



情報登録を自動化するメールアプリケーションの開発

メタデータ	言語: jpn 出版者: 公開日: 2013-12-20 キーワード (Ja): キーワード (En): 作成者: 辻本, 恭平, 窪田, 哲也 メールアドレス: 所属:
URL	https://doi.org/10.24729/00007630

情報登録を自動化するメールアプリケーションの開発

辻本恭平*, 窪田哲也**

Development of Mail Application for Automatic Registration System

Kyohei TSUJIMOTO*, Masanari KUBOTA**

ABSTRACT

近年、携帯電話やPCなどの普及により、一般ユーザでも簡単にe-mailを利用することができるようになった。e-mailの有効な利用方法として、基本的な文章のやり取り以外に、e-mailを用いてさまざまな処理を実行できるメールアプリケーションというものがある。本研究ではe-mailを用いてユーザ登録を行うメールアプリケーションを開発し実装した。さらに、設定ファイルを用いることでシステムの動作を容易に変更できるようにした。

Key Words: メールアプリケーション, DB, 正規表現

1. はじめに

1980年代、単純なメール送信のプロトコルとしてSMTP (Simple Mail Transfer Protocol) が開発され[1]、一般ユーザのe-mail使用が容易になった。さらに1999年、NTTドコモグループによってe-mail送受信機能を追加した携帯電話が発表された。その年以降、e-mail機能が追加された携帯電話が急速に普及し、e-mailの利用が一般的なものとなった。2005年12月末には携帯電話の普及率は89.6%にのぼり、パソコンを含めたインターネットの普及率は87%になった[2]。このように、e-mailを利用したシステムは現在さまざまな分野で非常に有用であると考えられる。

e-mailの有効な利用方法として、基本的な文章のやり取り以外に、e-mailを用いてさまざまな処理を実行できるメールアプリケーションというものがある。メールアプリケーションを使用すれば、携帯電話などの端末からe-mailを用いてコマンドを送信し、さまざまな処理を容易に行うことができる。昨年度の研究でメールアプリケーションを実装しその有用性を確認する際、MTA (Mail Transfer Agent) であるJamesを使用した[3]。Jamesとはメールアプリケーションの実装に特化したMTAで、MatcherやMailletといったJavaで書かれたプログラムを組み込むことで、容易にメールアプリケーションを実装

することができる[4]。しかし、すでにMTAが稼働している状態でJamesを起動すると、すでに動作しているMTAとポートが衝突したり、機能を追加するためにJavaを知らないユーザは言語の学習から始めなくてはならないなどの問題があった。これらの問題の解決方法は発見されているが、実際に運用することを考慮したとき、プログラムを作成してさまざまな機能を実装するよりも、テキストファイルによる設定ファイルの変更で複数の機能を切り替えて利用できる方が有効ではないかと思われる。

例えば現在、本校の体験入学などの行事の参加登録はメールや電話、Webで行っており、希望者のリストを作成する際に人の手でメールなどから登録内容を読み出し作成している。しかし、この手法では大量の登録情報を整理する場合、非常に多くの時間や労力がかかり入力ミスが起こる可能性がある。そこで、メールを受け取った時にこれらの情報を自動的に登録することができれば労力と人為的ミスの削減につながるとと思われる。さらに、返信用のテキストパターンを自分で作成し、それを簡単な手順で変更することができればさらに便利になるとと思われる。

そこで、本研究では既設のMTAからメールアプリケーションを呼び出して実行し、データベース(以下DB)と連携してメール内容を自動的に永続化するシステムについて提案する。

2. システム開発技術

2.1 メールアプリケーション

メールアプリケーションとは前述のとおりe-mailを用いてさまざまな処理ができるものであり、その代表的なものとしてメールを

2007年 4月 11日 受理

* 2006年度電子情報工学科卒業生

(Graduation of Electrical Eng. and Computer Sci.)

** 総合工学システム学科 電子情報コース

(Dept. of Industrial Systems Eng. :Electrical Eng. and Computer Sci. Course)

用いたユーザー登録やメーリングリストなどが挙げられる。図1にメールアプリケーションの基本的な流れを示す。携帯電話やパソコンなどの端末から特定のメールアドレスに対してメールが送られてきた場合、プログラムを起動しそのプログラムにメールの内容を受け渡す。プログラムは、メールのヘッダや内容を解析し、内容に応じた処理を行う。実行した結果、返信メールを送信させることも可能である。その場合はメールアプリケーションからMTAに対して返信要請を送り、あらかじめ用意しているテンプレートをもとに返信メールを送信する。



図1 メールアプリケーションの基本的な流れ

2.2 perl perlはプログラミング言語のひとつでテキスト処理やシステム管理によく用いられる言語である[5]。その理由のひとつとして、正規表現を使用することができる点が挙げられる。正規表現とは文字列の集合を一つの形式で表現する方法のことである。正規表現を用いることでテキスト内から特定の文字列を検索することができる。perlの正規表現はプログラム言語の中でも特に優れており、メタ文字の使用や他の正規表現を上回る多彩な記述方式を持っている。例えば、図2に示すプログラムはMailBox.txtの先頭からテキストを一行ずつ読み出す。読み出したデータは仮変数\$_に退避されるので、その値を\$dataに代入する。さらに、読み出した文字列に対してパターンマッチングを行っている。下線部で\$dataに代入されている行の先頭部分と“From”という文字列を比較している。一致した場合、その行の文字列の中からメールアドレスの形式と一致する部分を取り出す。取り出した文字列は仮変数\$1に退避されるので、\$1の内容を変数\$Fromに代入する。こうすることでメールのヘッダファイルから送信元のメールアドレスだけを抽出することが可能である。

```

open IN "MailBox.txt";
while(<IN>){

    $data =~ $_;

    if($data =~ /^[^From ]){
        $data =~ /([\\w\\-\\~]+@[\\w\\-\\~].([\\w\\-\\~.]+))/;
        $From = $1;
    }

}
close IN;
  
```

図2 正規表現を用いた perl プログラム

もうひとつの理由として、高い拡張性があげられる。perlではモジュールと呼ばれるものを追加することで、ネットワーク接続やDBアクセスなどさまざまな機能を追加することができる。

3. システム

3.1 システム概要 今回作成したシステムは、メールを受け取った際にメールの送信元アドレスと自動生成した8桁の英数字による仮パスワードをDBに登録し、そのパスワードと登録が成功したことを知らせるメールを送信者に返信する。また、仕様書の手順通りに行うことでこのシステムをMTAに容易に実装することができ、さらにDBに登録する項目についてもカスタマイズすることができる。今回、システムの検証用にサーバを構築した。サーバのOSにはFedora Core Linux 5を、MTAには postfix-2. 2. 8-1. 2を使用し、DNSにはbind-9. 3. 2-4. 1を使用した。

3.2 メールアプリケーション開発 メールアプリケーションを開発するにあたり、図1の流れをもとに開発した。まず始めに実行プログラムとしてDBにアクセスし、メールアドレスとパスワードを登録するプログラムを作成した。perlそのものにはDBにアクセスする機能が無いためDBIおよびDBDというDBアクセスに必要なモジュールをインストールした。プログラムの流れとして、まずDBにアクセスし現在登録されているIDのうち最も大きいIDを読み出す。その後IDを1増やして、パスワードとメールアドレスをそれぞれの項目に挿入するようにした。次にメールの内容の受信および返信要請の部分として、メールから情報を抽出し返信メールを作成して送信するプログラムの作成を行った。perlにはDBアクセス同様メールを送信する機能がないため、NET::SMTPというモジュールをインストールした。こうすることでSMTPサーバにアクセスすることができ、メールを送信することが可能となった。作成したプログラムはメールボックスから先頭のメールのヘッダ情報と本文を読み出し、正規表現を用いてヘッダ情報から送信元アドレスと件名を取り出す。そして、抽出された送信元アドレスに対して取り出した本文と件名を用いて返信メールを送信する。

これら二つのプログラムを組み合わせ、メールの本文をDBのパスワード項目に登録し、送信元アドレスをユーザのメールアドレス項目へ登録するようにした。さらにDBへの登録が完了した時点で登録完了のメールを返信するプログラムを作成した。しかし、これらのプログラムはコマンドラインからの入力で行われるためメールアプリケーションとして動作するものではなかった。そこで、メールアプリケーションとして動作させるために、

メールが届くと同時にプログラムが自動的に実行されるように設定した。これにはメールエイリアスというものを利用した。メールエイリアスとは、メールアドレスに別名をつけることができる。それにより、実態として一つのアドレスを複数のアドレスで使い分けことができ、どのアドレスに送っても同じアドレスに届くよう設定することが可能となる[6]のものである。その他にも、一つのエイリアスに複数人のアドレスを設定することで簡易メーリングリストを作成したり、プログラム名を指定することによってメール受信時にそのプログラムを起動し、標準入力の引数のようにメールの内容をプログラムへ渡すことができる。これを用いる事で、先に作成したプログラムをメールアプリケーションとして動作させることが可能となった。また、ユーザ登録に必要な機能として、仮パスワードの自動生成を行うプログラムを作成した。先に作成したプログラムでは本文の内容をパスワードとして登録するようにしているため、システムが自動的にパスワードを発行するシステムに変更する必要があった。仮パスワードは、アルファベットの大小52文字と数字10文字からランダムに8文字を選択して生成し登録するようにした。しかし、パスワードをそのまま登録するのはセキュリティ面で不安がある。そこでSHA-1というハッシュ関数を用いて変換した値を登録することにした。SHA-1は一方向関数であり、もとのデータからハッシュデータへの変換は容易だが、ハッシュデータからもとのデータへの変換は困難である[7]。図3は受信メールからメールアドレスをmailカラムへ登録し、自動的に生成して変換したパスワードをpassカラムへ登録した結果である。その後、図4のようなメールを作成し返信をしている。

1	12345678	test@localhost.localdomain
2	02c9c41cc649b44e9f4d828acd188a19b8569691	e14@localhost.localdomain

図 3 登録後の DB

送信元	test@localhost.localdomain
宛先	e14@localhost.localdomain
日付	Fri, 2 Feb 2007 11:31:50 +0900 (JST)
件名	登録完了
Message text	
e14@localhost.localdomainを登録しました。	
あなたのIDは2です。 パスワードは6rBy3GG9です。	

図 4 返信メール

いても個々のユーザ登録によって要求されるものが異なる。その場合、プログラムを直接書き換えるのではなく設定ファイルによりアプリケーションが管理されていれば、Configファイルを書き換えることでメール内容やその他の設定を変更することが可能になり、システムの利便性が高まると思われる。そこで、本システムでは返信メールのテンプレートと使用するDBがConfigファイルで変更できるようにプログラムの修正を行った。その実行結果を図5および図6に示す。機能を変更するメールアプリケーションと同じディレクトリにあるConfigファイルのmessという項目を変更すると、返信メールの内容が変更可能となる。Configファイルの変更により返信メールの内容が図4から図6のように変更された。同様にDBnameという項目を変更すれば使用するDBを変更することができるようになった。

Configファイルは現在最小限の設定のみ対応している。そこで、プログラムそのものを変更することを考慮し、項目ごとに分割してモジュール化するようにした。こうすることで、変更したい部分以外のプログラムはそのまま使用することができるようになる。モジュール化した際、外部モジュールを呼び出す方法として、はじめはsystemコマンドを用いてモジュールを起動するようにしていた。しかし、MTAに実装した場合、パーミッションの関係からうまく起動できなかった。そこで、perlがインタプリタであることを利用して、スクリプトを読み出して実行するようにし、この問題を解決した。インタプリタ言語はコンパイルする必要がないため、設定ファイルを変更した場合でもシステムを再起動する必要がない。こうすることでユーザ側の手間をさらに減らすことができた。

```
dir = /etc/MA/samp
DBname = "DB1"
driver = "mysql"
server = "localhost"
mess = "test.txt"
      ↑test2.txtに変更
```

図 5 Config ファイルの内容

送信元	test@localhost.localdomain
宛先	e14@localhost.localdomain
日付	Fri, 2 Feb 2007 11:32:45 +0900 (JST)
件名	メール内容変更
Message text	
メールの内容を変更しました。	

図 6 変更後の返信メール

3.3 設定ファイルによる修正機能の実装 ユーザ

登録に必要な項目は運用方法によってさまざまな組み合わせが考えられる。さらに、返信用のメールの文章につ

4. まとめ

本研究ではメールを受信した際, 8文字の仮パスワードを自動生成し, それと送信元アドレスをDBに登録し, その後登録成功の内容と自動生成したパスワードを記述したメールを返信するメールアプリケーションを構築した. さらにこのシステムを容易に実装可能なようにインストール手順と設定変更方法を記した仕様書を作成した. これにより誰でも簡単にユーザ登録のメールアプリケーションが実装できるようになった. しかし, 現段階では送られてきたメールの本文からデータを抽出して登録をする場合, こちらが決めた形式で記述しなければ情報を取り出すことができない. 形式が少しでも異なれば登録することができないため, メール本文の記述方法に依存しないように柔軟性を持たせる必要がある.

今後の課題として, 自然言語処理を用いて日常会話の文章の中から特定の情報を抽出することで, メール形式に依存しないシステムにすることがあげられる. また, 設定変更する際に一般ユーザが変更しやすいようにGUI (Graphical User Interface) を用意することが考えられる.

謝辞

本報告は平成18年度卒業研究論文をまとめたものである. 卒業論文の作成にあたり, 副査としてご指導, ご鞭撻を賜った大阪府立工業高等専門学校総合工学システム学科電子情報コース重井宣行助教授に心より感謝いたします.

参考文献

- [1] 梁瀬介次, インターネット・プロトコル詳説 (5) SMTP (Simple Mail Transfer Protocol)
<http://www.atmarkit.co.jp/fnetwork/rensai/netpro05/netpro01.html>
- [2] 社会実情データ図録
インターネット世帯利用率の推移,
<http://www2.ttcn.ne.jp/~honkawa/6200.html>,
携帯電話世帯普及数
<http://www2.ttcn.ne.jp/~honkawa/6350.html>
- [3] 丸谷将太, "MailアプリケーションのWebアプリケーションへの実装", 大阪府立工業高等専門学校卒業研究報告書, 2006
- [4] 【Jakarta/Apacheウォッチ】第7回 メール・アプリケーションの強力なプラットフォーム
James,
<http://itpro.nikkeibp.co.jp/members/ITPro/oss/20040417/1/>
- [5] アスキーデジタル用語辞典, 検索語 "perl",
<http://yougo.ascii24.com/gh/18/001875.html>
- [6] IT用語辞典 e-Words, 検索語 "エイリアス",
<http://e-words.jp/w/E382A8E382A4E383AAE382A2E382B9.html>
- [7] IT用語辞典 e-Words, 検索語 "SHA-1",
<http://e-words.jp/w/SHA-1.html>