DEIM Forum 2010 E2-2

RMXにおけるポリモルフィックルールとメール本文編集機能の導入

青山　陽亮† 　遠山元道††

†††慶應義塾大学理工学部情報工学科　〒223–8522 神奈川県横浜市港北区日吉 3–14–1
E-mail: †bluemountain@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

あらまし　ルールベースメール配送システム RMX において，メールアドレスに使用する配送ルールにいくつかの演算子が使用できる. その１つである'-' は従来，整数型の範囲を指定するだけであり，汎用性が低かった. そこで本論文では，ポリモルフィックなルールを提案し，配送ルール設定の汎用性を高めた.

また，ルールベースでデータベースからユーザ情報を自動で判別する RMX の特色を生かし，メール本文において配送ルールを文章ごとに指定することで，ユーザに適した文章をサーバ上で作成する機能と，あらかじめ作成しておいた定型文を，メール本文中の指定された場所にサーバ上で挿入する機能を提案する.

キーワード　RMX, 電子メール

Introduction of Polymorphic Rule and the E-mail Text Editing Function in RMX

Yosuke AOYAMA† and Motomichi TOYAMA††

†††Department of Information and Computer Science ,
Keio University
Hiyoshi 3–14–1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223–8522 Japan
E-mail: †bluemountain@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

Abstract　In rule-based email delivery system RMX, we can use some operators in an e-mail address. The '-' was defined as a range of the integer type, and the versatility was low. In this article, we hava introduced the rule that is polymorphic and raised versatility of the delivery rule definition.　In addition, we make use of a characteristic of RMX that distinguish user information automatic from a database on the basis of rules and introduced a function to choose a sentence suitable for a user specified by a delivery rule. Also, a function to insert the fixed form sentence prepared on a server.

Key words　RMX, email

1. はじめに

RMX(Rule-based e-Mail Exchange system) はユーザが設定したルールとメールアドレスを基に，データベースのアクセスを行い，得られたユーザ集合に対してメールを配信するメール転送エージェントである. しかし，先行研究では，ルールに与えられる変数は１つのみであり，２つ以上の変数を持ったクエリをルールとして設定することはできなかった. そこで，本論文では演算子の１つである'-' を使い，２つ以上の変数を持つクエリの設定を可能にした. 先行研究で'-' は与えられた２値の範囲のみを指定できるものであった. そこで，ポリモルフィズムの考え方を導入し，２つ以上の値を指定できるようにした. これにより，複雑なクエリの作成が可能となった.

また，本論文では，タグ形式を用いてサーバ上でメール本文を自動的に編集する機能を提案する.RMX というアプリケーションの特徴として，ルールとメールアドレスに基づいて自動的に対象であるユーザ集合を決定する. それを応用し，メール本文にも，対象のユーザとそうでないユーザを文章ごとに判断することで，受信ユーザごとに自分に該当する内容のみが載ったメールを受け取れるような機能が可能となる. ここで重要なのは，RMX のアプリケーション上の機能なので，この操作が全てサーバ上で行われるということである. よって，ユーザがメール本文に少し手を加えることによって，自動的にユーザごとに適したメールが作成される.　本論文では，以下の２つの機能を提案する. １つ目は，タグ形式で指定された配送ルールに従い，対象のユーザにのみ内容を載せる機能である Variable Matching. ２つ目は，あらかじめ用意しておいた定型文をタグ形式を用いて指定することにより，指定された定型文を挿入す

る Insert Text という機能である．

以下，本校の構成を示す．まず 2 章で RMX の概要と，ポリモルフィックルールの導入について述べる．3 章でメール本文編集機能について述べ，4 章でメール本文編集機能を用いた評価を行い，5 章で結論，6 章で今後の課題について述べる．

2. RMX

Rule-based e-Mail eXchange(RMX) は遠山研究室が提案している電子メール配信方式である．一般的にメール配信に利用されているメーリングリスト (ML) ではメーリングリストを代表するアドレスにメールが送信されるとメーリングリストに所属するメンバー全員に対してメールが配信される．

$< ML$ アドレス $>:=< ML$ 名 $> @ <$ ドメイン $>$

メーリングリストではメンバーのメールアドレスの更新など管理者の作業負担が必要となる．例えば大学でのメーリングリストの利用を考える．学生がメールアドレスを変更した場合にはクラス, 研究室, プロジェクトなど所属する全てのメーリングリストでアドレスの変更を行わなければならない．

それに対して RMX では下記のような記述により複数の送信先を指定する．

$< RMX$ のメール配信先指定 $>:=$

$<$ パラメータ $> @ <$ 配送ルール名 $> . <$ ドメイン $>$

RMX のメールアドレスは以上のように配送範囲記述部分とドメイン部分が”.”の記号で区分されている．配送範囲記述部は一つ以上のパラメータの組み合わせで構成され, @記号によってパラメータと配送ルール名に分けられる．RMX はこのような配送範囲記述を受け取る．そして, 指定された配送ルールとそのパラメータに基づきデータベースに問い合わせを行い実際の送信先アドレスを得る．最終的に得られたメールアドレスに基づき配送が行われる．図 1

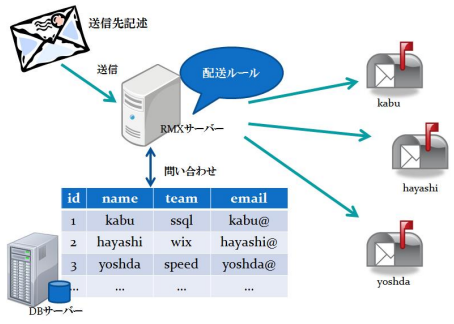

図 1　RMX におけるメール配信の流れ

RMX ではメール配信に必要な情報は全て利用組織のデータベース中で管理されている．そのため, 従来のメーリングリストのように一つ一つ登録アドレスを更新する必要はない．また, 送信者は配送ルールの記述により柔軟に送信範囲を指定できるため, メーリングリストのようにいくつものグループを用意する必要はない．例えば大学でメーリングリストを作成する場合を想定すると, 研究室, 学年, 学科などの連絡を多く取るであろうグループ毎に，メーリングリストを用意しなければならない．

2.1　配送ルール

配送ルールとは配送範囲記述とそれに基づき送信先メールアドレスを得るクエリを関連付けるルールである．配送ルールは以下のように定義する．

配送ルール名
Type:パラメータの型
query: 送信先メールアドレスを得るためのクエリ

query 部分は SQL によって記述される．RMX は記述された配送ルール名に対応するクエリにパラメータを挿入し, 問い合わせを行うことで送信先メールアドレスの集合を得る．このような配送ルールを用いることにより, 利用者は簡潔な記述で配送範囲を指定することができる．以下に配送ルール定義の例を示す．

grade
gradeType= integer
grade[1]= select s.address from student s
where s.grade = $1 ;

上記の例では学年を integer 型で受け取り, それに基づいてメール配信を行う grade ルールを定義している．メールの宛先が 4@grade.example.edu の場合，図 2 のように query 部分で利用者のメールアドレスと学年が格納されている表 student から, 学年が 4 年の学生のメールアドレスを得るクエリを記述している．

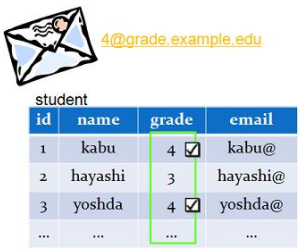

図 2　grade ルールと配送例

2.2　複数の配送ルールの組み合わせ

ここでは，複数の配送ルールを組み合わせてメールを送る際に，RMX で使用可能な演算子について説明する．演算子を Envelope-To フィールドの配送ルールに使用することにより, ユーザに対してより詳細な配送範囲の指定を可能とする．図 3 では，演算子の詳細を表にしている．

2.2.1　積　集　合

Syntax: $< par_1 > . \ \ldots \ . < par_n > @ < name_1 > . \ \ldots \ . < name_n > . < domain >$

Semantics: $name_1\ (par_1) \ \cap \ \ldots \ \cap \ name_n\ (par_n)$

演算子	ターゲット	意味	使用場所	優先順位
.	ルール	積集合	○@○	低 ↑
+	パラメーター	和集合	○@×	
-		多相	○@×	↓ 高

図 3　RMX の演算子

“.” は，Envelope-To フィールドに複数の配送ルールを使用したい時に用いられる演算子である．指定された複数の配送ルールの結果の積集合を取り，最終的に得られた結果に対して配送を行う．この演算子を使用するときは，“.” で区分されたパラメータと配送ルール名の数を一致させ，@を挟んでパラメータとそれに対応する配送ルール名を順番に並べる．

例：

toyama.1@lab.grade.example.edu

2.2.2　和　集　合

Syntax: $< par_1 > + \ \ldots \ + < par_n > @ < name_1 > . < domain >$

Semantics: $name_1\ (par_1) \ \cup \ \ldots \ \cup \ name_1\ (par_n)$

“+” は，ある配送ルールに対して論理和を得たい複数のパラメータを指定するときに用いられる．与えられた複数のパラメータをそれぞれ配送ルールのクエリに代入し，得られた結果の和集合を取る．そして，最終的に得られた結果に対して配送を行う．この演算子は，パラメータにのみ有効で，複数の配送ルールに対して “+” を適用するということはできない．

例：

3+4@grade.example.edu

2.2.3　多相 (ポリモルフィック)

Syntax: $< par_1 > - \ \ldots \ - < par_n > @ < name_1 > . < domain >$

Semantics: $name_1\ (par_1,\ \ldots,\ par_n)$

“-” は，ポリモルフィックな考え方を導入した演算子である．2.3 節において詳述するが，“-” によって与えられたパラメータの数により配送ルールから呼び出すクエリが異なる．

例：

aoyama-yosuke@name.example.edu

以上の 3 つの演算子は組み合わせて使用することもでき，ユーザが配送範囲を指定する際の手助けを行う．

例：

aoyama-yosuke+kabuki.toyama@name.lab.example.edu

2.3　RMX におけるポリモルフィックルールの導入

RMX の先行研究においては，演算子 “-” は整数型の範囲を指定すると定義されていた．これは，あまりにも汎用性が低く応用がきかないため，本論文でポリモルフィックルールを導入する．ポリモルフィックとは，多相的という意味であるが，本論文では，与えられた引数の数によって異なる動作を選択するという意味で使用している．本論文で新しく指定する rule の記述の例を以下に示す．

```
name
nameType: string
name[0] = select person.email from person
name[1] = select person.email from person
    where person.lastname=$1
name[2] = select person.email from person
    where person.lastname=$1 and person.firstname=$2
name[3] = select person.email from person
    where person.lastname=$1 and person.middlename=$2
          and person.firstname=$3
```

以上のひとまとまりを，name ルールとして宣言する．name の後に続く角カッコ内の数値が Envelope-To フィールドで与えられるパラメータの数と対応している．パラメータを複数指定する際は，引数を “-” で分けて記述する．“-” を 1 つ使い，2 つの引数を与えた場合は，name[2] が呼び出され，“-” を 2 つ使い，3 つの引数を与えた場合は name[3] が呼び出される．“-” で区切られた引数を左から$1,$2,$3…とし，クエリの対応する場所に代入する．name[0] というのは，引数が 1 つもなかった時のクエリを指すが，Envelope-To フィールド上で，@の左辺に値がないとエラーが起きてしまう．それを防ぐために，“*” または，“all” という定数を用いることで，name[0] を呼び出す．このように，引数の数を任意に定めることを可能にし，設定できるクエリの自由度を増すことで，複雑なクエリの生成が可能となり汎用性を増した．

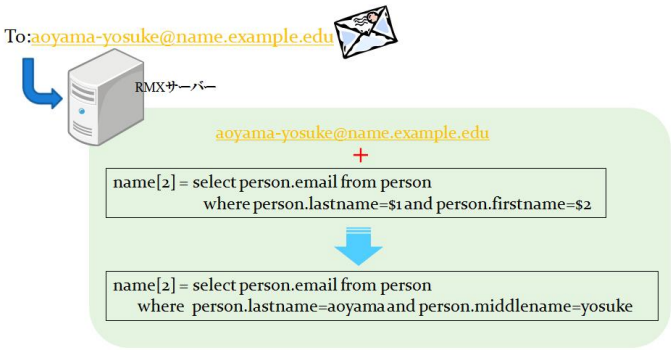

図 4　name ルールにおけるポリモルフィックルール処理

3.　メール本文編集機能

RMX は，ユーザが指定したルールに基づき，データベース

アクセスを行い，対象となるユーザを見つけるという作業をサーバ上で自動で行う．この特徴を生かし，本論文ではメール本文に記載されたタグ情報をもとに，サーバ上で自動的に受け取り手のユーザに適した文章を作成する機能を提案する．本論文で提案するのは，Variable Matching と Insert Text の 2 つの機能である．

3.1 Variable Matching

Variable Matching では，if タグを用いて文章ごとに配送ルールを適用する．

$$< if \text{ 配送ルール名} = \text{パラメータ} > contents < /if >$$

3.1.1 機 能 概 要

図 5 で示すように，文章を if タグで囲うことにより，指定されたユーザにのみ囲われた文章を抽出する．この作業はサーバ上で行われるため，送り手のユーザがタグを用いたメール本文を作成し，送信をすると，自動で受け取り手のユーザに適した文章だけが抽出されたメールが作成され送られる．if タグ内で使用されるルール名は 2 章で定義したものと同一である．すなわち，Envelope-To フィールドで使われる配送ルールと同一のものを使用する．2 章で説明された演算子としては，“-” と “+” が使用可能である．“.” の意味である Intersection は，if タグを階層化させることにより同一の結果を得ることができる．ここで，“.” の定義を行わなかったのは，階層化による Intersection の実現のほうが応用性が高いと判断したためである．

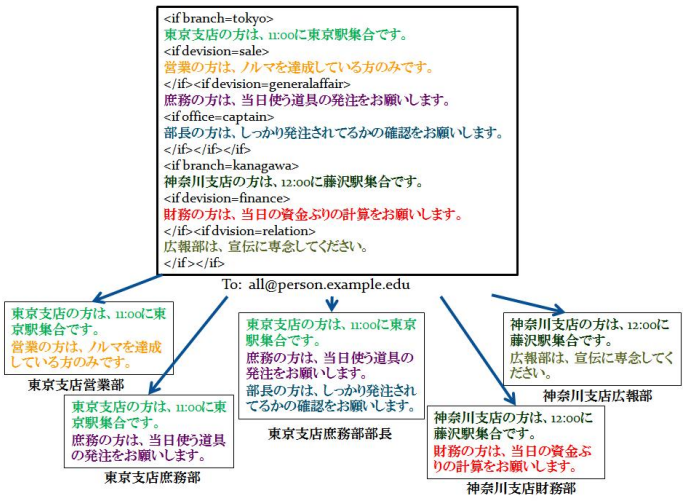

図 5　Variable Matching

3.1.2 エラー処理

もし，if タグを用いて作成したメールでエラーが起こり，受信者全員に何も処理がされていないメール本文 (if タグが残っているメール) が届いてしまった場合，他の受信者に送ってはいけなかった情報まで全員に送信されてしまう．このようなセキュリティー上の問題を解決するために，以下のケースを考える．

- if で指定される rule が定義されていなかった場合
- “-” で指定された変数の数に対応するクエリが存在しなかった場合
- ＜ if ＞と＜/if ＞の数が一致しない場合
- ＜ if ＞と＜/if ＞が正しくネストされていない場合

このようなエラーが発生した場合，それぞれのエラーに対し送信者に対してどのようなエラーが起きたかを記したメールを送信する．これにより送信者がエラーに気付き，再度正しいメールを送るように促す．

3.2 Insert Text

insert タグを用いてあらかじめ作成しておいたテキストファイルを本文に挿入する．

$$< insert \quad text \text{ 名} >$$

3.2.1 機 能 概 要

図 6 のように insert タグで指定されたテキストを本文に挿入する．テキストは”./insert”に格納されているテキストファイルを参照し，ファイル名を指定して呼び出す．現在の設定では，RMX 実行環境の相対パスによりファイルのアドレスを参照しているが，これでは汎用性が低いため，テキストをデータベースに格納し，管理する方法を検討中である．

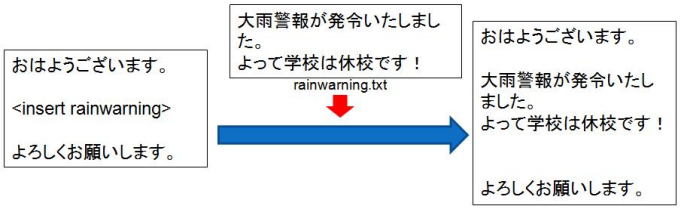

図 6　Insert Text

3.2.2 エラー処理

Insert Text でも，エラーが発生し，処理されずにメールが送信されてしまうと，相手に伝えたいテキストが挿入されずに送信されてしまう．このエラーを防ぐために指定されたテキストファイルが存在しなかった時に，その旨載せたメールを送信者に対して送信するという処理を行う．

4. 評　　価

本研究では，Variable Matching における評価を行う．

吹奏楽サークルにおいて，送信側と受信側に分けて評価実験を行った．送信側においては，一般の学生にメールを作成してもらい Variable Matching を使用する場合と使用しない場合での時間差を評価する．受信側においては，受け取った内容により，読む時間，どのくらい読んだかにどのような変化があるかについて評価を行う．

また，仮想組織内でメール作成を行う際，メール作成にかかる作業時間を任意に設定し，Variable Matching を使用した場合と使用しない場合での作業時間の試算を行い，評価する．

4.1 吹奏楽サークルにおける評価

4.1.1 送信側の評価実験

送信側の評価実験として被験者５人に対し，同じ内容のメールを Variable Matching を使用した場合と，曲ごとにメールを作成してもらう場合の２通りにおいてメールを作成してもらった．図 7 は，Variable Matching を使用した場合の被験者に作成してもらった文章である．まずこの文章のメールを作成してもらい，翌日にこの if 文ごとに配送ルールを分けたメールを作成してもらった．今回被験者には，簡単な配送ルールの説明と if タグの説明のみをした．

<if track= jubily>
演奏会のオープンにングです。冷静に、でも気持ちは前回でいきましょう。練習 でやってきたことをいかにしっかり出せるかが勝負です。がんばりましょう。 何といっても最初のファンファーレが命。みんなの気持ちをいきなりかっさらっ ていけるような演奏にしたいです。それでいて、軽くあること。1部1としてあ くまでさわやかさを忘れない。曲の性質も相まって、常に全体を、長いフレーズ を考えておらかに吹きましょう。
</if>
<if track=umio>
41、クレッシェンドを波のように 音を鳥羽臼為にはどうすればいいか、今一度考えてください。強く吹けば いいってわけじゃない。行きの流れをもっとまっすぐ太く。全ての音を鳴らそう。使える音を目指そう。たくさん息を吸って、まだまだ吸える。後悔だけはしない。
</if>
<if track=dairoku>
その1：一楽章3小節目を大切に その2：今アメ言われたことを確認すること その3：2楽章で寝ないこと！！（笑）→二楽章で集中力を切らさないこと！！ 頭はっきり、音丁寧に、後押ししない。 基本的なことができていれば、きっとあったかいハーモニーを楽しめるはず！こ のメンバー8人だからこそできる音楽をやりましょう。
</if>
<if track= slav>
民族色が濃い曲だからこそ、随所に現れる民族性を意識する。戦争と絡 めtあ音楽だからこそ、情熱を忘れずに。ベルキスより覚えやすいメロディだか ら、お客さんに印象づけるように。他の楽器との掛け合いが多いから、そこを最 後までわすれず。 「スラブ乗れればいいです」ってノリ曲希望を出すくらい、私はこの曲が好きです。出だしの重々しい感じも、途中の透明感あるメロディーも、ラストの勝利を 謳う華やかさも。目の前に津田先生がいる威圧感は半端ないですが、精一杯がん ばります！
</if>
<if track= siba>
最後当てます！以上。常に頭は冷静に。綿密に準備した音で、熱く聞かせましょう。私のこれまでのウ インド生活全部、最後のページにかけます！ 特にTpは4楽章の最後で、1つ1つの音の頭がすご く目立つので、今の勢いを残しつつ、頭が突出しないように頑張ってください。 Corは、お客様に届くような音を出し、かつ木管勢を超す指捌きをしま しょう！ 他のパートを耳をダンボにして聴きましょう！！ 休みの部分を最大限楽しみましょう！！演奏している舞台に一緒に乗ってるので すから、楽しまなきゃ損です！これはどの曲にも通ずる事ですね。
</if>

図 7　被験者に作成してもらったメール本文

4.1.2 受信側の評価実験

受信側の評価実験として，吹奏楽サークルに所属する 14 人にメールを読んでもらう実験を行った．このサークルでは，演奏会で演奏する５曲のうち１人３曲ずつ演奏する．そこで，合奏後にその日の注意が載ったメールを以下の３種類のメールとして送信した．

- 自分の演奏する曲の注意のみが載ったメール
- 全部の曲の注意が載ったメール
- 曲ごとにメールを分けた場合

この３種類において，

- メールを読むのにどれくらいの時間がかかったか
- メールを全部読んだか．

の２つの観点で評価してもらった．

4.1.3 送信側の評価

図 8 に被験者がメールを作成するのにかかった時間を示す．メールを作成するのにかかった時間を平均すると Variable Matching を使用した場合が 9.1 分で，曲ごとにメールを作成した場合が 12.1 分であった．被験者で多かったのは，Variable Matching を使用した場合は 1 通のメールなので，休憩を入れずに全部書ききったが，曲ごとにメールを作成する場合だと計５通のメールを作成しなければならず，１つメールないしは，２つメールを作成したら休憩を挟む人がほとんどであった．この結果平均して 33 %の作成時間の増加に繋がった．

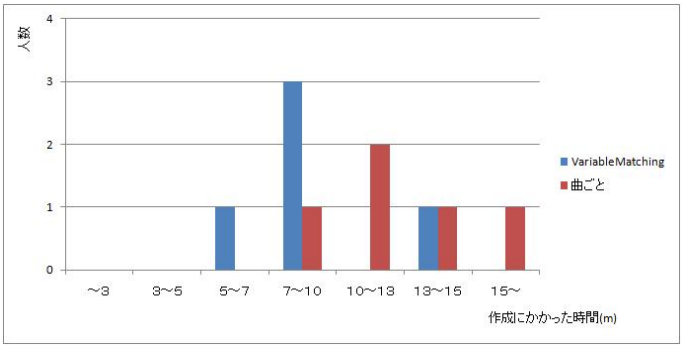

図 8　メール作成時間の割合

4.1.4 受信側の評価

図 9 に被験者がメールを読むのにかかった時間を示す．メールを読むのにかかった時間の平均は，曲ごとにメールが届く場合は 4.7 分，自分の曲のみが載ったメールの場合は 4.9 分，全曲載ったメールの場合は 5.1 分であった．また，ちゃんとメールを読んだかどうかに関しては，曲ごとにメールが届く場合は１人メールを読んでない被験者がおり，全曲載ったメールの場合は，4 人の被験者がメールを読んでいなかった．自分の曲のみが載ったメールの場合は全ての人がちゃんとメールを読んでいた．曲ごとにメールが届く場合のメールを読んでいない人１人を除いた平均時間は 5.0 分で，自分の曲のみが載ったメールの場合よりも若干だが長くなった．しかし，曲ごとにメールが別れているほうが見やすいという意見や，たくさんのメールがくるのは煩わしいという意見もあり，曲ごとにメールを分けた方がいいか，1 通にまとめた方がいいかは，ユーザによってとらえ方が違った．

以上の結果を踏まえると，ユーザに該当する内容のみが載ったメールを作成することは，メールをちゃんと読むことをうながし，メールを読む時間も短縮できるといえる．

送信側の評価と受信側の評価を併せて考えると，メールの作成時間は Variable Matching を使用した方が作業時間は軽減され，メールの購読率が上がり，読むのにかかる時間は減少したので，総合的に考えると Variable Matching は有用であるといえる．

4.2 仮想組織における試算

ここでは，任意に設定したメール作成する際の作業時間において Variable Matching を使用する場合と使用しない場合での作業時間の試算を行う．任意に設定した作業時間を表 1 に示す．

試算として，２つの配送ルールを使用してメール作成する場合と，３つの配送ルールを使用して作成する場合を考える．例えば，仮想組織において部署の配送ルールと支店の配送ルール

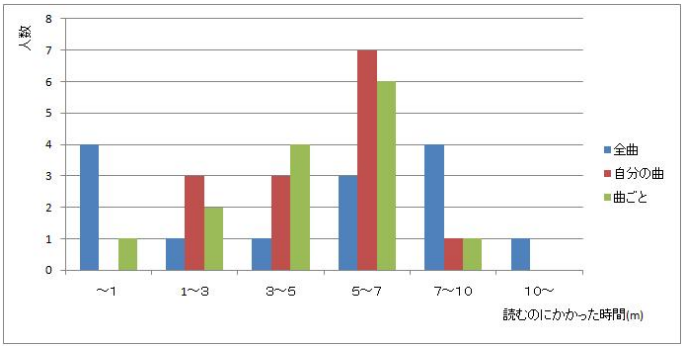

図 9　メールを読むのにかかった時間の割合

作業	所要時間
新規メール作成 メールアドレス入力	30s/1 通
if タグの入力	10s/1 個
本文作成	140s/ 1 値
メール送信時間	10s/1 通

表 1　メール作成に必要な各作業の所要時間

の 2 つを使用する場合と，他に役職という 3 つ目の配送ルールを使用してメール作成する場合がそれぞれに対応する．

本研究では，以下の 3 種類のメール作成方法についてそれぞれの所要時間を試算する．

- Variable Matching
- combination
- 値ごと

1 つ目は Variable Matching を用いてメール作成した場合である．つまり，図 5 中の上の文章のことである．2 つ目の combination は，図 5 中上の文章のような Variable Matching を階層的に用いて作成されたメール文章を図 10 の 5 つの文章のように，それぞれの階層の combination ごとにアドレスを作成し，メールを作成した場合である．3 つ目は，図 11 のように，Variable Matching を階層的に用いていない場合は，それぞれの値ごとにメールを作成することで，同様の結果を得ることができる．

図 5 と，図 11 を見てわかる通り，Variable Matching を階層的に用いているかいないかで，if タグの数が変わってくる．combination と比較する，if タグを階層的に使用しているメール本文に使用される if タグの数は，2 つの配送ルールを用いた場合，それぞれの配送ルールで指定する値の数を i,j(i が上位階層の配送ルールで指定される値の数) とすると

$$if\ タグの数 = i + i \times j$$

で求めることができる．3 つの配送ルールを用いた場合，それぞれの配送ルールで指定する値の数を i,j,k(i が最上位，k が最下位の配送ルールで指定される値の数) とすると

$$if\ タグの数 = i + i \times j + i \times j \times k$$

となる．

値ごとにメールを作成する場合と比較する，if タグを階層的に使用しないメール本文に使用される if タグの数は，2 つの配送ルールを用いた場合，それぞれの配送ルールで指定する値の数を i,j(i が上位階層の配送ルールで指定される値の数) とすると

$$if\ タグの数 = i + j$$

となる．3 つの配送ルールを用いた場合も純粋に 3 つの配送ルールでそれぞれ指定される値の数の和が if タグの数となる．以上のことを踏まえ，combination と値ごとにメールを作成する場合のそれぞれにおいて別々に Variable Matching との比較を行う．

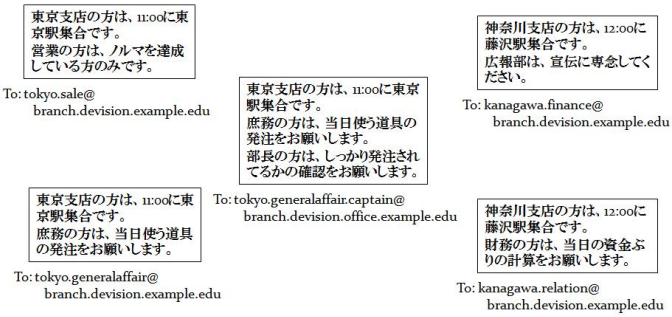

図 10　combination の例

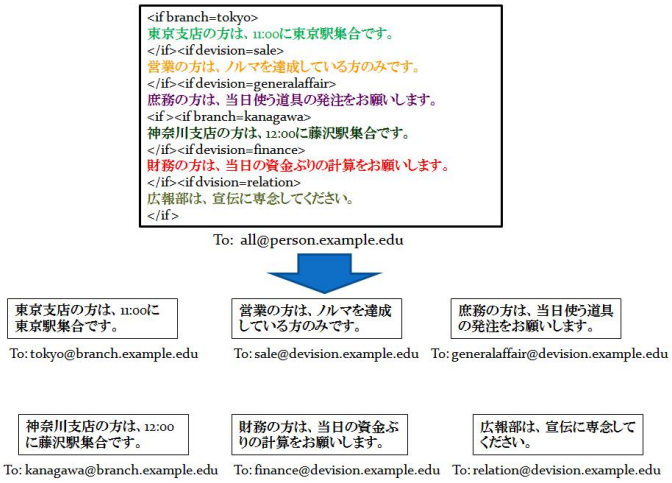

図 11　値ごとにメールを作成する場合の例

4. 2. 1　2 つの配送ルールを使用した場合の評価

図 12 は配送ルールを 2 つ用いた場合の Variable Matching と combination の作業時間の比較である．グラフでは見てとりにくいが，2 つの配送ルールがともに 1 つの値しか指定しない時に，Variable Matching よりもわずかに combination の方が作業時間が短い．指定する値が少しでも増えると Variable Matching のほうが combination よりも作業時間が短くなる．これは，両方の配送ルールで 1 つの値しか指定しない場合は，

Variable Matching，combination ともに作成するメールは１通である．しかし，Variable Matching は本文に if タグを用いて配送ルールを指定しなければならないため，その分少し作業時間が長くなる．他の場合では，combination のほうがメールを複数作成しなければならない分，急激に作業時間の差が広がっていく．

図 13 は，Variable Matching と値ごとにメールを作成した場合においての，作業時間差を示している．図 13 をみると常に Variable Matching の方が作業時間が短い．これは，Variable Matching と値ごとにメールを作成した場合において，前者の if タグの数と後者のメールを作成する数が同じになるので，この２つの作業時間の差からきている．よって，急激に差が開くことはなくとも少しずつ差が開いていく．

２つの配送ルールを用いた場合は，両方ともに１つの値しか指定しない場合を除けば，Variable Matching を使用したほうがメールの作成時間の短縮ができる．

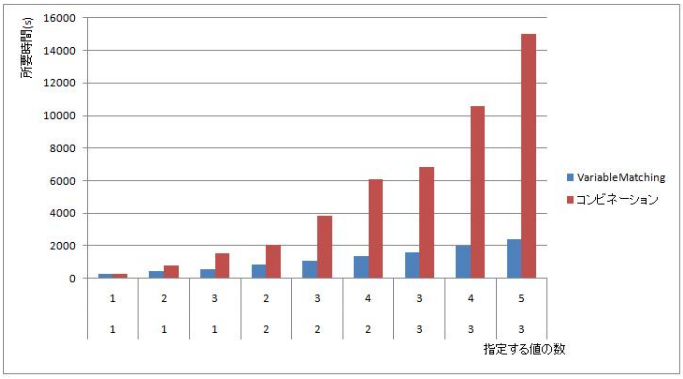

図 12　配送ルール２つの場合の Variable Matching と combination の作業時間差

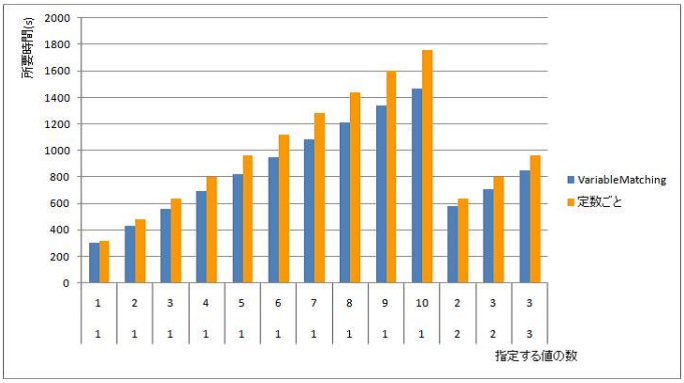

図 13　配送ルール２つの場合の Variable Matching と値ごとの作業時間差

4.2.2　３つの配送ルールを使用した場合の評価

図 14 は配送ルールを 3 つ用いた場合の Variable Matching と combination の作業時間の比較である．ここでも先ほどと同じようにあまり差は見えないが，３つの配送ルールとも１つの値しか指定しなかったとき，Variable Matching よりも combination のほうが作業時間が短い．先ほども述べたように if タグを用いる必要があるかどうかでこの差が生まれている．２つの配送ルールを使用した場合よりも作業時間が急激に増えていくのは，３つの配送ルールを用いることで combination の数も急激に増えるため，このような結果となった．　図 15 に Variable Matching と値ごとの作業時間の差を示す.Variable Matching を用いた場合と値ごとにメールを作成した場合との比較では，どのような条件でも Variable Matching のほうがいくらか作業時間が短いが，大きな差は見てとれない．このグラフを見ても分かる通り，常に Variable Matching の方が作業時間が短く済んでいて，少しずつであるが，差は開いているが大きな違いは出ていない．これは，Variable Matching における if タグの数と値ごとにメールを作成したにおけるメール作成数が同じになり，この２つの要素の差が少しずつ増えていくからである．しかし，この２つのメール作成方法で大きく違うのは，Variable Matching を用いた方は，自分に該当する文章だけ抽出され１通のメールとして届くが，値ごとにメールを作成した場合は自分に該当する分だけのメールが届く．すなわち Variable Matching を用いた方は，受信者は１通のメールを読むだけで済むが，値ごとにメールが作成された場合は，複数のメールを読むことになる．

３つの配送ルールを用いた場合でも，両方ともに１つの値しか指定しない場合を除けば，Variable Matching を使用したほうがメールの作成時間の短縮ができる．

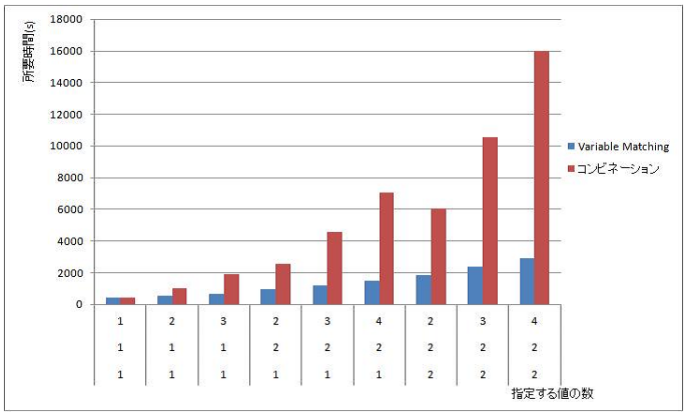

図 14　配送ルール３つの場合の Variable Matching と combination の作業時間差

今回の評価では，combination と Variable Matching の差は大きくでたが，値ごとにメールを作成した場合と Variable Matching を使用した場合では，小さかった．しかし，ここで注意するのは Variable Matching を使用した場合は，受信するメールは，自分が該当する内容が載った１通のみであり，値ごとにメールを作成した場合は，自分が該当する値の数だけのメールを受信するということである．

5.　結　　論

本研究では，RMX におけるポリモルフィックルールの導入

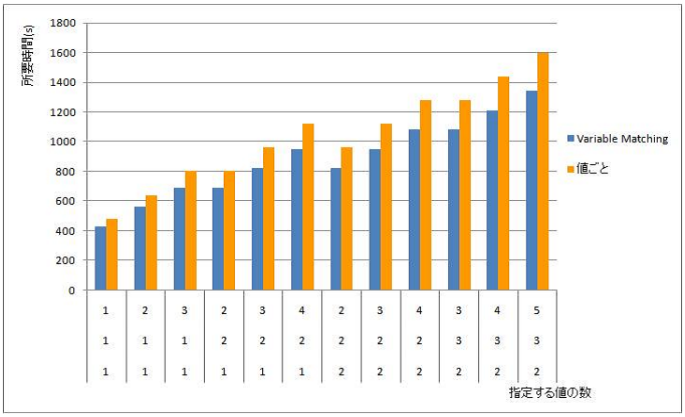

図 15　配送ルール 3 つの場合の Variable Matching と値ごとの作業時間差

およびメール本文編集機能の提案を行った．

ポリモルフィックルールの導入では，従来の RMX において，整数型の範囲しか指定できないという汎用性の低かった演算子 “-” を，ポリモルフィズムの考え方を導入し再定義を行うことで汎用性を高めた．これにより，ユーザが配送ルールの指定をする際に柔軟なクエリを作成する支援を行う．

メール本文編集機能においては，2 つの機能 ‘Variable Matching’ と ‘Insert Text’ の提案を行った．この機能は，RMX の特色を生かし，サーバ上で配信先のユーザに適したメール本文の内容を抽出したり，あらかじめ用意されている定型文を挿入する機能である．これにより，受け取り手のユーザがほしい情報だけの載ったメールを容易に作成することが可能となった．また，事務的な形式の決まった文章をあらかじめ作成しておくことで，ユーザがメールを作成する際の時間短縮に繋がる．

Variable Matching を用いた評価では，送信側では，メールを作成する時間を短縮し，受信側では，メールの購読率の向上，メールを読むのにかかる時間の短縮などの結果が得られた．

6.　今後の課題

6.1　ドメイン指定機能

現存の RMX では，複数の組織への対応ができておらず，1 つの組織に対してのみの対応で，アクセスするデータベースも一つのみの実装となっている．1 つの RMX を複数の組織が利用しようとしたとき，それぞれの配送ルール名がユニークでなくてはならず，同じ配送ルール名を使用することができない．しかし，実際は組織の中で使われる配送ルール名は類似することが多い．そこでドメイン指定機能の導入を検討中である．アドレスのドメイン部に組織ごとにエイリアスを設け，そのエイリアスを基に接続するデータベースや，配送ルールの指定を行うシステムを考えている．これに伴い，現在は配送ルールをプロパティファイルで管理しているが，これをデータベースで管理し，そこから配送ルールを持ってくる手法を検討中である．

6.2　ユーザインターフェース

RMX の弱点としてユーザインターフェースが挙げられる．現在は配送ルールの指定などを直接プロパティファイルを編集することで行っている．これでは，他の人が作成した配送ルールなども自由に編集できてしまい，セキュリティーの面でも問題がある．また，配送ルールで設定するクエリなどもユーザに一存しており，ミスがあった場合でも，そのままメールが配送されないままで終わってしまう．

また，メールを送った際に，誰に配送されたかを知る方法が現状ではない．送信されたメールが本来配信されるはずではなかった相手にも送信されている可能がある．メール本文編集機能においても，メール本文編集機能を用いて作成した本文が，誰にどのような文章で送信されているのかも現段階ではわからず，ユーザインタフェースとして確立する必要がある．このようなセキュリティ上の問題等も今後の課題として解決しなければならない．

文　　献

[1] 高畑理, 藤沼健太郎, 石橋玲, 遠山元道. ”Magic Mirror Mailing：個人情報データベースを利用する柔軟なメイル配送システム”, 情報処理学会データベースシステム研究報告 Pages: 123-128 July 2001
[2] Kim Hanki, Sang-Gyu Shin, Motomichi Toyama. ”A Rule-Based Mailing System for an Organization”, International Workshop on INformation Processing over Evolving Networks, June 2006
[3] 原田哲志, 慎　祥揆, 遠山元道. “ RMX における電子メール送受信範囲管理方式の提案”, DBWS2007
[4] sendmail. http://www.sendmail.org/
[5] qmail. http://www.qmail.org/
[6] fml. http://www.fml.org/ 深町 賢一.
[7] Majordomo. http://www.greatcircle.com/majordomo/
[8] P. Mockapetris. *Domain names - concepts and facilities*. RFC1034, 1987.
[9] P. Resnick. *Internet Message Format*. RFC2822, 2001.
[10] David Wood. *Programming Internet Email*. August 1999.
[11] Thomas Karagiannis, Milan Vojnovic. *Behavioral Profiles for Advanced Email Features*. WWW 2009 MADRID!.