

RMX の配送ルールによる主要な SNS 上でのメッセージ配信

箱田 尚大[†] 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: [†]hakoda@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

あらまし RMX (Rule-based e-Mail eXchange System) とは、ルールを予め定義しておくことで、データベースから動的に宛先を取得し、メール配送を行うシステムである。我々の日常において、電子メールは未だに重要なコミュニケーションツールではあるが、近年の情報技術の普及により、企業や政府機関など様々な分野において SNS の利用が進んでいる。しかし、従来の RMX はあくまでメール転送に特化したエージェントであったため、メッセージの送信媒体が電子メールに限られているという問題がある。本論文では、RMX が持つ柔軟にメッセージの送信先を指定できるという機構を、LINE, Twitter, Facebook Messenger という国内におけるユーザ数の多い、主要な 3 つの SNS に対して導入することで、それらの SNS の管理コストの削減、運用の容易化を提案する。

キーワード RMX, 電子メール, SNS

1. はじめに

RMX (Rule-based e-Mail eXchange System) は、ルール群管理者によって予め定義された配送ルールを基にデータベース問い合わせを行い、動的に配送先を指定できるメール転送エージェントである [1][2][3][4]。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 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で、データベースからアドレスの集合を得ることができるため、従来のメールと比較するとより多様性を持つものである。

現在の RMX では、メール転送に特化したシステムであるため、メッセージの送信媒体は電子メールに限定され、その他のメディアによるメッセージのやり取りはできない。確かに電子メールは、組織などにおいて今なお必須なコミュニケーションツールではあるが、近年の情報技術の普及に伴い、各国の企業や政府機関など多様な分野において SNS の利用が促進していることから、SNS の重要性が高まっていることが伺える [5]。

そのような背景から本論文では、RMX が持つ動的にメッセージの配送先を指定できるという機構を、国内におけるユーザ数の多い、LINE, Twitter, Facebook Messenger という 3 つの SNS に対して導入することで、大規模な組織などにおける、SNS の管理コストの削減及び運用の容易化を提案する [6]。

以下、本論文の構成を示す。まず、2 章で RMX の概要について述べ、3 章で RMX の現状について、4 章で SNS への RMX 機構の導入手法について述べる。そして 5 章で評価を行い、最後に 6 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 のメールアドレスの記述方法には標準形式と自然形式という 2 つの形式があり、ここではそのうちの標準形式の記述方法を示す。

< 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 のアドレスは上記のように、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 “ . ” の記号で区別されている。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、標準形式の場合は “ { } ” 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り、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問い合わせを行い、実際の配送先アドレスを取得する。最終的に、そうして得られたメールアドレスに対して配送が行われる (図 1)。

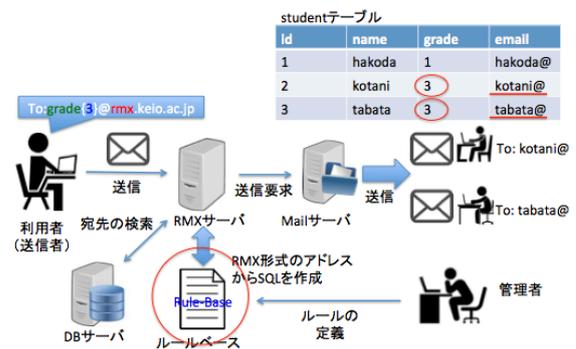


図 1 RMX におけるメール配信の流れ

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。

配送ルール名

query: 送信先メールアドレスを得るためのクエリ

RMX はアドレスに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、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得る。このような配送ルールを用いることで、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
grade[1] = select s.email from student s
where s.grade = $1
```

上記は学年を受け取り、それに基づいてメール配信を行う grade ルールの例である。例えば、メールの宛先が grade{3}@rmx.keio.ac.jp の場合、図 2 のように学生のメールアドレスとチームが格納されている student テーブルに対し、学年が 3 である学生のメールアドレスを得るクエリが実行されアドレスが取得される (図 2)。

grade{3}@rmx.keio.ac.jp

↓

id	name	grade	email
1	hakoda	1	hakoda@
2	hibi	4	hibi@
3	hatasa	2	kenji@
4	kotani	3	kotani@
5	tabata	3	tabata@

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、RMX において複数の配送ルールを組み合わせる際に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、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: <name₁> {<par₁>}. <name_n> {<par_n>}@ <subdomain>.<domain>
Semantics: name₁ (par₁) ∩ ... ∩ name_n (par_n)

“ . ” は積集合演算子を表し、複数の配送ルールによって取得される配送先集合の積集合を得られる。例えば、学年が 2 年がかつ研究室が toyama である学生に対してメールを送る場合、以下のように記述すればよい。

例: grade{2}.lab{toyama}@rmx.keio.ac.jp

2.2.2 和 集 合

Syntax: <name₁> {<par₁>} + ... + <name_n> {<par_n>}@ <subdomain>.<domain>
Semantics: name₁ (par₁) ∪ ... ∪ name_n (par_n)

“ + ” は和集合演算子を表す。複数の配送ルールによって取得される配送先集合の和集合を得られる。例えば、学年が 4 年である、または名前が hakoda である学生に対してメールを送る場合、以下のように記述する。

例: grade{4}+name{hakoda}@rmx.keio.ac.jp

また、配送ルール間の積集合を取る “ . ” と和集合を取る “ + ” では、積集合を取る “ . ” の方が評価の優先順位が高い。例えば、以下の様にアドレスが記述された場合、lab と grade の積集合を取った後に、その結果と name との和集合を取る。

例: name{hakoda}+lab{toyama}.grade{2}@rmx.keio.ac.jp

2.2.3 差 集 合

パラメータではなくルール間に “ - ” 演算子を記述することで、ルール間の差集合を取ることが可能である。例えば、学年が 2 年であるが、名前が hakoda でない学生に対してメールを送りたい場合、以下のように記述する。

例: grade{2}-name{hakoda}@rmx.keio.ac.jp

評価の優先順位は、差集合を取る “ - ” は和集合を取る “ + ” と同一順位である。また、同一順位の演算子の評価は左から順に行う。このため、一度差集合として取り除かれる配送先アドレスがその後の記述で再び取得される場合には、最終的な配送先アドレスの集合にそのアドレスも含まれる。例えば、以下の様にアドレスが記述された場合、hakoda という学生の学年が 2 年であったとしても、最終的な配送先アドレスの集合には hakoda という学生のアドレスが含まれる。

例: lab{toyama}-grade{2}+name{hakoda}@rmx.keio.ac.jp

2.3 サブドメインファイル

サブドメインルールベースでは、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。また、これをルールベースとも呼ぶ。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。サブドメインルールベースでは以下の項目においてデータベース情報を設定する。

- 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>

使用する配送ルールの定義もサブドメイン設定ファイルで行う。

3. RMX によるメッセージ送信の現状

近年、情報技術の普及に伴い、各国の企業や政府機関など多々

な分野において SNS の利用が促進している。

例として、

- 首相官邸における Facebook, LINE などの SNS を利用した情報発信
- 企業などが従業員間のコミュニケーション活性化のために運用する社内 SNS などが挙げられる。

しかし、現在の RMX はメール転送エージェントであるため、メッセージの送信媒体は電子メールに限定され、SNS といったその他のメディアによるメッセージ送信には対応していない。

こうした現状を踏まえ、本論文では RMX の動的にメッセージの送信先を指定できるという機構を、主要な SNS に導入することを提案する。

4. RMX の配送ルールによる主要な SNS 上でのメッセージ配信

以下に提案手法の概要について示す。

4.1 使用する SNS

まず、本研究において使用する SNS を選定する。その際に下記の 3 点を基準として設ける。

- (1) 国内におけるユーザ数、月間アクティブ率が共に高い値を示していること
- (2) API が Web 上で公開されていること
- (3) RMX が持つ有用性が活かせる、メッセージのやり取りが中心のサービスであること

これらの基準を考慮した結果、Facebook Messenger, Twitter を本論文においては主要な SNS とみなし、それらに対して RMX 機構の実装を施す。この 2 つのメディアを選んだ詳細な理由は以下の通りである。

まず、(1) についてだが、表 1 のように Facebook(今回は Facebook Messenger のデータがなかったため、Facebook のものを参照)、Twitter 共に、ユーザ数、月間アクティブ率の両方で高い値を示していた。

	Facebook	Twitter
ユーザ数	2,600 万人	4,000 万人
月間アクティブ率	56.1%	70.2%

表 1 国内におけるユーザ数と月間アクティブ率 (2016 年 9 月)

(2) については、前者は Messenger Platform, 後者は Twitter Developer という開発者向けの API が公開されている。そして (3) は、両者ともに多くの企業が商品の広告等に使用しており、広告の配信先を RMX のように動的に選択することができれば、その効果はより大きくなるため、RMX の有用性が活かせるサービスであると考えた。

4.2 RMX 機構を持つアプリケーションの作成

Facebook Messenger, Twitter それぞれに対して、RMX の配送ルールを用いてメッセージのやり取りが可能なアプリケーションを作成する。以下、このアプリケーションを " RMX bot " と名付け、その説明を行う。

4.2.1 アプリケーション (RMX bot) の概要

従来の RMX はメール転送エージェントであるため、企業がレコメンドメールを送ったり、あるいは大学で各学生に成績表を送ったりといったように、データベースを保持するメール送信者の、管理者向けツールとしての意味合いが強い。従って、送信者 (RMX ユーザ) と受信者が明確に分かれており、情報の伝達方向が一方向的なものであった。

それに対し SNS は、相互が手軽にコミュニケーションをとることで、人と人との繋がり (人間関係) を容易に構築できることが強みのサービスである。よって SNS に RMX 機構を導入する上で、この利点を損なわない設計にするよう留意する必要がある。この事を考慮し、RMX bot を以下のようなアプリケーションとして設計する。

- RMX bot 利用者は、RMX のアドレス記述方法を用いて、RMX bot 宛にメッセージを送信する
- RMX bot はデータベース問い合わせを行い、宛先を取得しメッセージを配送する
- RMX bot を中継することで、誰もが RMX の特性を利用することができ、相互的なメッセージのやり取りが可能なものとする

以下に RMX bot の具体的な使用例を示す (図 3)。

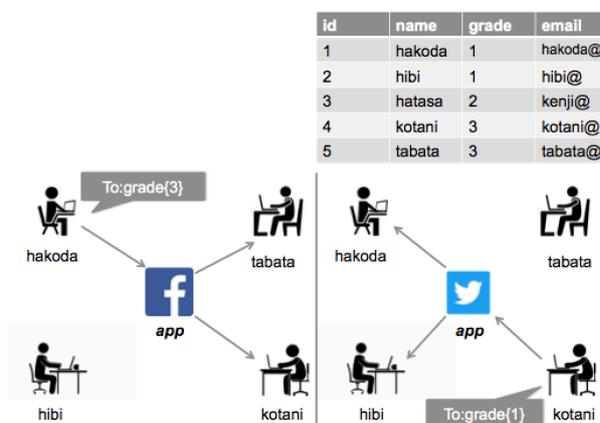


図 3 RMX bot の使用例

図の左側は、Facebook Messenger における RMX bot の使用例である。hakoda という送信者は、RMX bot 宛にアドレスを grade{3} と記述しメッセージを送信することで、そのチームの該当者に Facebook Messenger 上でメッセージを送信することができる。

一方、図の右側は、Twitter における RMX bot の使用例である。kotani という送信者は、RMX bot 宛にアドレスを grade{1} と記述しメッセージを送信することで、そのチームの該当者に Twitter

上でメッセージを送信することができる。

このように RMX bot を介せば、誰もが RMX の特性を利用できる設計にすることで、相互的なメッセージのやり取りを実現する。

4.2.2 各種設定

上記のシステムを実現する上での、各種設定について記す。

- 使用するメディアの選択

まず、使用するメディアの情報は、サブドメインファイルに記述する。下が Facebook Messenger におけるファイルの記述例である。

```
OUTMEDIA = facebook
grade
grade[1] = select s.facebook from student s
where s.grade = $1
```

このように、" OUTMEDIA " の右辺に出力メディアを記述し、その下に配送ルールを定義する。このサブドメインファイルは、各メディア毎に用意する必要がある。

- 構文解析

RMX では、電子メール形式で記述されたものを構文解析し、データベース問い合わせをするように実装されている。そのため、各メディアのテキストボックスで記述した配送ルールを、電子メール形式に変換する必要がある(図4)。



図4 電子メール形式への変換例

このシステムにおいては、RMX の構文解析器を変更しているわけではなく、各メディア毎に電子メール形式に変換する記述を用意している。そのため今後使用したいメディアを増やす際には、そのメディア用に変換するプログラムを追加すれば良い。

4.3 全体設計

以下に各メディアについての全体設計について示す。

4.3.1 Facebook

- メッセージの記述方法

RMX のアドレス記述方法に準ずる(図5)。冒頭に配送先を指定し、半角スペースを空けサブドメインを指定し、半角スペースを空け本文を記述する。このメッセージの宛先は RMX bot と

する。



図5 Facebook Messenger におけるメッセージ記述例

- メッセージ送信の流れ

以下にメッセージ送信の流れについて、順に示す(図6)。

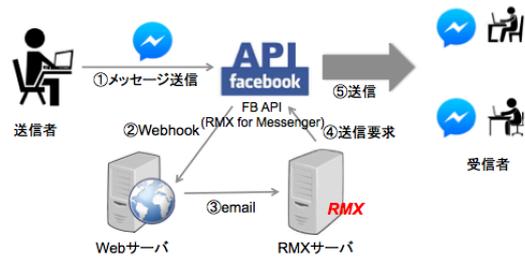


図6 Facebook Messenger におけるメッセージ送信の流れ

(1) Facebook API (RMX bot) が Web サーバに Webhook で情報を送信

(2) Web サーバ上のプログラムが Facebook API から受け取ったメッセージを RMX の電子メールの記述形式に埋め込む

(3) その電子メール形式の情報を RMX サーバに送信

(4) RMX サーバがデータベースに基づいて Facebook API に送信要求を出す

(5) Facebook API がメッセージャーで該当者にメッセージ送信

4.3.2 Twitter

- メッセージの記述方法

Facebook Messenger と同様に RMX のアドレス記述方法に準ずる(図7)。冒頭に配送先を指定し、半角スペースを空けサブドメインを指定し、半角スペースを空け本文を記述する。このメッセージの宛先は RMX bot とする。



図7 Twitter におけるメッセージ記述例

- メッセージ送信の流れ

以下にメッセージ送信の流れについて、順に示す(図8).

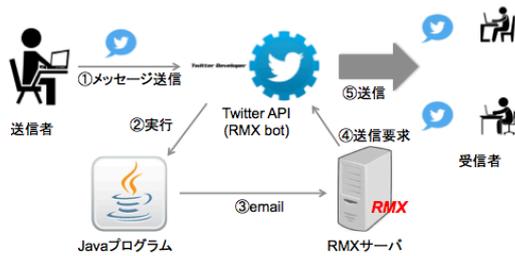


図8 Twitterにおけるメッセージ送信の流れ

- (1) Twitter API (RMX bot) がサーバ上にある Java プログラムを実行
- (2) Java プログラムが Twitter API から受け取ったツイートを RMX の電子メールの記述形式に埋め込む
- (3) その電子メール形式の情報を RMX サーバに送信
- (4) RMX サーバがデータベースに基づいて Twitter API に送信要求を出す
- (5) Twitter API がツイート形式で該当者にメッセージ送信

5. 評価

5.1 時間による評価

本研究で用いた Facebook Messenger, Twitter はいずれも、グループを作成しメッセージを一斉送信する機能は備わっている。その機能を用いてグループを作成し、メッセージを送信するまでにかかる時間と、今回作成した RMX bot を使用した際にかかった時間とを計測し、比較を行う。

5.2 閲覧率による評価

本研究では、SNS の利用が促進していることから、それらのメディアに対して RMX の機構を導入することを提案した。実際に電子メールによるメッセージ配信と、他のメディアによる配信とによってメッセージの閲覧率に差が出るかを検証する。

6. おわりに

本論文では、RMX が持つ柔軟にメッセージの送信先を指定できるという機構を、Facebook Messenger, Twitter という国内におけるユーザ数の多い、主要な 2 つの SNS に対して導入することで、それらの SNS の管理コストの削減、運用の容易化を提案した。

文献

- [1] 高畑, et al. "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." 情報処理学会研究報告データベースシステム (DBS) 2001.70 (2001): 123-128.
- [2] Kim, Hanki, Sang-Gyu Shin, and Motomichi Toyama. "A rule-based mailing system for an organization." 2006 Seventh International Conference on Web-Age Information Management Workshops. IEEE, 2006.
- [3] 青山陽亮, and 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." Syntax 1 (2010): 1.

- [4] 北園達也, 青山陽亮, and 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 (2011).
- [5] Ellison, Nicole B. "Social network sites: Definition, history, and scholarship." Journal of Computer Mediated Communication 13.1 (2007): 210-230.
- [6] Chu, Zi, et al. "Who is tweeting on Twitter: human, bot, or cyborg?." Proceedings of the 26th annual computer security applications conference. ACM, 2010.