

電子メールにおける柔軟な宛先アドレス指定の研究

間宮 章彦

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

1 はじめに

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利用されることが多い。メーリングリストはメッセージを1つのアドレスに送信するだけで、そのメッセージをグループ全員に配送できる。

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、投稿毎に、一時的にメンバの変更を行うことはできない。

本研究では、メーリングリストの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ーの変更が行える、柔軟なアドレス指定の実現を試みた。さらにメールサーバ間で協調してアドレス解決を行うという、機能をもたせた。これらの機能をもつシステムとしてFMS(Flexible e-Mail System)を提案し、実装および評価について述べる。

2 概要およびモデル

FMSのアドレス指定モデルを図1に示す。送信者は集合演算を含む宛先アドレス指定を行い、FMSに送信する。FMSは集合演算の構文解析を行い、必要なメンバ情報を取得し、演算処理を行った後、メールを配送する。メールの配送は既存のMTA(Mail Transfer Agent)の機能を用いる。

“The flexible destination address specification by cooperation of Email servers”

Akihiko Mamiya and Yuko Murayama

Graduate School of Software and Information Science, Iwate Prefectural University

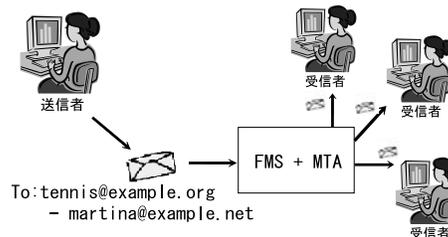


図 1: システムモデル

3 システム設計

3.1 集合演算を含む宛先指定

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

例えばテニス仲間のグループをメンバとするメーリングリストを `tennis@example.org` として、このメーリングリストから `martina@example.net` というメンバーを除いた場合、以下のような宛先指定が考えられる。

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

本システムでは、従来からある、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(Mail User Agent)で利用できるように設計した。集合演算を含む宛先アドレスの指定方法は、調査した全てのMUAで実装されていた、以下のようなコメント方法を用いた。

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

メールアドレスにはハイホン(-)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@example.org
      \- martina@example.net"
   <program@example.org>
```

3.2 メンバの情報収集

集合演算の演算には、メーリングリストや、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する必要がある。メーリングリストのメンバとして、他のメールサーバで管理されているメーリングリストのアドレスが含まれる場合、当該メールサーバで管理されているメーリングリストのメンバ情報が必要になる。このような場合、メールサーバ間での協調動作が求められる。

4 プロトタイプの動作概要

本システムのプロトタイプを実装した。図2に本プロトタイプの動作概要を示す。

本プロトタイプでは、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SMTP[1]で問い合わせる。この時、プロトタイプではあらかじめ設定されたメールサーバ間でのみ、メールサーバ間での協調を行う。

FMSに届いたメールは、まず集合演算を含む宛先アドレスを構文解析モジュールで解析し、次にメーリングリスト判別モジュールにより、他のメールサーバと協調を行うか、判定する。協調を行わない場合は、集合演算モジュールで演算し、SMTP送信モジュールによって、演算結果のメールアドレスに送信する。協調を行う場合は、SMTP送信モジュールによって、当該メールサーバに問い合わせを行う。この問い合わせの回答を受取り、全てのメンバ情報を集合演算モジュールで演算し、SMTP送信モジュールによって、演算結果のメールアドレスに送信する。

5 評価

従来からあるTo:フィールドに、集合演算を含む宛先アドレスを収容したことで、多くのMUAで利用できることを示すため、複数のMUAで送信実験を行い、動作検証を試みた。

その結果、Winbiff, Netscape, Outlook Express, Mailで正しく宛先指定が行えなかった。これらのMUAは自動で宛先アドレスの構文チェックを行う機能があり、集合演算を含む宛先指定にRFC2822[2]に準拠しない誤った変換が加えられるのが、原因であった。Winbiffでは宛先アドレスが4つのアドレスとして誤認識された。Netscapeでは集合演算を含む宛先アドレスの入力時にスペルチェック機能により自動で(\-)が(-)と変換された。Outlook Express

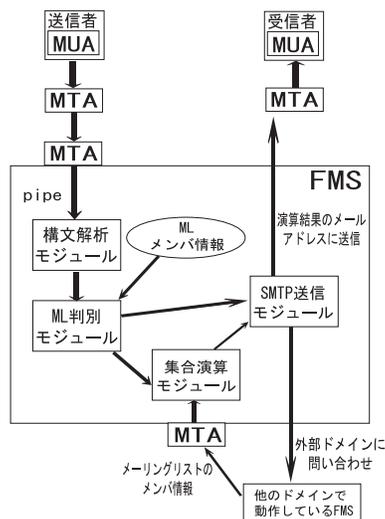


図2: プロトタイプ動作概要

では送信時に(\-)と(")が、それぞれ(\\-)と(\\")と変換された。MacOSX上のMailでは集合演算を含む宛先アドレスがJISエンコーディングに正規化されたために、アドレス解析に失敗した。

6 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を提案、実装し、その評価を行った。本システムは、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し、より多くのMUAでの動作を目指した。また、メールサーバ間で協調して動作し、より正確な演算を実現した。

複数のMUAで本システムの送信実験を行った結果、いくつかのMUAで送信できなかった。これは集合演算を含む宛先指定にRFC2822に準拠しない誤った変換が加えられるのが原因であった。それらのMUAの問題については今後の課題とする。

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)