

アーカイブ機能追加による RMX の拡張

金丸 恵美[†] 遠山 元道^{††}

[†] 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: [†]maru@db.ics.keio.ac.jp, ^{††}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配送ルールと個人の情報が登録されたデータベースに基づき、自動的にメールを転送するメール転送エージェントである。RMX の主な使用用途としてメーリングリストが考えられる。配送機能に関しては既存のメーリングリストシステムと同等の能力を持つが、配送以外に関してはアーカイブ機能が足りていない点がある。これを補いユーザーへのより便利なメーリングリスト運用を行うためには、既存システムの導入が必要となる。しかし、既存のシステムでは RMX で用いるメールアドレスの記述方法での運用に手間がかかる。このため、RMX に対応した独自のアーカイブシステムを構築する必要がある。よって、本論文では RMX に対応したアーカイブシステムを作成し、メーリングリストとしての運用をより便利にした。

キーワード 電子メール, RMX, メーリングリスト, アーカイブ

1. はじめに

RMX (Rule-based e-Mail eXchange System) とは、メールアドレスを始めとして個人の情報を登録したデータベースと、あらかじめ設定した配送ルールに基づき、自動的にメールを転送するメール転送エージェントである。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 2 次元的なものであるのに対して、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、3 次元的なものである。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。

RMX の主な利用方法として、メーリングリストが考えられる。RMX を用いると、パラメータによって条件を満たした人へのみメールを配送するため、メーリングリストと同等の配送能力を持つ。一方、配送以外の機能面に関しては、既存のメーリングリストシステムにはアーカイブやそれに伴うメールの検索などがあり、RMX ではこれらをサポートできていない。ユーザーへのより便利なメーリングリスト運用のためにはこれらの機能をサポートする必要があると考えた。この時、これらの機能を用いるために既存のシステムを用いて RMX を運用することが考えられたが、既存のシステムでの RMX を用いた運用には手間がかかる。

そこで、本論文では RMX に組み込むことのできる独自のアーカイブシステムを提案、実装する。

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。3 章では RMX の現状について、4 章では提案するアーカイブシステムの概要について、5 章では実際の使用例を述べ、6 章では評価を行う。最後に 7 章で結論を述べる。

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の配信方式である [1] [2] [3] [4] [5]。RMX ではメールアドレスを下記のように記述することによって、複数の送信先を指定する。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設定ファイルの名前に相当する。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。その内、パラメータは 1 つ以上で構成される。サブドメインによって設定ファイル (後述) が選択され、配送ルールの記述によって設定ファイル内のルール部分が参照される。その後パラメータを用いてデータベースに問い合わせを行い送信先のアドレスを取得する。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)

以下、RMX の配送ルール (2.1)、複数の配送ルールの組み合わせ (2.2)、サブドメイン設定ファイル (2.3)、拡張プラグイン機構 (2.4) について概説する。

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

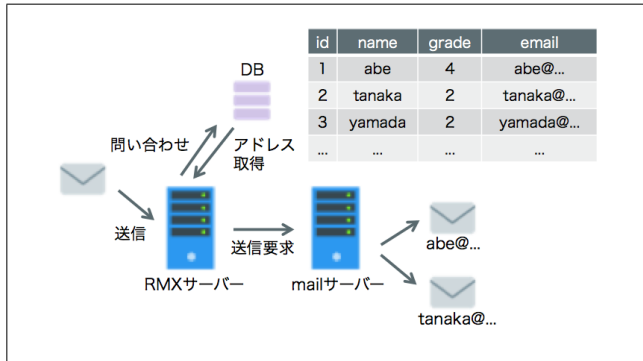


図 1 RMX におけるメール配信の流れ

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 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
deptType= String
dept[1]= select s.address from student s where
s.dept = '$1';
```

上記の例は, 所属学部ごとにメールを送信するための dept ルールである. 所属学部を String 型で受け取り, それに基づいて, メールを送信を行う. メール宛先が dept{law}@student.krmx.jp の場合, 図 2 のように query の部分で利用者のメールアドレスと所属学部が格納されているテーブル student を参照し, そこから, 学部が法学部の学生の, メールアドレスを取得するクエリを記述している.

To : dept{law}@student.krmx.jp

id	name	dept	email
1	suzuki	law	suzuki@...
2	nakai	business	nakai@...
3	mori	science	mori@...
...	...	...	...

図 2 dept ルールと配送例

2.2 複数の配送ルールの組み合わせ (関数形式)

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.

2.2.1 和集合

Syntax: $\langle name_1 \rangle \{ \langle par_1 \rangle \} + \dots + \langle name_n \rangle$

$\{ \langle par_n \rangle \} @ \langle subdomain \rangle . \langle domain \rangle$

Semantics: $name_1(par_1) \cup \dots \cup name_n(par_n)$

“+”を用いることで, 指定された複数の配送ルールの結果の和集合を取り, その和集合によって得られた結果に対して配送を行う. また, ルール間だけでなくパラメータ間でも和集合の扱いとなる.

例: dept{business}+dept{law}@student.krmx.jp

dept{business+law}@student.krmx.jp

上記の例では, 経済学部または法学部に所属している学生にメールを送信する.

2.2.2 積集合

Syntax: $\langle name_1 \rangle \{ \langle par_1 \rangle \} . \dots . \langle name_n \rangle \{ \langle$

$par_n \rangle \} @ \langle subdomain \rangle . \langle domain \rangle$

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

“.”を用いることで, 指定された複数の配送ルールの結果の積集合を取り, その積集合によって得られた結果に対して配送を行う. パラメータ間で “.” を用いると, 次の節の多相と同じ扱いとなる.

例: dept{business}.grade{1}@student.krmx.jp

上記の例では, 経済学部にも所属している 1 年の学生にメールを送信する.

また, 配送ルール間の積集合をとる “.” と和集合をとる “+” 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{mori}+grade{2}.club{soccer}@student.krmx.jp

上記のメールアドレスの場合, grade と club 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 2 年のサッカー部に所属する学生, または森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.

2.2.3 多相 (ポリモルフィック)

Syntax: $\langle name_1 \rangle \{ \langle par_1 \rangle - \dots - \langle par_n \rangle \} @ \langle$

$subdomain \rangle . \langle domain \rangle$

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

“-” は, ポリモルフィックな考え方を導入して, 1 つのルールに対して複数のパラメータを指定するための演算子である. また, パラメータ間に使用される “.” も同様の演算子である. “-” 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. 次の例ではルール grade に 2 つのパラメータ 1 と 3 を与えるため, 2 引数の grade ルールを使用する.

例 : grade{1-3}@student.krmx.jp

これによって 2 つの引数をもつクエリが実行される。引数が 2 つの grade ルールが

```
grade
gradeType= integer
grade[2]= select s.address from student s where
s.grade >= $1 AND s.grade <= $2;
```

となっていた場合、先ほどのメールアドレスから学年が 1 年から、3 年の学生にメールが送信される。

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。メールアドレスは以下の形式を取る。

```
ListOpe := -
UnionOpe := +
RuleOpe := . | + | -
Arg := string | integer
Para := Arg | Arg ListOpe Para
ParaList := Para | Para UnionOpe ParaList
Exp := rule { ParaList }
ExpList := Exp | Exp RuleOpe ExpList
Address := ExpList @ subdomain . domain
```

例 : grade{1-3}.club{soccer}+name{suzuki}.club{baseball}@student.krmx.jp

上記の例では、1 年から 3 年のサッカー部に所属している学生と、野球部に所属している鈴木という名前の学生にメールを送信する。

2.3 サブドメイン設定ファイル

サブドメイン設定ファイルでは、データベースへの接続、使用するルールの設定を管理する。ファイル名をサブドメイン名.properties とし、.properties の前をサブドメイン名として使用する。例えば、student.properties というファイルを作成し、サブドメインが student となるメールアドレスを受信した場合、student.properties を参照し、配送ルールを取得する。サブドメイン設定ファイルでは以下の項目においてデータベース情報を設定する。

- dbDriver = <データベースドライバ>
- dbUrl = <接続するデータベースの URL >
- dbId = <データベース上のユーザ ID >
- dbPassword = <ユーザ ID に対応するパスワード>
- RMX で用いる配送ルール

2.4 拡張プラグイン機構

先行研究において、メール配送以外の機能を RMX 本体の

メール配送機能から切り離された。この拡張プラグイン機構を利用することで、新たに RMX 本体に機能を追加を行うことが可能である。拡張プラグイン機能を利用するには、所定の形式のアドレスを使用する。形式を以下に示す。

形式 : #<plugin>.<command>.<arg₁>.....<arg_n>#<target>

<plugin> : プラグイン名

<command> : 使用するコマンド名

<arg> : コマンドで用いる引数

<target> : RMX 形式のアドレス、またはドメインのみ

3. 現状の RMX

RMX をメーリングリストとして活用するときの現状について説明する。

3.1 メーリングリスト

現在、RMX の主な使用用途はメーリングリストとしてグループ単位でのメール配送や、任意の宛先に対してメールを配送することが考えられる。既存のメーリングリストシステムとの比較を行った時、メール配送以外の機能に関してアーカイブ機能が足りていないことが挙げられる。ユーザへのより便利なメーリングリスト運用のために必要な機能だと考えられたため、補うために既存システムを導入することを考える。この際、RMX を用いた運用を行うと以下の問題が生じると考えられる。

• 新規メーリングリスト登録の手間

新規メーリングリスト登録の手間とは、従来メーリングリストとして運用するには特定のメーリングリストのアドレスを 1 つ取得し、そのアドレスを使う。RMX では、配送ルールの組み合わせにより自由にメーリングリストを作成することができる。考えられるメーリングリストの個数はルールとパラメータの組み合わせにより大きく膨れ上がる。その多くをメーリングリストとして使用すると外部システムを用いた場合、その都度用いたいアドレスをメーリングリストのアドレスとして登録する必要があり手間となる。

• メンバーの概念

RMX にはメンバーという概念がないため、外部システムで登録を求められる個別のアドレスを登録することができない。よって実際使用するためには、RMX のアドレスを記述することになり、メーリングリストとしてのアドレスと個別のアドレスとで重複してしまう。

こうした現状を踏まえ、本論文では、RMX に対応した独自のアーカイブシステムを提案し実装する。これにより既存のシステムを使うよりも容易に、かつ RMX でのメーリングリスト運用が便利になると考えられる。さらに、アーカイブの設定方法も RMX に沿ったものとすることでより管理が容易にできるよ

うにする。

4. アーカイブシステムの概要

4.1 アーキテクチャ

本論文での提案手法の概要を図3に示し、細かい流れを以下に示す。

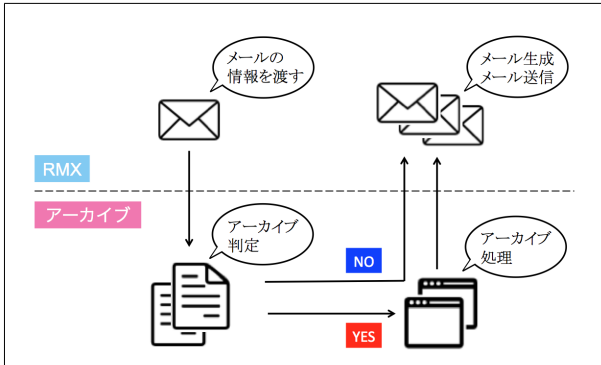


図3 提案手法の概要

- (1) RMX内でメールの情報を Archive.jar に渡す
- (2) アーカイブシステム内でアーカイブ処理の対象判別
- (3) アーカイブシステム内でアーカイブ処理
- (4) RMX内でのメール処理
- (5) アーカイブされたメールの確認

各項目についての詳細は後述。

4.1.1 アーカイブ処理の対象判別

メール受信時、RMXからメールの情報が引数としてアーカイブシステムに渡される。その際、プロパティファイルを参照しアーカイブの対象であるか判断を行う。

このためにあらかじめ、システム内にある2つのプロパティファイルに、アーカイブをするメールに関して設定をする。1つのファイルではアーカイブ処理を行うメールアドレス、2つのファイルではアーカイブした際にメールに加える文章のテンプレートを設定する。それぞれの記述形式を以下に示す。

- (1) `<subdomain>.<domain>.properties`
`<subdomain>` : サブドメイン
`<domain>` : ドメインの先頭部分

アーカイブするメールアドレスを記述する設定ファイル。設定には正規表現を用いて又は直接記述して指定する。定義できるアーカイブの種類は3つである。

- rulepara
ルールとパラメータをそれぞれ1つ指定する場合
記述の型 : `rulepara = <rule>{<para>}`
`<rule>` : 配送ルール
`<para>` : パラメータ

- multipara
ルール1つにパラメータを複数指定する場合
記述の型 : `multipara = <rule>{<para>\\W<para>...<para>}`
`<rule>` : 配送ルール
`<para>` : パラメータ
(任意のパラメータの際にはこのまま使用可)
\\W : 演算子
(任意の演算子の際にはこのまま使用可)

- rule
ルール単体を指定する場合
記述の型 : `rule = <rule>`

(2) `<subdomain>.<domain>.detail.properties`
アーカイブした際にメールに加える文章のテンプレートを指定する設定ファイル。設定できる種類は `subject` と `body` の2つである。

- subject
タイトル文頭に追加する文章
- body
本文末尾に追加する文章

設定するテンプレートに、特定の文字列を記述することでメールの通し番号や、メーリングリストアドレスを埋め込むことが可能。各文字列を以下に示す。

- 通し番号 = %c
- メールアドレス = %a

これらを文章中に記述しておくことで、アーカイブ時に実際の数値、文字列が埋め込まれる。

これらの設定をした上でメールが送信されると、アーカイブシステムがメールアドレスを解析し、該当するプロパティファイルとの比較を行いアーカイブの有無を判別する。

4.1.2 アーカイブ処理

アーカイブの判定が有りだった場合に、メインプロセスに入りアーカイブの処理を行う。メインプロセスで行う処理を以下に示す。

- アーカイブ用フォルダの作成
アーカイブするメールアドレスのサブドメイン、ドメインや、アーカイブする種類をフォルダ名としてメーリングリストごとのフォルダを作成する。
- メール情報をデータベースに格納-1

メーリングリストが使用された回数を記録するため、メールアドレスと回数を記録用データベースに登録。回数の記録は、新規登録時には 1 とし、既に登録がされている場合には値の更新を行う。記録用データベースのスキーマを以下に示す。

```
archive_count(id, subdomain, domain, type ,
               address, count)
```

- id : 主キーとしての id
- subdomain : サブドメイン
- domain : ドメインの先頭部分
- type : rulepara, multipara, rule のいずれか
- address : メールアドレスの@以前
- count : アーカイブ回数

- プロパティファイルをもとにメール編集

2 目目のプロパティファイルを参照し、メールのタイトル、本文への追加文章を生成。設定した文章中に回数を意味する文字列が存在する場合には、データベースから回数を取得し埋め込む。

- メール情報をデータベースに格納-2

タイトル、本文への追加情報を加えたメールの詳細情報を詳細データベースに登録する。詳細データベースのスキーマを以下に示す。

```
archive_detail(id, list_id, sender, subject ,
               body, date)
```

- id : 主キーとしての id
- list_id : 回数用テーブルの id
- sender : 送信者アドレス
- subject : メールタイトル
- body : メール本文
- date : 送信日時

- 作成フォルダにメールのテキストを出力

最初に作成したアーカイブ用フォルダ中に、メーリングリストの回数をタイトルとしてメールの詳細情報をテキストファイル形式で出力、蓄積する。

- 実行結果を返す

実際のメール生成を行う RMX 内のプログラムに、アーカイブシステムで編集したメールタイトル、本文を返り値として渡す。配送ルールが組み合わせられていて複数のアーカイブに該当する場合は、該当するアーカイブごとに実行結果を返す。

4.1.3 RMX 内でのメール処理

アーカイブシステムから返される値を元に、送信メールのタイトル、本文の編集を行う。複数のアーカイブに該当している場合は返り値がその分だけ存在する。返り値がない場合には通常のメール作成を行う。

4.1.4 アーカイブメールの確認

アーカイブされたメールを確認する方法は、アーカイブフォルダを参照する方法と、RMX のプラグイン機構を用いる方法との 2 種類。

- アーカイブフォルダ

アーカイブ初回時に作成されたフォルダ内にある、個別メールのファイルを参照する。しかしこの方法では、限られた人しか利用できない可能性がある。

- RMX におけるプラグイン機構プラグイン機構を利用し、閲覧したいアーカイブメールの情報を取得、閲覧する。その際に用いるプラグインは 3 種類で、以下に示す。

(1) #archive.list#

アーカイブがされているメールアドレスのリストを返す。

(2) #archive.list.<引数 1>#

引数 1 を id にもつメーリングリストアドレスのメール一覧を返す。

(3) #archive.list.<引数 1>.<引数 2>#

引数 1 を id にもつメーリングリストアドレスのメールのなかで、引数 2 を id に持つメールの詳細情報を返す。

3 つのプラグインを順に用いれば特定のメール情報を取得することができるが、この方法には手間がかかり、またメール検索ができないためユーザは限られた情報から自分が得たいメールを特定しなければならない。よってユーザへのより良いメール情報取得を支援するために Web アプリを作成した。

- Web アプリ

上記 2 つの閲覧方法では足りない機能を補う。この Web アプリで行える機能を以下に示す。

- メール検索
- メールダウンロード

5. アーカイブシステムの実装

5.1 プロパティファイルの記述例

プロパティファイルの設定例を挙げる。例に用いるサンプルテーブルを図 4、図 5 に示す。

id	name	grade	dept	email
1	abe	1	law	abe@...
2	tanaka	3	business	tanaka@...
3	yamada	4	science	yamada@...

図 4 サンプルテーブル

id	subdomain	domain	type	address	count
1	student	krmx.jp	rulepara	dept{law}	4
2	student	krmx.jp	rulepara	grade{2}	2
3	student	krmx.jp	multipara	dept{law+business}	5
4	student	krmx.jp	rule	dept	1

図 5 サンプルテーブル 2

- アーカイブ設定プロパティ

例 1 : dept が law の人へのメール

rulepara = dept{law}

例 2 : dept が law と business の人へのメール

multipara = dept{law+business}

例 2' : dept が law とそれ以外の学部の人へのメール

multipara = dept{law+<para>}

例 2'' : dept に law とそれ以外の学部が含まれているメール

multipara = dept{law\\W<para>}

例 3 : dept 単位でのメール

rule = dept

- 文章テンプレート設定プロパティ

例 1 : dept{law} に対しメールを送信する

タイトルの記述例 : [No.%c]

実際の表示 : [No.5] <タイトル>

例 2 : dept{law}+grade{2} に対しメールを送信する

(該当アーカイブが複数)

タイトルの記述例 : [%c mail]

実際の表示 : [5 mail][3 mail] <タイトル>

5.2 アーカイブメールの確認例

- アーカイブフォルダから読む

生成したアーカイブフォルダ中にあるメール毎のテキストファイルを読む方法。生成されたメールのテキストファイルの様子を以下の図 6 に示す。

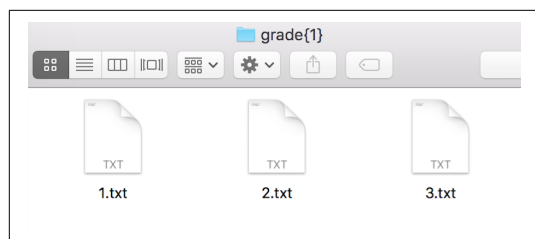


図 6 アーカイブされたメール毎のファイル

ただし現時点ではメールの番号がテキストファイルのタイト

ルとなっているため、ユーザーが求めるメールがどれなのかがすぐにわからず 1 件 1 件ファイルをチェックする必要がある。よってユーザーを支援する要素を増やす必要がある。

- プラグイン機構を用いる

(1) メールアドレスのリストを取得する

プラグイン#archive.list#を用いて、アーカイブされているメールアドレスのリストを取得する。

使用例 : #archive.list#@student.krmx.jp

(2) メール一覧を取得するプラグイン#archive.list.<引数 1>#を用いて、メールアドレス id が 2 であるメールアドレスのメールリストを取得する。

使用例 : #archive.list.2#@student.krmx.jp

(3) メールアドレス id が 2 であり、メール id が 5 であるメールの内容を取得するプラグイン#archive.list.<引数 1>.<引数 2>#を用いて、メールのリストを取得する。

使用例 : #archive.list.2.5#@student.krmx.jp

- Web アプリ Web アプリを用いたアーカイブメール閲覧

の例を図 7, 8 に示す。

dept{law}@student.krmx.jp メール一覧					
検索 : 表示					
id	sender	subject	body	date	download
1	abe@...	[No.1] test1	...	2016-01-20 17:46:33	クリック
2	abe@...	[No.2] test2	...	2016-01-21 13:51:31	クリック
5	mori@...	[No.3] 連絡です	連絡事項です...	2016-01-23 15:12:21	クリック
6	watabe@...	[No.4] 確認事項	確認です...	2016-01-23 20:56:24	クリック

図 7 メール一覧

検索 : 表示					
検索ワード: 連絡					
検索結果					
id	sender	subject	body	date	download
5	mori@...	[No.3] 連絡です	連絡事項です...	2016-01-23 15:12:21	クリック

図 8 メール検索例

6. 評価

6.1 外部システムとの比較

本論文で提案するアーカイブシステムによる RMX のメーリ

ングリスト運用の向上を評価するため、既存のメーリングリストシステムとの比較を行う [6] [7] [8]。比較対象とした既存のシステムは以下の 3 つ。

- Mailman
- fml
- Majordomo

評価項目として、メール配送以外の機能を主に取り上げた。各項目と評価結果を以下の表 1 にまとめた。

表 1 既存システムとの比較				
項目	Mailman	fml	Majordomo	本システム
配信機能				
ML 配信				
メール (コマンド) 操作				
メール一覧取得	×	×		
メール記事取得	×			
Web 操作 (ユーザ)				
メール一覧, 検索		×		
メール記事取得		×		
Web 操作 (管理者)				
各設定事項				×
その他				
記号の使用	×	×	×	
メンバー管理				×
導入の複雑さ	大きい	大きい	大きい	小さい

• 配信機能

従来の RMX でも可能であり、既存システムと同等の機能。

• メール (コマンド) 操作

メーリングリストへの操作をメールの送信により行う機能。比較した既存システム全てにおいてメール操作が可能。その中で行える操作の種類について比較を行うと、Mailman ではメーリングリストへの参加、退会、メール配信の開始/停止などは行えたが、メール一覧、記事の取得は行えなかった。一方、fml でも同じく参加、退会ができる他、get コマンドによる記事の指定取得が可能であった。しかし、記事一覧の取得は行えなかった。Majordomo、本システムでは一覧、記事取得が行えた。

• Web 操作 (ユーザ)

メーリングリストのメンバーが行える操作。

fml には、メーリングリストメンバーが使える CGI がなく、メールの記事取得はコマンド操作から行う。Mailman には CGI があり、メールを zip 形式でダウンロードすることが可能。ただし記事の検索はできない。Majordomo にも CGI があった。本システムでもこれらの操作が行えるような Web アプリを用意した。

• Web 操作 (管理者)

メーリングリストの管理者が行える操作。

既存システムでは、ユーザ使用するためだけでなく管理者がメーリングリストの設定を行うことができるようになっている。今回の実装では、ユーザへの利便性に対しての設計となったため本システムでは管理者に対しての実装ができていない。このため、現在管理者がメーリングリストの設定管理を行うには、アーカイブシステム内にある設定ファイルを直接編集する必要がある。このままでは、ファイルの編集ミスや予期せぬエラーが起こる可能性があるため、より安全に管理を行うための管理者に向けたツールの開発が必要だと感じた。

• その他

既存システムでは全て、RMX のメールアドレスに含まれる記号を使用することができない。この点では本システムの有用性が高い。

メンバー管理の項目で本システムに×がついているが、これはメーリングリストメンバーの管理が本システム上で行うのではなく、管理者がデータベースの値を更新することで行われることより、本システム上での機能として実装していないということである。

導入の複雑さでは、既存システムでは全て公開されているパッケージをインストールし、各種環境設定を行う必要がある。一方で本システムは、RMX 本体に組み込まれているため、RMX を導入しコードを少量書き換えるだけで使用が可能となる。また、既に RMX を用いている場合への組み込みもコードとプログラムを配置するだけで良い。よって複雑さが小さいと考えられる。

6.2 今後の課題

比較結果より、本システムの導入からメーリングリストとしての配送以外の機能について、ほぼ既存システムと同等の機能を持つようになった。今後の改善点として、先ほどあげた管理者に対する支援ツールの開発があげられる。また、既存システムが持つ、比較評価とした項目以外の機能についても順次実装し、さらに既存システムとの差異を減らしていくことが必要になると感じた。

7. おわりに

本研究では RMX にアーカイブ機能を組み込むことで、既存のメーリングリストシステムを導入する必要をなくし、RMX のメーリングリストとしての運用をより便利に。また、既存のアーカイブシステムとの比較を行うことで、独自アーカイブシステムの有用性を示した。

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 pp123-128 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, IEEE, June 2006
- [3] 青山陽亮, 遠山元道. “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”, DEIM2010 IEICE, 2010
- [4] 北園達也, 青山陽亮, 遠山元道. “RMX における関数形式アドレスおよびデバッグ支援機能の実装”, DEIM2011 IEICE, 2011
- [5] 松本 洋平, 北 和人, 遠山 元道. “RMX における拡張プラグイン機構の導入及び各種プラグインの開発”, DEIM2014, 2014
- [6] Mailman. <http://docs.python.jp/contrib/mailman/>
- [7] fml. <http://www.fml.org/>
- [8] Majordomo. <http://www.greatcircle.com/majordomo/>