

図 6: 重ならないグループ間の信頼関係

6 開発環境

これらの課題への対応を検討しながら，現在、単一ホストで動作するバージョンを開発中である．既に舛本 [3] の Lex,Yacc,C 言語によるプロトタイプが存在するが，SunOS4.x の C コンパイラに依存した実装である．また，一部の論理演算しか実装されていないプロトタイプであるため，再設計および perl による実装を進めている．一部 Flex[10], Bison[11] による実装も検討中である．

7 まとめ

本稿では宛先アドレスに集合演算を用いることにより，メッセージ毎に一時的にメンバの変更を行い，柔軟な宛先指定を実現するシステムの機能と設計について述べた．本システムの特徴は，独自のメールヘッダを使わず，既にある To:フィールドのコメント部分を利用することにより，MUA を選ばないで，サービスを利用できることである．また，このコメント部分への記述方法についての考察を行った．

このシステムで集合演算を解決するには，他のメールサーバと協調する必要がある．このため，メールサーバ (ドメイン) 間の信頼関係をあらかじめ設定する方法を提案した．この方法では信頼関係を多く持っているメールサーバはアドレス解決能力の高いメールサーバとなり，利便性が高くなることが予想される．

参考文献

[1] J. Klensin, Editor. “Simple Mail Transfer Protocol” RFC2821, April 2001. (Obsoletes RFC821, RFC974, RFC1869) (Status: Proposed Standard)

[2] P. Resnick, Editor. “Internet Message Format” RFC2822, April 2001. (Obsoletes RFC822) (Status: Proposed Standard)

[3] 舛本 克典. “メールアドレスへの集合演算の導入” 広島市立大学 情報科学部 情報工学科 4 年生卒業論文, 1998 年.

[4] S. Hambridge, A. Lunde. “DON’T SPEW: A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam*)” RFC2635, June 1999. (Status: Informational)

[5] Alan Schwartz, Simson Garfinkel. *Stopping Spam: Stamping Out Unwanted Email and News Postings.* O’Reilly & Associates. October 1998

[6] Sendmail.org, http://www.sendmail.org/, (Last access 23 Feb 2002)

[7] The Postfix Home Page, http://www.postfix.org/, (Last access 23 Feb 2002)

[8] D. J. Bernstein, The qmail home page, http://www.qmail.org/, (Last access 23 Feb 2002)

[9] Brent Chapman, John Rouillard, Chan Wilson. Majordomo home page, http://www.greatcircle.com/majordomo/, (Last access 23 Feb 2002)

[10] G.T.Nicol, Flex 入門, アスキー出版, 1999.

[11] Charles Donnelly, Richard M. Stallman, Bison 入門, アスキー出版, 1999.