

# RMX 配送ルール管理ツールの開発

土居 靖<sup>†</sup> 遠山 元道<sup>†</sup>

<sup>†</sup> 慶應義塾大学理工学部情報工学科 〒 223-8522 横浜市港北区日吉 3-14-1

E-mail: <sup>†</sup>doi@db.ics.keio.ac.jp, <sup>††</sup>toyama@ics.keio.ac.jp

あらまし RMX(Rule-based e-Mail eXchange system) とはあらかじめ設定されたルールに基づき、自動的にメールを転送するメール転送システムである。このシステムの利用を開始するには、あらかじめメール配送のルールを定義した SQL クエリを記述する必要がある。本研究では、その際に、このシステムを運用する管理者の SQL の記述力やデータベースの知識の個人差に対応することができるツールを提案する。またこのツールでは、生成された配送ルールの合成やパラメータの変更をするという配送ルールの管理により、RMX 運用の際の配送ルールの表現力のサポートをして、送信者のメール送信の際の負担の軽減を実現する。

キーワード RMX, 電子メール, メーリングリスト

## 1. はじめに

Rule-based e-Mail eXchange(RMX) は管理者があらかじめ定義したルールに基づき、データベースから動的にメールを転送するメール転送システムである [1,2]. 従来のメールアドレスはアカウント名とドメイン名から構成されるのに対して、RMX では配送ルール、パラメータ、ドメイン名の 3 つから構成されている。これらの 3 種類の構成要素を組み合わせることによって、データベースからアドレスの集合を得ることができ、従来のメールと比べ広がりを持つ。

このシステムの利用を開始するには、あらかじめメール配送のルールを定義した SQL クエリを記述する必要がある。よってこのシステムを運用する管理者の SQL の記述力やデータベースの知識に個人差があると想定して、対応すべきであると考えられる。そこで本研究では、データベースの知識や実際のテーブルの属性などを十分に把握せずに配送ルールの半自動的に生成をするツールを作成した。またこのツールでは、送信者がアドレスを指定してメールを送信する際にアドレスの記述の間違いの可能性を軽減するために、実際にデータベースにアクセスして、アドレスが得られるかを確認するデモ送信の機能もある。さらに、すでに作成された配送ルールの合成やパラメータの削減などの機能もあり、RMX 運用の際の配送ルールの表現力のサポートを実現する。

本論文の構成は以下の通りである。まず 2 章で RMX の概要について述べる。次に 3 章で RMX の現状と課題について述べる。4 章では今回提案するアプリケーションについて、5 章では関連研究について述べ、6 章では評価について、7 章で結論と今後の課題について述べる。

## 2. Rule-based e-Mail eXchange

### 2.1 RMX とは

Rule-based e-Mail eXchange (RMX) は遠山研究室が提案している電子メールの配信方式である。RMX では管理者があらかじめルールを定義することにより、データベースからメール

アドレスを動的に取得し、複数の送信先にメールを自動的に転送する。

RMX ではアドレスの記述方法は関数形式と自然形式の 2 種類があり、これはそのうち関数形式の記述方式である。

<RMX のメール配信先指定 (関数形式)>:=

<配送ルール名>{<パラメータ>}@<サブドメイン>.<ドメイン>

RMX はこのように、配送範囲を記述するサブドメイン以前の部分と、ドメイン移行が “.” によって組み合わせられている。配送範囲の中でも @ 以後のサブドメインの部分は後述するルールベースの名前に相当する。また @ 以前の部分は、“{ }” の左側に示される配送ルールと、その “{ }” 中に記述されるパラメータという 2 つの部分に区別される。これらの 3 種類の構成要素を組み合わせることによって、ルールベース (後述) の選択、ルールの参照、データベース問い合わせを行い実際の送信先アドレスの集合を得て、メールの配送が行われる (図 1)。

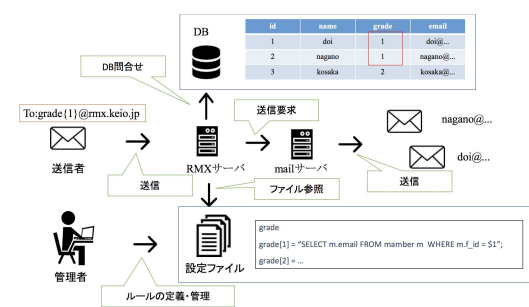


図 1 RMX におけるメール配信の流れ

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、メーリングリストのように、グループそれぞれに一つずつアドレスの登録を行う必要はない。もしユーザを追加する場合は、データベースのテーブルを一つ加える事で、メールの送信が可能になる。データベースを活用することで、メーリングリスト管理者の負担を削減し、柔軟な宛先指定を可能にする。

## 2.2 RMX ルールベース

RMX においてルールが記述されている場所をルールベースと呼ぶ。RMX ルールベースでは、データベースへの接続および使用するルールを管理する。RMX のメールアドレスでメールが送信されると、RMX システムは指定されたサブドメインのルールベースから配送ルールを取得する (図 2)。

現在の実装では、プロパティファイルを RMX ルールベースのサブドメインのテキスト表現として使用している。使用する設定ファイルは、“サブドメイン名.properties” とする。たとえば、keio.properties というファイルを作成し、サブドメインが keio のアドレスから電子メールを受信すると、keio.properties を参照して配信ルールを取得する。

それぞれのルールベースでは、以下の項目を用いて記述を行う。

```
DB_DRIVER = <データベースドライバ>
DB_URL = <接続するデータベースの URL >
DB_ID = <データベース上のユーザ ID >
DB_PASSWORD = <ユーザ ID に対応するパスワード>
DB_ENCODING = <データベースの文字コード>
DEBUG_USER = <デバッグコマンドが使用できるメールアドレス>
<使用するルールの設定>
```

上記のように記述することで、サブドメイン毎に接続するデータベースを選択できる (図 2)。また、ルールもサブドメイン毎に指定することが可能となる。

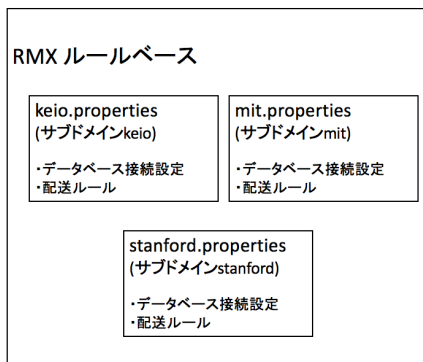


図 2 ルールベース構造

## 2.3 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のメールアドレスを取得す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

query: 送信先メールアドレスを得るためのクエリ

query は SQL によって記述する。配送ルール名に対応するクエリにパラメータを挿入し、データベース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取得する。このような配送

ルールを用いることで、ユーザは簡潔な記述で配送範囲を記述することができる。以下に配送ルールの定義の例を示す。

dept

```
dept[1]= select s.email from student s where s.dept = '$1';
```

上記の例では所属学科ごとにメールを送信することができる。dept ルールである。このルールに基づいてメールの送信を行う。メールの宛先が dept{ics}@demo.krmx.jp の場合、図 3 のように query の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テーブル student を参照し、そこから所属学科が ics の学生のメールアドレスを取得するクエリを記述している。

id	name	grade	lab	dept	email
1	doi	3	toyama	ics	doi@...
2	tatsu	3	saifo	sd	tatsu@...
3	nagano	4	toyama	ics	nagano@...
4	machida	3	kaneko	sd	machida@...
5	kosaka	4	saifo	ics	kosaka@...

図 3 dept ルールと配送例

## 2.4 配送ルールの組み合わせ

RMX では、複数の配送ルールを組み合わせることが可能である。複数の配送ルールを組み合わせることでメールを送る際、RMX で使用可能な演算子について説明する。To フィールドの配送ルールに、演算子を使用することによって、ユーザに対してより詳細な配送範囲の指定を可能にする。

### 2.4.1 積集合

Syntax: <name<sub>1</sub>>{<par<sub>1</sub>>}...<name<sub>n</sub>>{<par<sub>n</sub>>}@<subdomain>.<domain>

Semantics: name<sub>1</sub> (par<sub>1</sub>) ∩ ... ∩ name<sub>n</sub> (par<sub>n</sub>)

”.”は、論理積によって複数の配送ルール及びパラメータを指定する際に用いられる。各パラメータをそれぞれの配送ルールのクエリに代入し、得られた結果の積集合から配送を行う。よって上記の例では、学科が ics で研究室が遠山である学生に向けて、メールを配送することになる。

例:dept{ics}.lab{toyama}@demo.krmx.jp

### 2.4.2 和集合

Syntax: <name<sub>1</sub>>{<par<sub>1</sub>>}...<name<sub>n</sub>>{<par<sub>n</sub>>}@<subdomain>.<domain>

Semantics: name<sub>1</sub> (par<sub>1</sub>) ∪ ... ∪ name<sub>n</sub> (par<sub>n</sub>)

”+”は、論理和によって複数の配送ルール及びパラメータを指定する際に用いられる。各パラメータをそれぞれの配送ルールのクエリに代入し、得られた結果の和集合から配送を行う。論理和は、同じ配送ルールでも異なる配送ルールでも使用が可能である。

例:dept{ics}+grade{1}@demo.krmx.jp

よって上記の例では、学科が ics である学生または学年が 1 年である学生に向けて、メールを配送することになる。

また、配送ルール間の積集合をと“.”と和集合をと“+”が同時に使用された場合、積集合をとる“.”の方が優先順位が高いものとする。

例: name{doi}+grade{4}.dept{ics}@demo.krmx.jp

例えばこのようなメールアドレスの場合、grade と dept の積集合をとった後、その結果と name との和集合をとる。つまりこの例の場合は、学年が 4 年の情報工学科 (ics) に所属する学生、または doi という名前の学生にメールを送信するという結果になる。

### 3. RMX の現状と提案

#### 3.1 RMX の運用

RMX の運用を開始し、最初にメールを送信するまでの手順を述べる。

- (1) RMX サーバのダウンロード
- (2) 環境設定ファイル (env.properties) の変更
- (3) プロパティファイル (.properties) の作成
- (4) 配送ルールの生成
- (5) RMX サーバの起動
- (6) メール送信

本論文では、すでにデータベースと RMX サーバは用意されている状態を想定して、論じていく。しかし、その他の手順に関してはメール送信を除いて、管理者が行わなければならない。そこで、これらの手順を一括して、一つのツールで行うことができれば、RMX の運用における管理者の負担が軽減できると考えられる。

#### 3.2 配送ルールクエリの記述

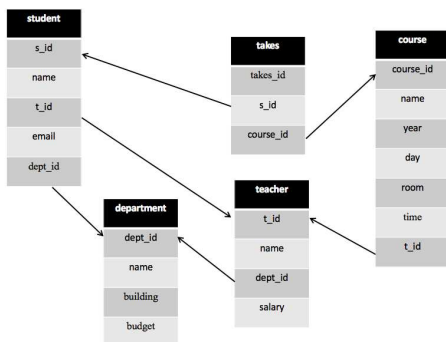


図 4 大学のスキーマの例

図 4 のようなスキーマに対して RMX で利用する配送ルールクエリを作成する場合を考える。例えば、ある学生 (student テーブル) が履修している授業 (course テーブル) の各属性に対する条件に該当する学生にメールを送信するためには、以下のような SQL クエリの記述が必要になる。

```
SELECT student.email FROM student
JOIN takes ON student.s_id = takes.s_id
JOIN course ON takes.course_id = course.course_id
WHERE course.(course の各属性) = $1;
```

このように、JOIN 句によってテーブルの外部キーに対してテーブルを結合し、さらに RMX の配送ルールとしてのクエリを記述するためには、データベース上の外部キーなどの参照関係の把握や、実際の SQL クエリの記述スキルなど、RMX の配送ルールの生成は管理者のデータベースの知識に依存するという面があり、実際にルールの生成をすることが困難な場合があると考えられる。そこで、十分な SQL クエリの記述スキルを持たず、また実際のテーブルの関係を把握していなくても、自動で配送ルールを生成することができれば、上述した課題の改善となると考えられる。

#### 3.3 RMX のメール送信の際の負担

管理者が生成した配送ルールを用いて、送信者がメールを送信する。例えば、

例 3 : grade{1}.adviser{ 安藤 }@keio.krmx.jp

という学年が 1 年生でかつ、担当教員が安藤の人にメールを送信するとする。しかし、この際、アドレスとして記述した配送ルールやパラメータに入力ミスがあり、メールを送信したつもりが、メール送信ができていないといった送信者に対する負担が発生する場合が考えられる。そこで実際にメールを送信する前に、そのアドレスが結果として得られるかを確認し、確認後にメールを送信することにより、上述した入力ミスによりメールが送信できていないといった可能性をできる限り軽減させる必要があると考えられる。それにより、送信者の負担も軽減できる。

#### 3.4 複数のパラメータを持つ配送ルールのメール送信

さらに複数のパラメータを持つ配送ルールに対しては、メールを送信する際、パラメータをそれぞれに入力する必要がある。その際に、使用頻度の高い組み合わせがあれば、パラメータがあらかじめ指定されている配送ルールを生成しておくことにより、第 3.3 節で挙げたような入力ミスによる送信者の負担は軽減することができる。例えば、

例 6 : gradefaculty{2-理工}@keio.krmx.jp

この際、送信者が学科が理工のルールを使用する頻度が高い場合、

'\$2' = '理工'

があらかじめ代入されている配送ルールを生成することにより、送信者のパラメータ入力といった負担が軽減できると考えられる。その際、新たな配送ルールを直接記述せずとも、既存の配送ルールから新たなルールを生成できれば、SQL 記述のスキルがない管理者であっても、生成することが可能となると考えられる。

## 4. 提案するツールについて

### 4.1 ツールの概要

RMX アプリケーション導入後の全体像を図5に示す。

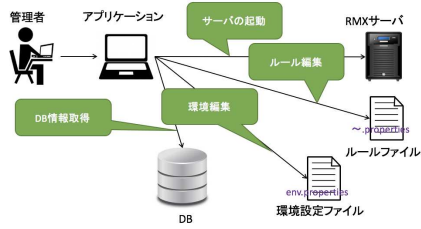


図5 RMX アプリケーション導入後の全体像

このツールを用いることにより、第3.1節で述べた手順(3)から(7)の作業を実行することができる。図5の通り、管理者はデータベースから情報の取得、環境設定、その他ルールファイルの編集、またRMXサーバ起動を行うことができる。

さらに、第3章で述べたような、配送ルールの自動生成やすでにある配送ルールの管理の機能を持たせることにより、このツールを利用する管理者のサポートをする。さらにはメール送信の際の負担の軽減も実現する。

### 4.2 各機能について

本研究で提案するツールの主な機能について説明する。

#### 4.2.1 配送ルールの自動生成機能

管理者はRMXの配送ルールをメール配送する際に使用するデータベースの情報が書かれたルールファイルから配送ルールを生成したいファイルを選択する。すると、データベースの情報を取得し、メールアドレスを値とする属性を持つテーブルから外部キーによる参照関係を持つテーブルを木構造で保持して、ツールの画面に表示する。

ここでの処理は、まず図4のスキーマに対して、図6のようにデータベース内の外部キーの参照関係を含めたテーブル情報を取得する。その際、mail属性をもつテーブルを保持し、そのテーブルから、参照関係のデータをもとにJOIN句の生成をする。管理者は生成したい情報を含む参照関係のもとにあるパスを表す行を選択することで配送ルールを自動生成できる。実際に選択したテーブルから木構造を元にmail属性をもつテーブルまで参照する。そこから参照したJOIN句をもとに図6のようにルールを生成する。

例えば、図7のように

student-teacher-department

の位置にあるdepartmentテーブルを選択すると、選択されたテーブルから木構造をたどって、JOIN句を参照する。この場合は、図6のテーブル情報から、図7のように、mail属性を持つstudentテーブルから

SELECT student.email FROM student

を生成、teacherテーブルとの参照関係から

constraint_name	table_name	column_name	foreign_table_name	foreign_column_name
teacher_dept_id_fkey	teacher	dept_id	department	dept_id
student_t_id_fkey	student	t_id	teacher	t_id
student_dept_id_fkey	student	dept_id	department	dept_id
course_t_id_fkey	course	t_id	teacher	t_id
takes_t_id_fkey	takes	t_id	teacher	t_id
takes_course_id_fkey	takes	course_id	course	course_id

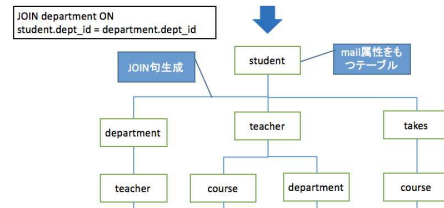


図6 テーブルの一括表示までの流れ

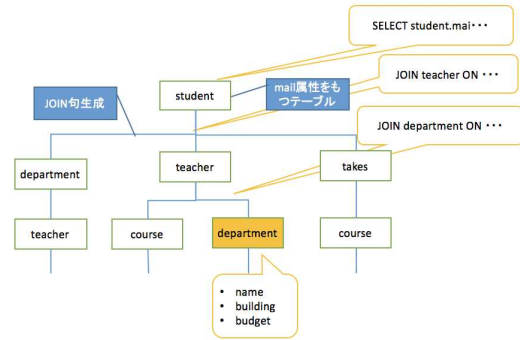


図7 配送ルール自動生成までの流れ

JOIN teacher ON student.t\_id = teacher.t\_id

が得られる。続いて、departmentテーブルとの参照関係から

JOIN department ON teacher.dept\_id = department.dept\_id

を得る。選択したテーブルまで、辿ると、最後に選択したテーブルの各属性に対するWHERE句

WHERE department.(属性) = '\$1'

を取得する。その後、取得された式を組み合わせることにより、配送ルールのクエリを生成することができる。

#### 4.2.2 ルールのクエリチェック

第3.3節で述べたように、アドレスとして記述した際の配送ルールやパラメータの入力ミスの可能性を軽減させるため、クエリチェックの機能を設ける。図8のように、使用する配送ルールを選択し、実際にパラメータを入力したクエリをチェックすることができます。これにより、取得できるアドレスが0件であるかテストをすることができ、またそのままメールを送ることができる。

例えば、図8のように、配送ルール“sudentyear”を選択し、さらにパラメータとなる値を入力する。すると、

SELECT mail FROM student WHERE year = \$1

に値が入力され、

SELECT mail FROM student WHERE year = 2017

データベースに接続し、結果が得られるかを確認することがで

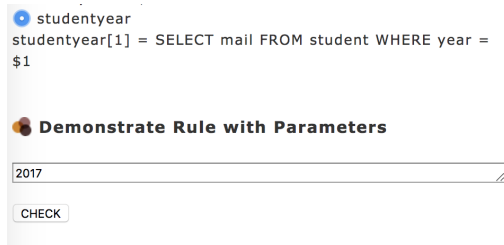


図 8 クエリチェックの画面

きる。この際、取得できるアドレスが 0 件であった場合、

There is no result.

と表示され、メールが送信できるか確認することができる。また、結果が得られた場合、ツールからメール送信することができる。

#### 4.2.3 配送ルールのパラメータ変更

配送ルールのパラメータ変更の機能について説明する。選択したルールに対するパラメータを指定することでパラメータの数を減少させた新たなルールを作成できる。

例えば、

$C[3] = \text{SELECT } \dots ' \$1 ' \dots ' \$2 ' \dots ' \$3 '$

というルールに対して、

'\$1'='AA'

と入力すると、

$AAC[2] = \text{SELECT } \dots ' AA ' \dots ' \$1 ' \dots ' \$2 '$

というパラメータ数の変更した新たな配送ルール名となる。

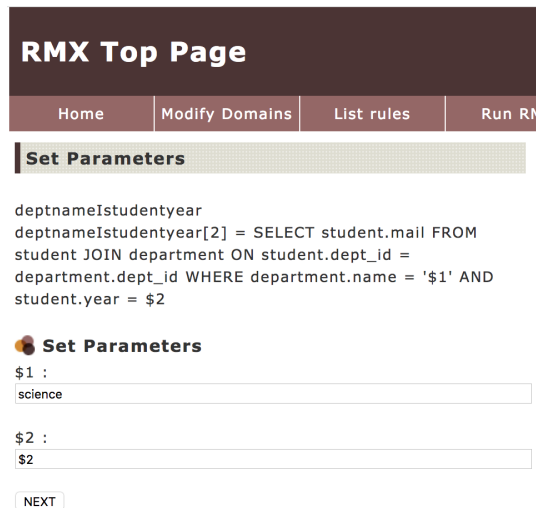


図 9 パラメータ変更画面の入力

図 9 のように配送ルールに対して、パラメータを入力すると、図 10 のような新たな配送ルールが生成することができる。

第 3.4 節で述べたように、新たな配送ルールの生成により、送信者の負担を軽減させた配送ルールの生成が可能となったと考えられる。

#### New Rule

```
sciencedeptnameIstudentyear
sciencedeptnameIstudentyear[1] = SELECT student.mail
FROM student JOIN department ON student.dept_id =
department.dept_id WHERE department.name = 'science'
AND student.year = $1
```

図 10 パラメータ変更画面の出力

#### 4.2.4 配送ルールの合成

既存の配送ルールの合成の機能について説明する。

RMX ではメール送信の際に複数のルールを組み合わせさせて送信することができる。その際に使用頻度の高いルールの組み合わせがあった場合、ツールを用いることにより、管理者があらかじめ既存の配送ルール同士を組み合わせさせた新たな配送ルールを生成することができる。

図 11 のように、配送ルールを選択していく。その際に、ツール内で“INTERSECT”か“UNION”か“EXCEPT”を選択することができる。選択した演算子が実際にクエリ同士を結合していく。最終的に決定した際の変数の数に合わせて、配送ルールのパラメータ数を変更される。

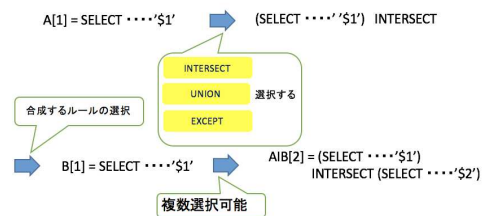


図 11 配送ルールの合成の流れ 1

例えば、以下のようなルールが存在するとする。

$A[1] = \text{SELECT } \dots ' \$1 '$

$B[1] = \text{SELECT } \dots ' \$1 '$

これらのルールからルール A を選択し、“INTERSECT”を選択する。すると、以下のようなクエリとなる。

$(\text{SELECT } \dots ' \$1 ') \text{ INTERSECT}$

さらにルール B を選択する。これらの手順を複数回繰り返すことも可能となっている。ルール A とルール B の積集合とすると、変数が 2 つであるため、

$AIB[2] = (\text{SELECT } \dots ' \$1 ') \text{ INTERSECT } (\text{SELECT } \dots ' \$2 ')$

というパラメータ数の変更した新たな配送ルール名となる。この機能により、パラメータを複数持つ配送ルールが生成することが可能となった。ちなみに 2 つの配送ルール名の結合子として“INTERSECT”の頭文字である“I”が使用されている。

## 5. 関連研究

本研究と同様にデータベースの知識が無い利用者向けのメーリングリストシステムとして挙動する既存のメール配信システムに blaynmail[3], Benchmark Email[4], Acces Mail[5] などがある。これらは企業がマーケティングメールを送信する手助けをするために開発されたシステムであり、これ以外にも多くのメール配信システムが存在する。これらは全て、企業がマーケティングメールを送信する手助けをするために開発されたシステムで、これ以外にも多くのメール配信システムが存在する。以下の表は RMX と 3 つのメール配信システムの機能を比較した表である。

	RMX	blaynmail	Benchmark Email	Acces Mail
グループの登録	X	O	O	O
既存の DB の利用	O	X	X	X
複数テーブルからの情報抽出	O	X	X	X
専用エディタ	X	O	O	O
差しこみコード数	-	∞	28 項目	7 項目

この表から RMX の大きな特徴として以下の 3 点があげられる。

- (1) グループごとの宛先の登録が不要
- (2) 既存のデータベースに組み込み可能
- (3) RDB を用いて広い範囲での宛先指定が可能

1 点目は、メールアドレスからその場で複数の送信先を決定するのでグループごとの宛先の登録が不要であるという点である。Benchmark Email では、送信するグループごとに登録が必要だが、RMX ではグループを個々で登録する必要はない。

2 点目は、既存のデータベースに組み込み可能であり、メール配信のために、改めて宛先の情報を登録する必要が無い点である。blaynmail ではメール送信のために宛先の登録が必要だが、RMX は既存のデータベースの使用が可能である。

3 点目は、SQL で記述できる範囲であれば、RMX では配送ルールを用いて、どのような組み合わせの宛先も抽出することができる点である。既存のメール配信システムでは、宛先はあらかじめグループごとで決められているか、同じテーブルの範囲内のみでの抽出しか行うことができない。

こういった点から、RMX は、既存のシステムと比べ、複数のテーブルをまたぐような広い範囲での宛先指定が可能であるといえる。これは既存のメール配信システムと大きく異なる点であると考えられる。

さらに近年の電子メールに関する研究を紹介する。Beng Heng らによる研究 [6] ではアドレスのリークによるスパムメールの被害を防ぐため、semi-private address という使い捨てのアドレスを提案し、それを利用するためのシステム SEAL を構築している。この SEAL システムでは受信者がコマンドを含むメールを送信することで自分の semi-private address に対して段階的に制限をかけることが出来る。Arne Tauber らによる研

究 [7] では重要な文書における機密性や信頼性を保証するために利用されるクローズドなシステム Certified mail system(CMS) において、組織間の CMS のギャップを埋めるためのフレームワークを提案している。Polina ZIberman らによる研究 [8] では、誤った受信者にメールが送信され、情報が流出するのを防ぐ手法を提案している。メールを送信する前に内容からキーワードを抽出し、過去のやり取りしたメールとの類似度を比較する。類似度が閾値に達した場合のみメールの送信が許されるもので過去にやり取りの無い送受信者間でも分類を適応できる点が特徴である。Changyan Chi らによる研究 [9] では、Dandelion という wiki をベースにした共同制作ツールを開発している。タスクの情報を管理者や複数の共作者が wiki と電子メールによって共有することを目的としたツールである。Dandelion ではあらかじめ用意されたテンプレートを用いてメールを作成する。管理者が新たなタスクについて情報を共有しようとした場合、テンプレートには全体的なメッセージ・登録するタスク・リマインダー設定を記述する欄が用意されている。これにより作成されたメールを Dandelion に送信すると、Dandelion は自動でタスクの内容を共作者達に送信し、同時にタスクの wiki ページを作成するという仕組みである。結果として、メールによってタスクの割当が出来る点や、自動で wiki ページ作成やメール送信がされることによって作業時間が減らせる点など、有用性の観点で優れていることが認められた。Jaclyn Wainer らによる研究 [10] では、電子メールのタイトルだけでなく読む電子メールか後回しにする電子メールかを判断する事が増えてきていることに着目し、どのような電子メールに対して人はすぐに読むや、後回しにするなどの判断しているかについて有用性と好奇心という尺度を用いて研究している。

また先行研究として、配送ルールの自動生成をするアプリケーションの紹介をする.[11] 利用者は RMX でメール配送する際に使用するデータベースの接続情報、生成されるファイル名を指定する。その後まずメール配送の際に必要なメールアドレスを保持したテーブルと、その属性を指定する。さらにそのテーブルを参照している、またそのテーブルが参照しているテーブルが表示され、利用者はどのようなテーブルを用いて配送ルールを生成するかを指定していく。図 9 はその際の様子である。

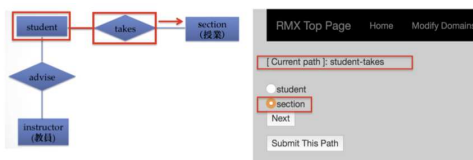


図 12 テーブル選択画面

このアプリケーションにより生成される配送ルールは、同じものであるが、しかしこのアプリケーションにより配送ルールを自動生成する場合、テーブルをひとつひとつ選択していくことで、参照関係をたどって行くため、あらかじめデータベースの ER 図を把握していなければならないという点がある。

また、すでにこのアプリケーションで、配送ルールの手動での記述は可能となっている。それ以外にも、配送ルールの削除やルール名変更なども可能である。

今回のツールは、その機能も取り入れることにより、ある程度 SQL 記述能力を持つ管理者にも対応することができる。

## 6. 評価と考察

### 6.1 比較する環境と条件

どの程度の複雑な配送ルールのクエリを表現できるか、その表現力について環境ごとに比較し、考察していく。

比較する対象はツール未使用、つまり通常のテキストエディタを利用して、クエリを記述するという環境、本研究のツールを利用するという環境、第 5 章で紹介したツールを利用するという環境の以上 3 つとする。これ以降、前者のツールをその性質から「一括表示型ツール」、後者のツールを「逐次指定型ツール」と称する。

- (1) ツール未使用
- (2) 一括表示型ツール
- (3) 逐次指定型ツール

また、その際に、以下のように管理者のデータベースの把握の有無、SQL の記述スキルの有無を条件に指定する。

- (A) 利用するデータベースの情報の把握あり + SQL 記述スキルあり
- (B) 利用するデータベースの情報の把握なし + SQL 記述スキルあり
- (C) 利用するデータベースの情報の把握あり + SQL 記述スキルなし
- (D) 利用するデータベースの情報の把握なし + SQL 記述スキルなし

### 6.2 RMX 配送ルールの表現力の例

本論文で評価すべき RMX 配送ルールの特徴とその例を以下に述べる。

- 利用するテーブルが一つ

```
name
name[1] = select mail from student where name = '$1'
```

- パラメータが一つ

```
adviser
adviser[1] = SELECT s.mail FROM student s, adviser
a WHERE s.adviser = a.id AND a.name = '$1'
```

- パラメータが複数

```
faculty
faculty[2] = SELECT s.mail FROM student s, faculty
f WHERE s.faculty = f.id AND f.name = '$1' OR
f.name = '$2'
```

- 関数や計算式を含む

```
grade
grade[1] = SELECT mail FROM student WHERE
2016 - year + 1 = $1
```

- 値の範囲指定

```
enter
enter[2] = SELECT mail FROM student WHERE year
>= $1 AND year <= $2
```

### 6.3 実験

#### 6.3.1 実験条件

第 6.1 節、第 6.2 節の条件のもと、さらにその他の条件を設定して、実際に配送ルールを生成できるか、実験を行なった。以下のように条件を揃える。

- SQL 記述スキルをもつ人 5 人の管理者を用意
- 管理者は RMX の概要を説明され、基本は把握している状態
- 第 6.2 節の特徴を持つ配送ルールをそれぞれ 5 件、こちらで指定したものを生成
- SQL 記述スキルなしの条件では、管理者は直接 SQL 記述はしないと設定
- データベースの情報なしのときには、管理者はテーブルの名称のみ把握

以上の条件をもとに、それぞれ全 25 件のうち、成功した件数から成功率 (百分率) を結果として示す。

#### 6.3.2 実験結果

第 6.3.1 節の実験の結果を以下の表にまとめる。

表 1 利用するデータベースの情報の把握あり + SQL 記述スキルありの実験結果

ルールの特徴	ツール未使用	一括表示型	逐次指定型
テーブルが一つ	100	100	100
パラメータが一つ	100	100	100
パラメータが複数	100	100	100
関数や計算式を含む	96	96	96
値の範囲指定	100	100	100

表 2 利用するデータベースの情報の把握なし + SQL 記述スキルありの実験結果

ルールの特徴	ツール未使用	一括表示型	逐次指定型
テーブルが一つ	40	100	100
パラメータが一つ	0	100	48
パラメータが複数	0	88	0
関数や計算式を含む	0	40	0
値の範囲指定	0	60	0

表 3 利用するデータベースの情報の把握あり + SQL 記述スキルなしの実験結果

ルールの特徴	ツール未使用	一括表示型	逐次指定型
テーブルが一つ	0	100	100
パラメータが一つ	0	100	100
パラメータが複数	0	100	0
関数や計算式を含む	0	0	0
値の範囲指定	0	0	0

表 4 利用するデータベースの情報の把握なし + SQL 記述スキルなしの実験結果

ルールの特徴	ツール未使用	一括表示型	逐次指定型
テーブルが一つ	0	100	100
パラメータが一つ	0	100	48
パラメータが複数	0	88	0
関数や計算式を含む	0	0	0
値の範囲指定	0	0	0

## 6.4 考察

第 6.3.2 節の表から、

- 一括表示型ツールの合成ルールの機能を利用するとき、データベースの情報がなしでは、正しいルールを合成できないことが数件あった

- SQL 記述において、関数や計算式を含むクエリが一番記述するのに困難であった

などが見られた。

第 6.3.2 節の結果を踏まえると、配送ルールを管理するツールを利用しない場合、管理者が RMX を十分に運用するためには、データベースの情報の把握と SQL 記述スキルが必須であると考えられる。一括表示型ツールを用いれば、逐次指定型ツールと比較しても、様々な環境、条件で表現できる配送ルールの幅が広がっている。主に、逐次指定型ツールから引き継いだ配送ルールの追加の機能により、配送ルールの直接記述が可能になったこと、逐次指定型ツールより、改善された配送ルールの自動生成により、データベースの情報を必要としないことが要因となっていると考えられる。

しかし、関数や計算式を含むクエリや値の範囲指定をするクエリは管理者の SQL 記述スキルに依存しているため、本研究のツールでは、複雑な配送ルールの生成は十分でないと考えられる。おそらく、計算式や関数といった、管理者の意図が現れる記述部分は、サポートするには限界があるのではないかと考えられる。

## 7. 結論と今後の課題

### 7.1 結論

本研究では、主に RMX 管理者向けのツールの実装を行なった。RMX 配送ルールの自動生成機能により、RMX やデータベースの知識、SQL クエリを記述するスキルを持たない管理者にも基本となる配送ルールの生成が可能となった。また、配送ルールの合成やパラメータ数の変更といった、送信者の使用頻度に合わせて、既存の配送ルールを編集する機能も開発したた

め、生成できる配送ルールの表現できる範囲が広がっただけでなく、送信者のクエリチェックの機能も含めて、間接的に送信者のサポートも行うツールが実現された。

### 7.2 今後の課題

今回提案したツールの今後の機能拡張や、RMX 運用またはその利用のさらなる利便性の向上のために行う作業について述べる。第 6. 章でも述べたように、本研究で提案したツールで設定する配送ルールの SQL クエリには関数や計算式の利用が考慮できない、値の範囲指定ができないといったより複雑な配送ルールを設定するための機能が必要になる。

また送信者のサポートとして、クエリチェックの機能を開発したが、この機能では結果が得られるかはチェックできるが、この際、送信者の意図したアドレス通りの結果が得られているかのチェックはできない。管理者だけでなく、送信者へのサポートもまだまだ高度なものが有効になると考えられる。

以上で述べた通り、さらなる RMX の利用者の幅を広げるためならば、高度な配送ルールに対する機能の追加や改善、メール送信者に対するさらに高度なサポートも必要になると考えられる。

## 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ール配送システム". 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] "blaynmail", <http://blaynmail.jp/>
- [4] "Benchmark Email", <http://www.benchmarkemail.com/jp/>
- [5] "Acces Mail", <http://www.accessmail.jp/>
- [6] Beng HengNg, Alexander Crowell, Atul Prakash: "Proceedings of the 7th ACM Symposium on Information". *Computer and Communications Security* Pages 69-70, 2012, ACM
- [7] Arne Tauber, Jorg Apitzschb, Luca Boldrin: "An interoperability standard for certified mail systems". In *Journal of Computer Standards Interfaces '12* Volume 34 Issue 5, September, 2012 Pages 452-466
- [8] Polona Ziberman, Asaf Shabtai, Lior Rokach: "Analyzing Group Communication for Preventing Data Leakage via Email". *collaborative methods for security and privacy August, 2010*
- [9] Changyan Chi, Michelle Zhou, Wenpeng Xiao, and Eric Wilcox Min Yang: "Using email to facilitate wiki-based coordinated, collaborative authoring". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11* pages 3459-3468, New York, NY, USA, 2011. ACM.
- [10] Jaclyn Wainer, Laura Dabbish, and Robert Kraut: "Should i open this email: inbox-level cues, curiosity and attention to email". In *Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11* pp.3439-3448, New York, NY, USA, 2011. ACM.
- [11] 日比 英憲, 遠山元道: "RMX 管理者の運用サポートアプリケーションの開発". *DEIM2017, 2017*