

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

松本 洋平[†] 北 和人[†] 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: [†]{yohei,kita}@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、先行研究では、データベースを利用したメール配送の基本的な機能が実装されていて、アイデア次第で RMX に様々なアプリケーションを導入できる。しかし、現在の RMX では、アプリケーションの機能と RMX 本来のメール配信機能は分離されていないため、第三者が拡張機能を開発することが困難だった。本論文では、RMX からメール配送やデータベース情報取得機能以外の機能を切り離し、拡張機能をプラグインとして追加する手法を提案した。これらのプラグインは、本体の RMX とは分離されているため、本体側の処理に影響を与えない。プラグインの例として、ある特定のイベントへの出席・欠席を管理する拡張機能や、使用可能なルール及び各ルールの使い方をメールで確認できる拡張機能などがある。

キーワード 電子メール, RMX

1. はじめに

RMX (Rule-based e-Mail eXchange System) は管理者が設定したルールを基に、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。先行研究において、アドレスの先頭にコマンドを書き込むことで、メール配信以外にも様々な機能を使用することができる機能を開発した。しかし、このような機能と RMX 本体のメール配信機能は分離されていないため、機能を追加するごとに RMX 本体が重くなり、単体テスト・保守が難しくなるという問題があった。また、コマンドによる機能が RMX 本体から分離されていないため、第三者が RMX の拡張機能を開発することが困難だった。そこで本論文では、RMX からメール配送やデータベース情報取得機能以外の機能を切り離すことで、拡張機能をプラグインとして自由に追加できる手法を提案した。これにより、単体テスト、保守がしやすくなるだけでなく、Facebook や Twitter などのように、第三者が拡張機能を安全に開発することが可能となる。また、本論文ではプラグインの例として、ユーザ支援プラグインと、イベント管理プラグインの 2 つを提案した。ユーザ支援プラグインは、RMX において使用できるルール及び各ルールの使い方等の情報をメールの形式で取得することができる機能である。イベント管理プラグインは、同窓会や忘年会などのイベントへの出席・欠席を管理する機能である。ユーザはメールを送ることで、自分が出席する、もしくは欠席するという情報をデータベース上で更新することができる。このデータベースにより、管理者側は誰が出席か、欠席かということを知ることが可能となる。

以下、本稿の構成を示す。まず、2 章で RMX の概要について述べる。3 章で拡張プラグイン機構について、4 章でユーザ支援プラグインについて、5 章でイベント管理プラグインについて、6 章で評価について述べ、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配信方式である。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 ではアドレスの記述方法は標準形式と自然形式の 2 つがあり、これはそのうち標準形式の記述方式を示している。

< RMX のメール配信先指定 >:=

< 配送ルール名 > { < パラメータ > } @ < サブドメイン > . < ドメイン >

RMX のメールアドレスは以上のようにサブドメイン以前を表す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、“{ }”の左側にルール名、内側にパラメータが記述される。サブドメインは後述する設定ファイルの名前に相当する。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)

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

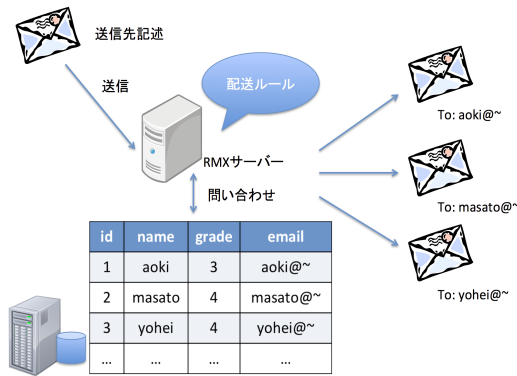


図 1 RMX におけるメール配信の流れ

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 grade{4}@rmx.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。



grade{4}@rmx.example.edu

id	name	grade	email
1	aoki	3	aoki@~
2	masato	4	masato@~
3	yohei	4	yohei@~
...	...	...	...

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 (標準形式)

ここでは、複数の配送ルールを組み合わせでメールを送る際に、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。

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

“.” 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。

例：

lab{toyama}.grade{1}@rmx.example.edu

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

“+” は、論理和を得たい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。

次の例は、同じ grade という配送ルールに複数のパラメータを与えた場合のアドレスである。

例：

grade{3}+grade{4}@rmx.example.edu

また、異なる配送ルール間で論理和を得る場合を考える。例えば学年が 4 年である、または名前が yohei である学生に対してメールを送ろうとした場合以下のように記述すればよい。

例：

grade{4}+name{yohei}@rmx.example.edu

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には、積集合をとる‘.’の方が優先順位が高いものとする。例えば以下の様なメールアドレスが作成された場合、grade と group の積集合をとった後に、その結果と name との和集合をとる。

例：

name{yohei}+grade{4}.group{rmx}@rmx.example.edu

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た 1 つのルールに複数のパラメータを指定するための演算子である。“-”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 name に 2 つのパラメータ matsumoto と yohei を与えるため、2 引数の name ルールを使用する。

例：

name{matsumoto-yohei}@rmx.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。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：

```
name{matsumoto-yohei}.lab{toyama}+name{kita}.
lab{toyama}@rmx.example.edu
```

2.3 自然形式アドレス

自然形式アドレスは以下の形式を取る。

< RMX のメール配信先指定 >:=

<パラメータ>@<配送ルール名>.<サブドメイン>.<ドメイン>

自然形式アドレスでは、“{}”を使用せず、ルール名を“@”の右に書き、一般的なメール送信で使用されているアドレスと似た形式で RMX のアドレスを記述する。

例えば、学年が3年、または4年で、名前が yohei である学生へメールを送る場合の例は以下のようになる。

例：

```
3+4.yohei@grade.name.rmx.example.edu
```

2.4 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や、使用するルールを設定を管理する。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、sample.properties というファイルを作成すれば、サブドメイン sample とすることでこの設定ファイルを参照し、同様に sample1.properties というファイルを作成し、サブドメインを sample1 とすることでこちらの設定ファイルを参照する。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>

使用するルールの設定もサブドメイン設定ファイルで行う。

3. 拡張プラグイン機構

3.1 RMX アーキテクチャ

拡張プラグイン機構を実現するために、RMX システムからメール配送やデータベース情報取得機能以外の機能を排除し、各種レイヤーごと分割する。各種レイヤーごとの依存関係を小さくすることで、単体テストを容易にし、保守性・拡張性の高いシステムを構築することが可能となる。

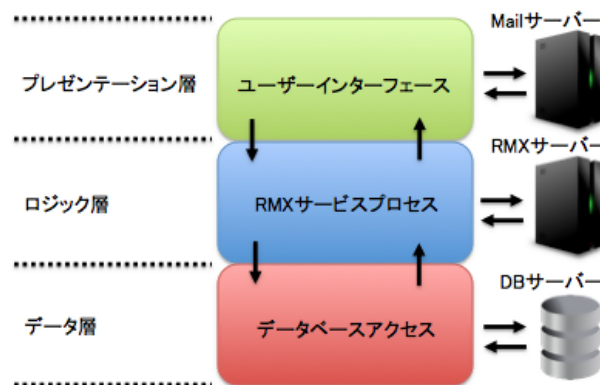


図3 RMX システムアーキテクチャ

3.2 インターフェース設計

上記 RMX システム内に拡張プラグインを読み込み、RMX の新しい機能として利用できるようにシステムを追加する。ここでのプラグインは以下のように定義した。

- プラグインは RMX システム内でどのように処理されるかどうかは考慮する必要がない
- ファイル化したプラグインを特定のディレクトリに配置するだけで自動的に RMX 内に組み込まれる
- ディレクトリ内に複数のプラグインを配置していても、全てのプラグインを認識し、メール内容や状況に応じて使用するプラグインを切り替えることができる

ここでは RMX とプラグインはインターフェースを介して結合される(図4)。RMX 内部には、“PluginInterface.java”というインターフェースが予め用意されており、インターフェース内に実装したい機能の型を宣言しておく(図5)。プラグインを作成する際には、最初に“PluginInterface.java”を継承したクラスを作成し、pluginStart メソッド、getAvailableFunctionNames メソッドを再定義する。ここで、プラグイン名を1つ設定し、getAvailableFunctionNames メソッドで取得できるようにする。プラグイン名は、他のプラグインと重複しないように設定する必要がある。また、プラグインに1つ、もしくは複数のコマンド名を設定しておき、それぞれのコマンドごとに実行する処理を決める。プラグインを使用する際に、プラグイン名とコ

マンド名を指定することで、どのプラグインのどのコマンドを実行するかを判断する。プラグイン名を指定すると、そのプラグイン名を持つ“PluginInterface.java”を継承したクラスのインスタンスが生成され、pluginStart メソッドが実行される。この pluginStart メソッドを実行することで、プラグイン内で処理が行われ、最終的に配送するメールが RMX の本体に返される。このクラス内で使用される他のクラスを 1 つの jar ファイルにまとめることで、拡張プラグインを作成する。作成した拡張プラグインの jar ファイルは、特定のディレクトリに配置することで、RMX 本体で自動的に読み込まれる。

この様に、インターフェースを継承したクラスを作成し、宣言されている機能のふるまいを記述することで、メソッド名は同じだが、ふるまいが異なる拡張プラグインを複数作成することができる。また、RMX 側でプラグインを読み込む際は型及びメソッド名は統一されているので、インターフェース継承により作成されたプラグインならば、全てのプラグインを読み込むことができる。

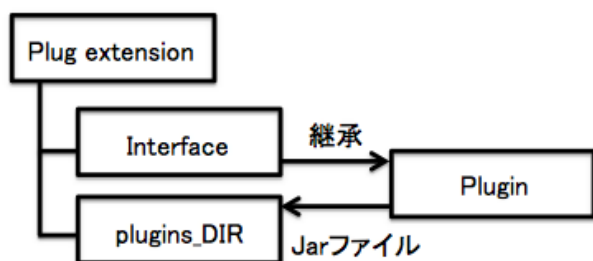


図 4 インターフェース継承によるプラグインの作成

```

public interface PluginInterface {
    public ArrayList<Message> pluginStart(Message originalMessage);
    public ArrayList<String> getAvailableFunctionNames();
}
  
```

図 5 PluginInterface.java のソースコード

3.3 プラグインの使用方法

以下の形式のメールを送信することにより、プラグインとして RMX に導入されているアプリケーションの各種コマンドを実行することができる。

#<plugin>.<command>.<arg1>...<argn>#<target>

<plugin>では、使用するプラグインのプラグイン名、<command>では、使用するコマンドを記述する。<arg>は、各コマンドで使用する引数を表している。引数の数はコマンドによって異なり、引数を 1 つも使用しないコマンドも存在する。<target>には、RMX 形式のアドレス、もしくはドメインのみを記述する。

4. ユーザ支援プラグイン

RMX は、管理者が指定したルールに基づき、データベースへアクセスを行い、対象となるユーザを見つける作業をサーバ上で自動で行う。使用できるルールは設定ファイルの中に記述されていて、編集は管理者が行う。この設定ファイルには使用するデータベースの情報が記述されているため、セキュリティの問題上、設定ファイルを一般のユーザが閲覧できる状態にすることは好ましくない。しかし、先行研究段階において、設定ファイルを自由に閲覧できない場合、自分が使用できるルールをユーザが知る方法は存在しなかった。また、使用できるルールの名前を知っていても、それがどのようなルールで、どのように使用すれば良いかという情報を得る方法が無かった。そこで本論文では、使用できるルールの名前とその使用方法などの情報をメールの形式で得る手法を提案した。

4.1 機能概要

ユーザ支援プラグインにおけるプラグイン名は“help”で、以下の様な help コマンドを使用する。

#help.help.<language>#@<subdomain>.<domain>

<language>の部分で、取得するメールの言語(en, ja 等)を指定する。

例えば、図 6 のようにメールを送ることで、各ルールの名前及び関連する情報を指定した言語で取得することができる。

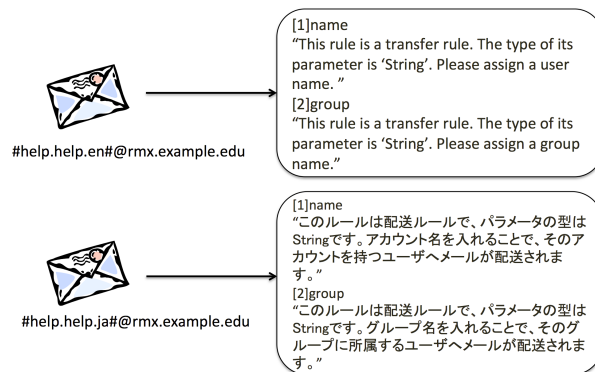


図 6 ユーザ支援プラグインの使用例

4.2 コメントの設定

ユーザ支援プラグインでユーザに提示する情報は、あらかじめ管理者側が設定しておく。ユーザ支援プラグイン用のプロパティファイルを言語ごとに用意し、その中にユーザに提示したい情報をコメントとして記述する。図 7 の例では、name, group というルールに対し、それぞれコメントを記述している。

5. イベント管理プラグイン

同窓会や忘年会などのお知らせをメールで行う場合、一般的には幹事の人々がメーリングリスト等を利用し、招待したい人々へメールを送る。次に、お知らせのメールを受け取った人々は、自

rmx_help_en.properties	rmx_help_ja.properties
name = "This rule is a transfer rule. The type of its parameter is 'String'. Please assign a user name."	name = "このルールは配送ルールで、パラメータの型はStringです。アカウント名を入れることで、そのアカウントを持つユーザーへメールが配送されます。"
group = "This rule is a transfer rule. The type of its parameter is 'String'. Please assign a group name."	group = "このルールは配送ルールで、パラメータの型はStringです。グループ名を入れることで、そのグループに所属するユーザーへメールが配送されます。"

図 7 ユーザ支援プラグイン用のプロパティファイルの設定例

分が出席する、もしくは欠席するということを幹事の人にメールで伝える。しかし、この方法では幹事が招待者からのメールを見て、誰が出席、もしくは欠席であるかを記録しておく必要がある。また、出席者のみにメールを送りたい場合、欠席者のみにメールを送りたい場合など、送り先を限定することが困難である。そこで本論文では、メールを送ることでデータベースを更新し、誰が出席か、もしくは欠席かという情報をデータベース上で管理できる機能をイベント管理プラグインとして提案した。

5.1 機能概要

イベント管理プラグインでは、member テーブル、attend テーブル、event テーブルの 3 つを使用する (図 8)。

memberテーブル				attendテーブル		
id	name	email	lab	m_id	e_id	status
1	yamada	yamada@~	toyama	1	1	attend
2	tanaka	tanaka@~	toyama	2	1	absent
3	sato	sato@~	sasase	3	2	uncertain
...	...	...	...	...	...	...

eventテーブル					
id	code	name	place	date	update
1	ob2014	2014年OB会	新宿	2014-06-08	true
2	welcome2014	2014年新入生歓迎会	渋谷	2014-04-04	false
...	...	...	...	...	...

図 8 イベント管理プラグインで使用するテーブル

member テーブルには各ユーザの情報、event テーブルには各イベントの情報を格納する。attend テーブルには、招待されたユーザの id と、招待したイベントの id を格納することで、誰がどのイベントに招待されているかを記録する。また、"status" という属性には、それぞれのユーザが出席 (attend)、欠席 (absent)、または未定 (uncertain) であるかという情報を格納する。

イベント管理プラグインにおけるプラグイン名は"event"で、使用できるコマンドは主に以下の 4 種類である。

- invite コマンド

招待メールを送る際に使用する。

- attend/absent コマンド

招待された人が、自分が出席するか、もしくは欠席するかを伝える際に使用する。

- send コマンド

招待されている人々にメールを送る際に使用する。出席者だけにメールを送る等、送り先を限定することも可能。

- close コマンド

イベントへの参加者の募集を締切の際に使用する。

5.2 invite コマンド

invite コマンドは、ある特定のイベントへの招待メールを送る際に使用する。使い方は以下の通りである。

```
#event.invite.<code>#<rule>{<para>}@<subdomain>
.<domain>
```

<code>の部分には、event テーブル内の属性"code"の値が入る。この<code>により、どのイベントの招待メールであることを識別する。invite コマンドを使用すると、メールが送られたメンバーの id と、対応するイベントの id が attend テーブルに挿入される。attend テーブルにより、誰がどのイベントに招待されているかという情報を管理することが可能となる。

例えば、"遠山研究室"という研究室に所属しているメンバーに対し、"2014 年 OB 会"というイベントへの招待メールを送りたい場合は、メールアドレスの表記は以下ようになる。

例:

```
#event.invite.ob2014#lab{toyama}@rmx.example.edu
```

5.3 attend/absent コマンド

attend コマンド、absent コマンドは、イベントに招待された人が出席・もしくは欠席の通知を行う際に使用する。使い方は以下の通りである。

```
#event.attend.<code>#@<subdomain>.<domain>
#event.absent.<code>#@<subdomain>.<domain>
```

<code>の部分で、どのイベントに対してコマンドを使用しているかを識別する。attend/absent コマンドを使用すると、attend テーブル内の属性"status"の値が"attend"、もしくは"absent"に更新される。このコマンドは、event テーブル内の対応するイベントの属性"update"が"true"の間のみ使用可能である。

例えば、"2014 年 OB 会"というイベントへの出席の連絡をする場合は、メールアドレスの表記は以下ようになる。

例:

```
#event.attend.ob2014#@rmx.example.edu
```

5.4 send コマンド

send コマンドは、招待した人々にお知らせを送る場合に使用する。使い方は以下の通りである。

```
#event.send.<code>#@<subdomain>.<domain>
```

<code>を指定することで、そのイベントに招待されている人々全員にメールを送ることができる。

send コマンドでは、引数をもう 1 つ記述することで、出席者、欠

席者, 未定の人等, 特定の人々に限定してメールを送ることも可能である.

```
#event.send.<code>.<status>#@<subdomain>.<domain>
```

<status>の部分は, "attend"(出席者), "absent"(欠席者), "uncertain"(未定) のいずれかが入る.

例えば, "2014 年 OB 会" への出席・欠席が未定の人に対し, 連絡を催促するメールを送る場合は, メールアドレスの表記は以下になる.

例:

```
#event.send.ob2014.uncertain#@rmx.example.edu
```

5.5 close コマンド

close コマンドは, 特定のイベントの参加者募集を締め切る際に使用する. 使い方は以下の通りである.

```
#event.close.<code>#@<subdomain>.<domain>
```

このコマンドを使用すると, 対応するイベントの属性"update"の値が"false"に更新され, このイベントに対して attend/absent コマンドが使用できなくなる.

例えば, "2014 年 OB 会" の参加者募集を締め切る場合は, メールアドレスの表記は以下になる.

例:

```
#event.close.ob2014#@rmx.example.edu
```

6. 評価

6.1 コード量による評価

各プラグインのコード量による評価を行った. まず, 各プラグインで共通するファイルのコード量は表 1 の通りになり, 合計すると 1578 行となった.

表 1 各プラグインで共通するファイルのコード量

ファイル名	コード量
DatabaseDao.java	82 行
GetMailAddressesFromDBService.java	31 行
ParseOf1stFormService.java	567 行
ParseOf2ndFormService.java	667 行
ParseService.java	231 行

次に, 共通部分を除いた, 各プラグイン固有のコード量は表 2 の通りになった.

今後, 新たな拡張プラグインを開発する際は, 共通部分の 1578

表 2 共通部分を除く各プラグインのコード量

プラグイン名	コード量
ユーザ支援プラグイン	116 行
イベント管理プラグイン	539 行

行のコードを書く必要はなく, 表 2 の様に, 比較的少ない量のコードを書けば良い.

6.2 イベント管理プラグインにおける評価実験

研究室の OB 会を想定し, 被験者 5 人にはイベントの幹事として, 出席予定, 欠席予定, 未定の人に対してメールを作成してもらう. メール作成は, 通常の方法と, イベント管理プラグインの send コマンドを使用する方法の 2 通りで行ってもらう. イベントに招待される人数は 5 人, 10 人, 15 人の 3 通りの場合で行ってもらう. 招待された人は, 未定の場合を除き, 幹事に出席・欠席を通常のメールで伝えていて, イベント管理プラグインの attend テーブルでも出席・欠席の情報を更新しているものとする.

メールは, 出席予定の人は図 9, 欠席予定の人は図 10, 未定の人

件名: 2014 年 OB 会へ出席される皆様
本文:
2014 年 OB 会へ出席される皆様, 遠山研 OB 会担当の
です.OB 会の開始時間を 17 時とお知らせして
いましたが, 18 時に変更になりました.
申し訳ありませんが, よろしくお願ひします.

図 9 出席予定の人に対するメール

件名: 2014 年 OB 会を欠席される皆様
本文:
2014 年 OB 会を欠席される皆様, 遠山研 OB 会担当の
です. ご連絡ありがとうございました.
次回の OB 会には是非ご参加ください.
今後よろしくお願いひします.

図 10 欠席予定の人に対するメール

件名: 2014 年 OB 会への出席が未定の皆様
本文:
2014 年 OB 会への出席が未定の皆様, 遠山研 OB 会担当の
です. 出席のご連絡は明日まで受け付けていますので,
是非ご参加ください. よろしくお願ひします.

図 11 未定の人に対するメール

イベント管理プラグインを用いる方法では, 以下の 3 つの A

ドレスに対してメールを送る。

```
#event.send.ob2014.attend#@testy.rmxdev.db.ics.keio.ac.jp  
#event.send.ob2014.absent#@testy.rmxdev.db.ics.keio.ac.jp  
#event.send.ob2014.uncertain#@testy.rmxdev.db.ics.keio.ac.jp
```

実験の結果、出席予定の人に対するメールの作成時間は図 12、欠席予定の人に対するメールの作成時間は図 13、未定の人に対するメールの作成時間は図 14 の様になった。

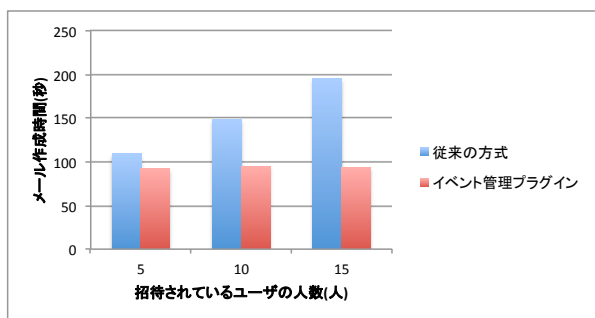


図 12 出席予定の人に対するメールの作成時間

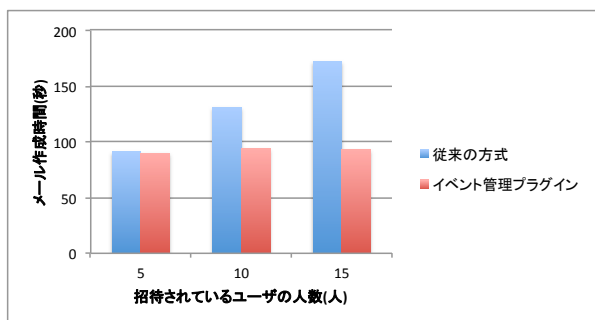


図 13 欠席予定の人に対するメールの作成時間

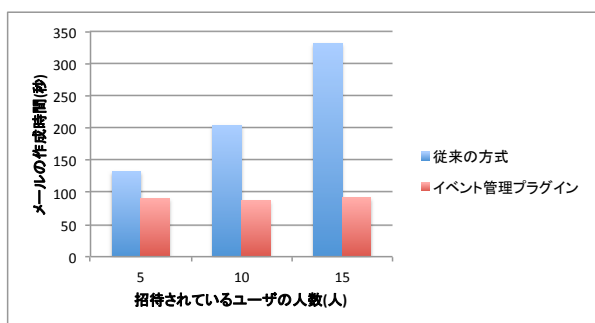


図 14 未定の人に対するメールの作成時間

従来の方式でメールを作成した場合、招待した人数が増えるにつれて、メールの作成時間が増加した。従来の方式でメールを作成する場合、招待した人々からのメールを読み、誰が出席か、もしくは欠席かを確認する必要があるため、人数が増えることで作成時間が増加したと考えられる。また、出席・欠席が未定の

人々へメールを送る場合、送り先の抽出に時間がかかり、最終的なメールの作成時間が他よりも長くなる傾向が見られた。

一方、イベント管理プラグインを使用した場合、招待人数、送り先がどの場合でも、作成時間はほぼ一定であった。send コマンドを使用した場合、送信者側が出席・欠席・未定の人々を抽出する作業を省くことができる。そのため、招待人数や送り先が変化しても、メールの作成時間に影響がなかったと考えられる。これらのことから、招待する人数が多いほど、イベント管理プラグインが有用であると考えられる。

7. おわりに

本研究では、RMX の構造を見直し、RMX における拡張プラグイン機構の導入、拡張プラグインの例として、ユーザ支援プラグイン、イベント管理プラグインの提案を行った。

拡張プラグイン機構を導入することで、第三者が拡張機能を開発できるようになり、RMX で使用したい機能を管理者自身が決定することが可能となった。

また、ユーザ支援プラグインでは、使用できるルールの名前及び使用方法などの情報をメールの形式で取得できるようになり、実際に自分が使用できるルールを知らないユーザを支援することが可能となった。

また、イベント管理プラグインでは、イベントへの招待、出席・欠席の管理等をデータベースを利用して管理することが可能となった。

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 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] 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
- [5] 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
- [6] 小船井 寛, 青山 陽亮, 北園 達也, 遠山 元道. "RMX におけるルール軍管理機構・エイリアス機構等の実装", DEIM2012
- [7] 松澤 慧, 北園 達也, 小船井 寛, 遠山 元道. "RMX における受信者別メール本文生成機能および本文参照型アドレスの実装", DEIM2013
- [8] Luca Cardelli, Peter Wegner, "On Understanding Types, Data Abstraction, and Polymorphism", In *Computing Surveys*, Vol 17 n. 4, December 1985, pp.471-522
- [9] Raphael M. Reischuk, Michael Backes, Johannes Gehrke, "SAFE Extensibility for Data-Driven Web Applications", In *WWW '12 Proceedings of the 21st international conference on World Wide Web* Pages 799-808