

# インターネット上の新しいコミュニケーションツール

増 井 俊 之†

ネットワーク上では、古くからの様々な情報交換ツールが使用されており、時代とともに人気が変遷している。従来は電子メールやメーリングリスト (ML)、ネットニュース、掲示板などがよく使われていたが、最近は Web ページ上の掲示板、メールマガジン、インスタントメッセージャー、Wiki Wiki Web のような新しい情報交換ツールも広く用いられるようになってきている。

本稿では、新しいコミュニケーションツールとして今後が期待される Wiki Wiki Web とその拡張について述べ、またメーリングリストを非常に簡単に作成することができる QuickML システムについて紹介する。

## New Group Communication Tools on the Internet

TOSHIYUKI MASUI†

Various new group communication tools on the Internet are emerging, and people can use a system that is most convenient for their communication needs. In this talk, we introduce two new systems for effective group communication. First, we introduce the *Wiki Wiki Web* system, which is a set of Web pages where anybody can modify the contents using standard Web browsers. Second, we introduce the *QuickML* mailing list management system, with which anybody can create and manage his own mailing list only by sending an e-mail message to the mailing list server.

従来のネットワーク上では電子メールやネットニュースが情報交換ツールとして広く使用されてきたが、Web や携帯電話の普及にとともに、様々な新しい情報交換ツールが登場してきている。

本稿では、Web 上の新しいコミュニケーションツールとして今後が期待される Wiki Wiki Web 及びその関連システムについて紹介し、また携帯電話などからメーリングリストを非常に簡単に作って運用することができる QuickML システムについて紹介する。

### 1. Wiki Wiki Web

Wiki Wiki Web<sup>7)</sup> (以下 Wiki と略) は、オブジェクト指向分析やパターンランゲージのコンサルティングを行なっている Cunningham & Cunningham, Inc. が開発した協調作業支援システムで、誰でも自由に Web ページ群の編集を行なうことができるという特徴を持ったシステムである。一般の Web サイトでは、サイト管理者のみがページの修正を行なうことができるようになってきているのが普通であるが、Wiki サイトの各ページには「編集」ボタンがついており、CGI を使って誰でも自由に中身を修正したり追加したりすることができる。

Wiki の各ページには、大文字と小文字を交互に並べた「WikiName」と呼ばれる形式の名前がついており、文字列中に WikiName を記述するだけでそのページへのリンクが自動的に作成されるようになってきている。たとえば「FrontPage」

という名前の Wiki ページがある場合、別のページの文中で「FrontPage」という名前が出現するとそれらはすべて FrontPage へのリンクに変換される。

Wiki ページに書込みを行なうとき、WikiName がリンクに変換されるだけでなく、簡単なマークアップタグを使用してテキスト整形を指令することができる。たとえば、入力テキスト中で“----”という記述を行なうと、これは横線を示す HTML タグ (“<hr>”) に変換される。表 1 にテキスト整形の規則を示す。

WikiName	別ページへのリンク
空行	<p>
----	<hr>
tab *	リスト項目レベル 1 (<ul>/<li>)
tab tab *	リスト項目レベル 2
tab 1	数え上げ項目 (<ol>/<li>)
tab (項目): (説明)	数え上げ項目 (<dt>/<dd>)
行頭空白文字	<pre>指定
tab 空白文字: tab	引用
''(文字列)''	<i>(文字列)</i>
'''(文字列)'''	<b>(文字列)</b>
URL	リンク

表 1 Wiki の変換規則

#### 1.1 Wiki 使用例

図 1 は Wiki の FAQ ページの先頭部分で、図 2 は FAQ ページの最後の部分である。これだけだとごくシンプルな

† ソニーコンピュータサイエンス研究所

Sony Computer Science Laboratories, Inc.

<http://www.c2.com/>

“WikiWikiWeb”, “FrontPage”などは WikiName であるが、“wiki”,

“ABC”, “POBox”などは WikiName ではない。

<http://c2.com/cgi/wiki?TextFormattingRules>



図 1 Wiki FAQ の先頭部分

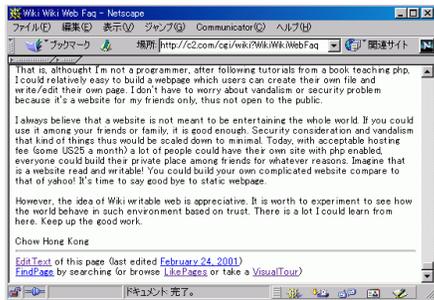


図 2 Wiki FAQ の最後

Web ページに見えるが、図 2 の左下の「Edit」リンクをクリックすると図 3 のようにページ内容編集画面があらわれて内容の修正が可能になる。

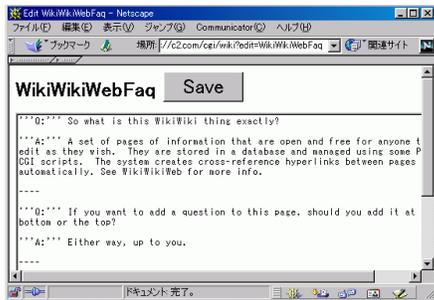


図 3 Wiki Faq の編集画面

編集後「Save」ボタンを押せば FAQ ページが本当に書き換えられてしまう。このように大事なページは通常はページ管理者しか編集できないようにするものであるが、Wiki では誰でも編集できるようになっているのが面白い。

図 3 では強調フォントを指定するためのマークアップ「'''」と横線を指定するためのマークアップ「----」が使われている。

Wiki の原理は非常に単純であり、使用方法も簡単なので、アイデア次第で幅広い範囲に応用することができる。インターネット上の不特定多数が情報共有を行なうための黒板のように利用するのに便利であるし、既存の掲示板と同じような使い方をすることもできる。「超整理法」のように、情報

を時間順に並べて個人的なメモを管理するといった使い方もできる。

オリジナルの Wiki は約 300 行という小さな Perl プログラムなので、これを拡張して様々な機能を追加する多くの試みが行なわれている。表 2 のようないろいろな「Wiki クローン」が出現して利用されており、現在もその数は増え続けている。

名前	実装言語	システム
YukiWiki <sup>1</sup>	Perl	Web
RWiki <sup>2</sup>	Ruby	Web
Tiki <sup>3</sup>	Ruby	Web
PukiWiki <sup>4</sup>	PHP	Web
PalmWiki <sup>5</sup>	C	Palm
HashedWiki <sup>6</sup>	PHP	Web
PnutsWiki <sup>7</sup>	Pnuts	Web
EmacsWiki <sup>8</sup>	Emacs Lisp	Emacs

表 2 様々な Wiki システム

これらのシステムでは、システムや言語が違うだけでなく、様々な見栄えの変更、機能の変更、タグの拡張などが行なわれている。特に、日本語を WikiName のかわりに使うためにはいろいろ苦労がされているようである。

## 1.2 Wiki の問題点

Wiki やそのクローンシステムは、単純な枠組みの割には大変便利であるが、技術面や運用面で問題が感じられるところも散見される。

### 悪意あるユーザの危険

誰でも自由に書き込みを許すようにした場合、誰かに情報を削除されてしまう危険は常に存在するし、無用の情報を書き込まれてしまう危険もある。書き込み情報のログを保存しておけば回復は可能であるが、手間がかかる可能性があることは確かである。

### 同時更新による上書き

一般的な Wiki では編集時にロックを行なわないものが多いため、複数のユーザが同時に書き込みを行なった場合は新しいものだけが保存されることになってしまう。

### 時間情報がわかりにくい

ひとつのページのいろいろな場所にいろいろなユーザが情報を書き足す場合、誰がいつどこに書き込みを行なったのかわからないので、ページ内の構造がわかりにくくなる場合が多い。Wiki ページを掲示板のように利用する場合は、新しい情報は必ずページのトップに記述するとか書込んだ人の名前を常に明記するといった規則を決めておく必要がある。

### 編集モードの存在

ページ編集の手間が多いのも問題になる。Web ページ上の Wiki のページを編集する場合、まず「編集」ボタンを押して編集モードに入った後、修正を行ない、「書込」ボタンを押すことになるが、1文字修正するだけの場合でもいちいちこのような手順を踏むのは結構面倒なものである。通常のエディタの場合は、現在注目している文字の場所にカーソルやマウスを移動して修正を行なうだけですむが、一度編集モードに入ってから修正箇所を捜して修正して書込むというのは手間がかなり違う。

### 記述フォーマットの制限

Wiki にはテキスト整形タグが用意されているが、HTML の全機能を使えるわけではないので、あまり凝った型式のページを作ることはできない。たとえばオリジナルの Wiki

	掲示板	Wiki
利点	時間順や投稿時刻がわかりやすい 誰からの情報かがはっきりわかる 世間でよく認知されている	誰でもどこからでも議論に参加可能 簡単に Web ページを作ることができる ひとりで使ってもなんとかなる 時間関係がわかりにくい
欠点	古い情報が消えてしまう ひとりで使うことはできない	誰が編集したのかわからない なんとなく敷居が高い

図 4 Wiki と掲示板の比較

では枠つきテーブルを表示することもできない。

直接 HTML を書きたい場合も多いと思われるが、そのような機能は通常は用意されていない。このため、Wiki のページは比較的地味なものが多く、最近の派手な Web ページに混じると貧弱に見えてしまう。

#### 検索機能

多くの Wiki クローンでは検索機能が用意されているが、充分とはいえないようである。

#### バージョン管理

ページ内容を変更するとき、古いページを残すようなバージョン管理を行なっているシステムもあるが、古いデータを検索することはできないなど、可能な操作は限られていることが多い。

#### 1.3 さらになる拡張

Wiki の問題点を解決し、さらに使いやすいシステムとするための各種の方法が考案されている。

#### 掲示板との融合

Wiki も掲示板も、いろいろな人々の間で情報を交換したり共有したりするためのツールであるという点は共通しているが、本原稿執筆時点 (2003 年 1 月) 時点では、Wiki よりも掲示板の方がはるかに広く使用されているようである。

図 4 のように、Wiki と掲示板には本質的な違いがあり、相補的な特徴も多い。掲示板は時間的な経過はよくわかるが、話題がどんどん変化していくため、固定的な情報を編集するには向いてない。一方、Wiki の場合は時間経過はあまり明らかではないが静的な情報を生成するには向いている。たとえば、ひとつの話題について議論するには掲示板が向いているのに対して、その議論の結果をまとめて保存しておくには Wiki が向いているということができる。

掲示板と Wiki をうまく組み合わせれば両者の利点のみを活用することができると思われる。Wiki ページに投稿フォームを用意し、投稿順にページが構成されるようにした「パラグラフベース」の Wiki が最近流行しているようである。

#### フィルタ利用

Wiki では、書込みを行なうとき、WikiName をリンクに変換したり「\*」の並びを<li>タグに変換したりといった処理を行なっているが、さらに高機能な処理を行なうフィルタをいろいろ用意しておけば便利である。たとえば、日付とテキストの列を表に変換してカレンダーのように表示するフィルタや、URL とタイトルのリストをリンク集のように変換するフィルタなどが考えられる。

#### 各種ライブラリの利用

手作業で Wiki ページの編集を行なうかわりに、掲示板のような投稿フォームを用意することにより、Wiki と掲示板を融合することができるが、投票フォームやリストの順番入れ替えといった各種の機能をもつ CGI を用意しておけば、こういった機能を簡単にライブラリのように使用することができる。

#### 1.4 WikiBBS

これらの拡張をふまえて、著者も簡単な Wiki システムを作成して利用している。

日記にしてもメモ帳にしてもスケジュール管理にしても議論にしても、極論すれば、人々が生成した文章を並べたものにすぎない。いろいろな人の意見をトピック毎に時間順に並べたものは掲示板に見えるし、意味を考えて読みやすいように並べたものは普通の文書に見えるであろう。また、考えたことをとりとめもなく時間順に並べたものは日記に見えるし、予定だけ取り出してカレンダーの上に並べればスケジュール帳に見える。

最初から定型文書を作成することを目標とせず、非定型文書を並びかえることにより定型文書を作成するという方針にしておけば、メモをとったり意見交換したりスケジュール調整したりといったような非定型要素の多い仕事に向けた文書管理システムができると考えられる。

このように考えると、次々に作成されるテキストを時間順に並べて表示する仕組みと、内容ごとにまとめて編集しやすい仕組みさえあれば、多くの仕事をうまくまとめることができそうである。前者のためには掲示板を使用し、後者のために Wiki を使用すればよいというアイデアにもとづき、掲示板と Wiki を融合した WikiBBS システムを試した。

WikiBBS では以下のような点を目標とした。

- Wiki と掲示板を混在させられるようにする
- 超整理法的な情報管理手法と相性が良いようにする
- 一度入力した情報は決して消さない
- フィルタ機能など Wiki の拡張機能を導入する

#### タイトルリスト

図 5 は、すべての掲示板と Wiki ページをリストしたトップページである。Wiki ページと掲示板は色分けされている。

Wiki / 掲示板	最終更新	10分	1時間	1日	1週間	1月	1年	10年
予定表	2008/10/26 22:36:08							
掲示板とWikiの比較	2008/10/26 22:26:29							
日記	2008/10/26 22:04:14							
家計簿	2008/10/26 22:02:09							
PageNo リーバの計算機能	2008/10/26 20:26:55							
ブックマーク	2008/10/26 19:51:17							
面白情報	2008/10/26 09:58:23							
パーコードによる書籍管理	2008/10/26 08:40:03							
晴/ゴジップ	2008/10/26 08:36:33							
更新履歴	2008/10/26 08:30:23							
ビデオ上映会	2008/10/25 14:47:56							
名言集	2008/10/24 10:44:58							
CSL ニュース	2008/10/24 10:32:32							
掲示板システム改良	2008/10/23 23:55:47							
掲示板アクセスの視覚化	2008/10/23 23:50:14							

図 5 タイトルリスト

右側には、各掲示板や Wiki ページのアクセス状況を視覚化して表示している。たとえば過去 10 分以内にアクセスされた場合は一番左の部分に棒が表示され、時間が経過することに右に移動していく。

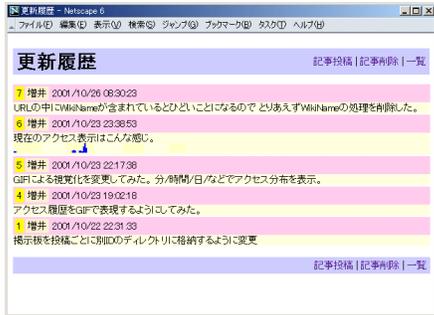


図 6 掲示板の例 (更新履歴)

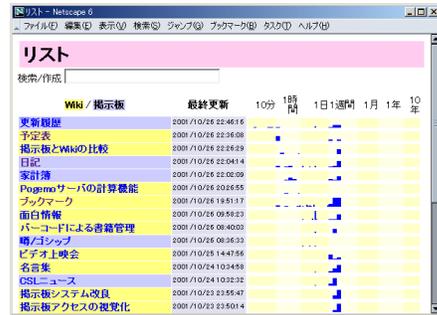


図 9 記事投稿後のリスト

## 掲示板

「更新履歴」という掲示板を選択すると図 6 のような掲示板が表示される。更新履歴のように時間の経過が重要な場合は掲示板形式で情報を格納しておくことで便利である。

## 掲示板への投稿

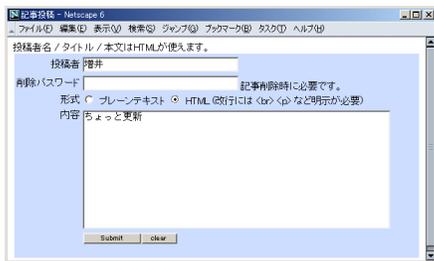


図 7 記事投稿

「記事投稿」をクリックすることにより、図 7 のようなフォームを使って新しい記事を投稿することができる。投稿後の掲示板は図 8 のようになる。



図 8 投稿後

また、掲示板が更新されたので全体のリストも図 9 のように変化し、「更新履歴」がリストのトップに移動する。

## Wiki ページの編集

図 10 に Wiki ページの例を示す。このページのように、時間と関係のない情報を格納するには Wiki ページを使うのが便利である。

ここで「編集」ボタンを押すと、図 11 画面のように Wiki ページを編集することができる。

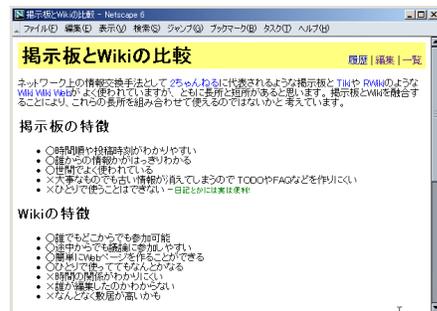


図 10 Wiki ページの例

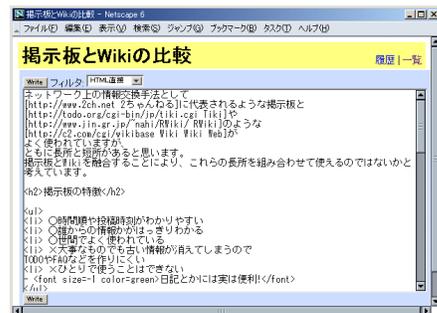


図 11 Wiki ページの編集

ここでは HTML を直接記述しているが、“[”, “]” で URL と名前を囲むことによりリンクを記述する簡易記法も採用している。

“[[” と “]]” で名前を囲むと、その名前をもつ Wiki ページへのリンクが生成され、“<<<” と “>>>” で名前を囲むと、その名前をもつ掲示板へのリンクが生成されるので、掲示板や Wiki ページ間で簡単に相互参照を行なうことができる。

## Wiki ページのフィルタ

図 12 は予定表を示す Wiki ページである。このページはユーザが予定表の HTML を直接記述しているわけではなく、単純な予定データをフィルタに通して HTML のテーブルを自動生成することによりカレンダーを生成している。

図 12 で「編集」をクリックすると図 13 のような編集画面が表示される。このように、データはごく簡単なテキストで表現されており、これにカレンダー用のフィルタを通すことにより図 12 のような HTML が生成されるようになっている。



図 12 予定表



図 13 予定表編集画面

カレンダー以外にもいろいろなフィルタを用意しておくことにより、いろいろな用途に対応させることが可能である。たとえば毎日入力される家計簿データをもとにして収支グラフを生成するフィルタを作ることができる。

### WikiBBS の運用

筆者は 1 年以上にわたり、アイデアや予定などあらゆる個人情報の整理に本システムを使用している。あらゆる情報を Web 上で一括管理できることは非常に便利である。また、グループウェアとして、WISS2002 ワークショップや未踏プロジェクトの情報交換にも使用している。WISS2002 ワークショップでは会場にサーバを設置することにより、会議参加者がいろいろな情報を共有するのに非常に便利であった。とはいうものの、現在のところはまだまだユーザが多いため、グループウェアとして他のシステムに比べてどれほど有用であるかは不明である。

## 2. QuickML: 超お手軽メーリングリスト

ネットワーク上での情報交換のために「メーリングリスト」が広く使用されている。メーリングリストとは、複数の参加者に同じメールを簡単に送れるようにしたシステムで、あるひとつのアドレスにユーザがメールを送ることにより、参加者全員に同じメールが配送されるようにしたものである。沢山の人間でメールをやりとりする場合、すべてのメンバを毎回宛先に指定するのは面倒であるが、メーリングリストを使えばひとつのアドレスに送ったメールが自動的に全員に届くので、すべての宛先について気にする必要がない。またメールシステムによっては宛先の数によって通信料金が増える場合もあるので、通信費の節約になる場合がある。

メーリングリストは、インターネットが普及する以前から広く使用されてきた。「インターネットはメーリングリストに始まりメーリングリストに終わる」と主張している人間もいる<sup>11)</sup>。

### 2.1 メーリングリストの問題点

メーリングリストを使うと、メールを送るだけで手軽にグ

ループメンバ間で情報交換を行なうことができるので便利であるが、このような機能を実現するためにはメール配送システムをカスタマイズする必要があるために、実装はあまり簡単でないことが多い。

通常、電子メールの配送は、プロバイダや会社、学校などの組織にあるメッセージ転送エージェント (MTA) と呼ばれるシステムが面倒をみている。普通のユーザがメールの読み書きに使う Outlook, Eudra, Becky のようなおなじみのシステムはメッセージユーザエージェント (MUA) と呼ばれ、メッセージの配送処理は行なっていない。

メールを複数のメンバに配送するというメーリングリストのような処理は MTA で実行する必要があり、MUA でメーリングリスト機能を実現することはできない。たとえば Unix の場合、メール配送システムは通常はシステム管理者でないと変更することができないので、メーリングリストを作成するにはシステム管理者に作成や管理を依頼しなければならない。自分がシステム管理者である場合でも、メールの配送はなにかとトラブルや障害がつきものなので、メンバを沢山もつメーリングリストの管理は大変である。

歴史の長いメーリングリスト管理システムである *Majordomo* の論文<sup>4)</sup> では、手作業によるメーリングリスト管理の大変さをまず最初に指摘している。具体的には

- 「メーリングリストに参加したい」
- 「メーリングリストから退会したい」
- 「私はメーリングリストのメンバーか?」

といった要求や質問にいちいち応えるのは時間を消費する作業であるため、そういった単調な作業を自動化するために *Majordomo* を開発したということである。

*Majordomo* を使ったメーリングリストでは、コマンドメールと呼ばれる特殊なメールを用いて、メーリングリストのメンバー管理を行なう。コマンドメールとは、*Majordomo* のシステムが受け取って解釈するメールである。たとえば、*Majordomo* で運営されているメーリングリスト `foo@example.com` に参加するには、`Majordomo@example.com` に、本文に「subscribe foo」というメッセージが含まれたメールを送る。*Majordomo* はこのようなコマンドメールを元に、メンバーの参加や退会といった処理を自動的に行なう。*Majordomo* と似た機能を提供する *fml*<sup>10)</sup> や *Mailman*<sup>3)</sup> のようなメーリングリスト管理システムも同様の方式を採用している。

しかし、これらのシステムを導入するには、メールサーバにソフトウェアをインストールする権限が必要だったり、メールサーバ管理者としてのノウハウが必要だったりするため、一般のユーザには敷居が高い。また、メーリングリストを作るためには、専用のコマンドを実行したり設定ファイルを編集したりといった手間のかかる作業が必要である。管理者に頼めばメーリングリストを作ってもらえることもあるが、個人的な旅行の打ち合わせのためのメーリングリストを作りたい、などとほのかなか気軽には頼みづらい。メーリングリストの作成に制限を課している組織も多い。

一方、メーリングリストの一般参加者からしてみても、参加や退会のために専用のコマンドを覚えなくてはいけないのは大きな負担である。退会のためのコマンドメールを間違えてメーリングリスト本体に送ってしまうという失敗はよく見受けられる。コマンドがメーリングリストシステム毎に異なるということもユーザの負担を大きくしている要因である。

誰でも簡単にメーリングリストの作成/運用ができるようにするために、*Yahoo Groups* や *FreeML* のような Web 上のメーリングリスト作成サービスが近年よく使われている。このようなシステムを使用すると、システム管理のノウ

<http://groups.yahoo.com/>

<http://freeml.com/>

<http://www.csl.sony.co.jp/person/masui/WISS2002/>

ハウのない普通のユーザでも簡単にメーリングリストを作ることができるが、メーリングリストの作成やアカウントの登録のためにブラウザが必要になる上に、Web上のフォームにあれこれと入力する手間がかかる。

このように、メーリングリストというものは何かと手間が多く、ある程度の覚悟がないと導入は難しいと考えられていた。大きなプロジェクトの情報交換や永続的なグループ内の議論などのように、はっきりした目的があって長続きするメーリングリストであれば、手間をかけて設定を行なってでも運営する価値があるが、旅行やミーティングの打ち合わせのような一時的な用途には敷居が高く、あまり気軽には使われていなかった。

## 2.2 QuickML

QuickMLは、メーリングリストの利点を残しつつ、前述のような問題点をすべて解消した超お手軽メーリングリスト管理システムである。QuickMLでは、メーリングリストを作るためにメールサーバの設定を変更したりWebブラウザを立ち上げたりする必要はなく、普通にメールを送るだけでメーリングリストを作って活用することができる。

### メーリングリストの作成

任意の名前@quickml.comのような任意のアドレスにメールを送るだけで、新しいメーリングリストを作成できる。たとえば宴会のメーリングリストを作るにはenkai@quickml.comにメールを送ればいい。このとき、From:とCc:にリストされたアドレスがメーリングリストに登録される。

```
Subject: 来週の宴会
To: enkai@quickml.com      ← 作りたい ML のアドレス
From: satoru@example.jp   ← 自分のアドレス
Cc: masui@pitecan.com     ← 参加者リスト

突然ですが、宴会好きのメー ← 本文
リングリストを作ってみました。
```

同名のメーリングリストがすでに存在したときは、「あなたはメーリングリストのメンバーではありません」というエラーメッセージが返される。その場合は、違う名前をつけて作り直す必要がある。QuickMLではenkai@gotanda.quickml.comのようにアドレスに任意のサブドメインが入ったメーリングリストを作ることができるため、名前の衝突を容易に回避できる。

### メーリングリストへ投稿

メーリングリストへ投稿するには、メーリングリストのアドレスに普通にメールを送ればいい。メーリングリストにはメンバーだけが投稿できる。メンバー以外が投稿した場合は、「あなたはメーリングリストのメンバーではありません」というエラーメッセージが返される。

### 新しいメンバーの追加

すでに参加しているメンバーは新しいメンバーをCc:で追加できる。To:でメーリングリストをアドレスを、Cc:に新しいメンバーのアドレスを指定してメールを送ればいい。Cc:でメンバーを勧誘するというこの仕組みは、友達に声をかけて仲間を誘うという行為をモデルとしている。メンバーが追加された際には、メールの本文の先頭に

```
ML: enkai@quickml.com
新メンバー: komatsu@example.ac.jp
```

のような情報が付加される。また、メールの末尾には次のように、メンバーリストが付加される。

```
<enkai@quickml.com> のメンバー:
satoru@e...
masui@p...
komatsu@e...
```

不特定多数のメンバーが参加するメーリングリストでは、アドレスの流出が問題になることがあるため、メンバーリストのアドレスは@マークの2文字目以降を伏せる形で載せている。

### メーリングリストへの参加

すでに参加しているメンバーをCc:で指定してメールを投稿すると、メーリングリストに参加できる。参加のために送ったメールはメーリングリストに流れるため、簡単な自己紹介などを書いておくことが推奨されている。Cc:で知り合いのメンバーを指定するというこの仕組みは、友達に声をかけて仲間に入れてもらうという行為をモデルとしている。このように、QuickMLのメンバー登録は日常のグループコミュニケーションを元にして設計されている。

### メーリングリストから退会

メーリングリストに空メールを送ると、メーリングリストから退会できる。空メールを送るという操作は一種のコマンドメールと言えるが、“unsubscribe enkai@quickml.com” (Majordomo) や “# bye” (fml) といったコマンドを覚えるよりは簡単である。

### メーリングリストへの復帰

退会したメーリングリストに再びメールを送ると、メーリングリストに復帰できる。メーリングリストに復帰する場合は、メーリングリストに新規に参加する場合と異なりCc:にメンバーを指定する必要はない。

### メンバーの削除

間違ったアドレスを登録してしまったなどの理由でメンバーのアドレスを削除したい場合は、Cc:に削除したいアドレスを指定してメーリングリストに空メールを送ればいい。他のメンバーを勝手に削除するといった悪用も考えられるが、削除されたメンバーには削除された旨のメールが届くため、知らないうちに削除されていた、というトラブルは防げる。

## 2.3 QuickMLの弱点

QuickMLは手軽にメーリングリストを作って活用できる反面、いくつかの弱点も存在する。

### メーリングリストの規模

QuickMLは小さなメーリングリストを運営するには適しているが、

- メンバー数を最大100人に制限している
- メンバーが参加するたびに参加メッセージが流れるといった性質を持つため、たくさんのメンバーが参加するメーリングリストの運営には向いていない。その点、Majordomoやfmlなどのシステムは、メーリングリストに参加メッセージを流さずにメンバーを追加できるため、大規模なメーリングリストを運営するのに向いている。

### セキュリティ

Majordomoやfmlなどのシステムは、メーリングリストに参加する際に、アドレス登録の確認を行なう機能を持っている。悪意のあるユーザが他人のアドレスを勝手にメーリングリストに登録するのを防ぐためである。この仕組みは、セキュリティを高める一方で、メーリングリストへの参加の敷居が高くなるという欠点を持つ。

一方、QuickML は、手軽さを優先し、アドレス登録の確認を行なう仕組みは持たない。セキュリティの点で不安は残るものの、現在のところ悪用のトラブルは起きていない。

#### i-mode との相性

QuickML はメールを送るだけでメーリングリストを作っただけで活用できるため、携帯電話から利用すると特に便利である。しかし、2002 年現在、国内で最も普及している携帯電話である i-mode の端末は Cc: をつけてメールを送ることができないため、QuickML でメンバーの追加を行なうことができない。i-mode 用にメンバー追加を行なう Web インタフェースを作って対応することも考えているが、実現にはまだ至っていない。

また、i-mode の端末はヘッダの Reply-To: を無視するため、普通に返信を行なうと送信者にメールを送ってしまうという問題も抱えている。QuickML はメールの末尾に

```
—
ML: enikai@quickml.com
使い方: http://QuickML.com/
```

というフッタを挿入するため、ML: の右側のアドレスをカーソルで選択すれば、メーリングリストに返信が行なえる。au や H<sup>®</sup> の端末では Cc: も Reply-To: も問題なく扱えるようになる。

#### 2.4 実 装

QuickML は SMTP (Simple Mail Transfer Protocol)<sup>6)</sup> のサーバとして実装を行なった。QuickML サーバはメーリングリストへの投稿を受け付け、メンバー管理などの処理を行なったのちに、メールの配送を配送用メールサーバに依頼する。QuickML サーバではメールの配送やキューイングといった処理は一切行わず、*qmail*<sup>2)</sup> や *Postfix*<sup>9)</sup> といった既存のメールサーバに任せるといった方針をとった。

通常のメーリングリスト管理システムでは、メールが届くたびに、メールサーバの `/etc/aliases` などの設定ファイルを介して、メーリングリストシステムが外部プロセスとして実行される。一方、QuickML では、メーリングリスト管理システムがサーバとして常駐しているため、プロセス呼び出しのオーバーヘッドはなく高速に処理が行なえる。

#### メーリングリストの自動管理

QuickML の大きな特長のひとつに、メーリングリスト管理者という概念が存在しないという点がある。QuickML には、メンバーの自動削除とメーリングリストの自動消滅の仕組みが備わっているため、メーリングリスト管理の作業は一切必要ない。特別な権限を持った管理者は QuickML には存在せず、すべてのメンバーが対等の立場でメーリングリストに参加する。

#### メンバーの自動削除

アドレス消滅やネットワーク障害などの理由によりエラーメールが 5 回返ってきたアドレスはメーリングリストから自動的に削除される。ただし、単純にエラーメールを数えると流通量の多いメーリングリストでは短時間のうちに 5 通に達するため、最後にカウントしたエラーメールから 24 時間以内に届いたエラーメールはカウントしないという方針をとった。このため、一時的にメールサーバが故障した程度ではアドレスが削除される心配はない。削除の際には、メンバーを削除した旨のメッセージがメーリングリストのメンバー全員に通知される。なお、エラーメールのカウントはそのアドレスからの投稿があった時点で 0 にリセットされる。

配送不能アドレスを自動的に削除する機能のないメーリングリストシステムでは、エラーメールをチェックして配送不

能アドレスを削除するという作業を管理者が行なう必要があるが、QuickML ではその必要は一切ない。

エラーメールの自動処理は *qmail* および *Postfix* に実装されている *VERP* (Variable Envelope Return Paths)<sup>1)</sup> 拡張を利用して実現した。VERP 拡張つきで送ったメールには `envelope from` に、宛先ごとに一意の ID が付加されるため、戻ってきたエラーメールのヘッダを見れば、どのアドレスがエラーを返しているかを識別できる。

#### メーリングリストの自動消滅

すべてのメンバーが退会したメーリングリストは最後のメンバーがいなくなった時点で自動的に消滅する。QuickML にはメーリングリスト管理者という概念が存在しないため、メーリングリストの作成者が退会しても、メーリングリストはそのまま存続する。

投稿が途絶えてから 1 か月経過したメーリングリストも自動的に消滅する。自動消滅する 1 週間前に「Subject: [ ] メーリングリストもうじき消滅」というお知らせのメールがメーリングリストのメンバー全員に届けられる。その後 1 週間以内に投稿がない場合はメーリングリストが消滅する。

放置しておけばメーリングリストが自動的に消滅するというこの仕組みにより、投稿の途絶えたメーリングリストが永遠に残ってしまうという問題が回避されるため、後始末を考えずに気軽にメーリングリストを作って活用することができる。

#### サブドメインの処理

QuickML では、アドレスに任意のサブドメインを含むメーリングリストを作ることができる。この機能は、DNS のワイルドカード MX<sup>8)</sup> の仕組みを利用して実現している。ワイルドカード MX を利用すれば、あらゆるサブドメイン宛のメールをひとつの QuickML サーバで受け取れるため、アドレスにどのようなサブドメインが指定されていても、QuickML のメーリングリストとして処理できる。

ワイルドカード MX を利用するための BIND ネームサーバ用の設定例を次に示す。この例では QuickML サーバは 192.168.0.1 (`qml.example.com`) で動いていて、`example.com` のあらゆるサブドメインへのメールを QuickML サーバが受け取って処理を行なうものとしている。

```
$ORIGIN example.com.
@      IN  MX  10  qml
*      IN  MX  10  qml
qml    IN  A   192.168.0.1
       IN  MX  10  qml
```

#### エージェントとの連携

QuickML は極めてシンプルなシステムとして実現されているため、多くのメーリングリスト管理システムが提供する、投稿されたメールの保存機能すら備えていない。QuickML システムを拡張すれば、メールをファイルとして保存する程度の機能は簡単に追加できるが、このような拡張を加えていくとシステムが肥大化してしまう可能性がある。

そこで、QuickML では本体をシンプルに保ち、追加機能はメールエージェントとして実現するという方針を採用した。ここでいうエージェントとはメールアドレスを持つプログラムのことである。メーリングリストのメンバーとしてエージェントを登録すると、そのエージェントの持つ機能を利用できる。

メールをファイルとして保存するアーカイバ・エージェン

---

SMTP の MAIL FROM: で指定する差出人アドレス。メールのヘッダには現われない。

<http://www.isc.org/products/BIND/>

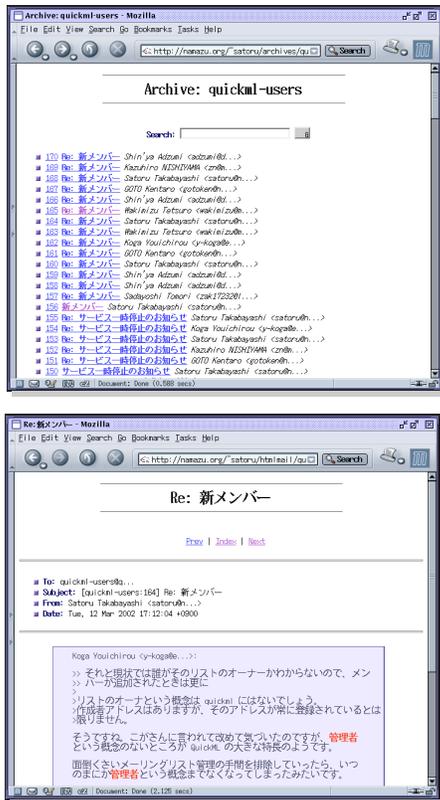


図 14 アーカイバ・エージェントが作成した Web ページ

トを作成した。アーカイバ・エージェントは、メーリングリストに投稿されたメールをファイルに保存し、メールを Web ブラウザから閲覧するための Web ページを作成する。

図 14 は quickml-users@quickml.com メーリングリストに登録したアーカイバ・エージェントが作成した Web ページである。議論の流れをスレッド表示する機能は持たないが、簡単な検索機能は備えている。スレッド表示の機能を追加した場合はエージェントのプログラムを修正するだけでよく、QuickML サーバ本体に手を入れる必要はない。

他にも、メーリングリストの自動消滅を防ぐために投稿がしばらく途絶えると延命のためのメールを送ってくる ping エージェントや、論文の締め切りが近づくともメールを送ってくるリマインダエージェントなどを作って活用している。

## 2.5 議論

我々は 2001 年 11 月に QuickML.com を立ち上げ、2 か月ほど少人数のグループでテストを行なったのちに 2002 年 1 月に一般公開を行なった。ここでは運用によって得られた経験と QuickML.com の利用状況について述べる。

### 利用経験

QuickML.com の運用を通じて、通常のメーリングリストとは異なる、QuickML ならではの使われ方がわかってきた。QuickML の特徴的な利用経験を以下にまとめる。

### 管理者権限

QuickML.com の公開当初は、管理者の権限が欲しい、細かくカスタマイズしたい、といった要望が多く寄せられた。

しかし、次第に、そういった要望はほとんど届かなくなった。これは、日常的なグループコミュニケーションにおいては、管理者の権限やカスタマイズといった機能は不要であるという QuickML のコンセプトが理解されていったためと考えられる。

### ヘビーユーザ

メーリングリストを大量に作る「ヘビーユーザ」が現われた。QuickML のヘビーユーザは話題ごとに Subject を変える代わりに、話題ごとに新しいメーリングリストを作って活用している。これは、手軽にメーリングリストを作れる QuickML ならではの活用法といえる。実際、著者は 9 か月の間に 220 のメーリングリストを作成し、1 日に受け取るメールの 31% が QuickML 経由のものであった。

### 携帯電話ユーザ

全体のユーザ 18,000 人のうち、携帯電話ユーザは 6,000 人弱と約 1/3 の割合を占めている。この結果から、QuickML はモバイル環境でのグループコミュニケーションに活用されていることがわかる。我々は、ちょっとした待ち合わせなどにも携帯電話を使ったメーリングリストを作って活用している。

### メーリングリストのアドレス

当初、我々は a@quickml.com のような極端に短い名前のメーリングリストが作られて名前の衝突が頻繁に起きるのではないかと心配していたが、実際にはそういった極端に短いアドレスが好まれる傾向は見られなかった。

一方、サブドメイン付きのメーリングリストは、それほど利用されないのではないかと予想していたが、実際には全体の 1/4 のメーリングリストがサブドメイン付きのアドレスを利用していることがわかった。これは、名前の衝突を避けるためにサブドメイン付きのアドレスが有効に活用されているためと考えられる。

### セキュリティ

QuickML は参加メンバー以外はメーリングリストに投稿できないという方針をとっているため、第三者によって広告などの迷惑メールがメーリングリストに無差別に送られるといったトラブルは起きていない。

QuickML サーバはオープンソースのソフトウェアとして公開されているため、プログラミングの技術を持ったユーザは、QuickML のシステムの動作の仕組みやセキュリティを完全に把握できる。

### 2.6 利用状況

QuickML サーバの利用ログを元に、QuickML.com の利用状況の分析を行なった。

#### ユーザ数の推移

QuickML.com のユーザ数の推移を図 15 に示す。一般公開して 9 か月が経過した 2002 年 10 月上旬時点でのユーザ数は一意のメールアドレスで数えて約 18,000 人である。グラフをみると、雑誌で QuickML が取り上げられた 4 月の半ばで傾きが上向きに変化しているものの、ユーザ数の伸びは常にほぼ線形を維持している。

#### 投稿数の推移

図 16 は QuickML.com への投稿数の累計を表すグラフである。9 か月の間に 18 万通、1 日あたり平均 700 通のメールが投稿されていることがわかる。投稿数の推移を見ると 1 日周期で波が見られる。これは、深夜から早朝にかけて投稿数が減るという、人間の生活リズムを反映した傾向を示している。深町は、メーリングリストの投稿の時系列的な解析を

指数的なユーザ数の伸びを予想としていた我々としてはやや残念な結果である。

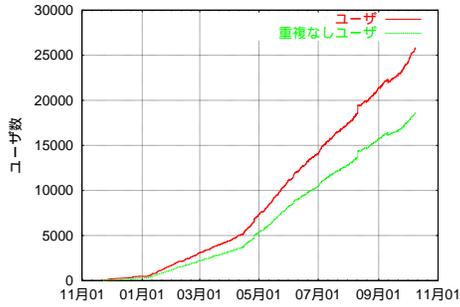


図 15 ユーザ数の推移

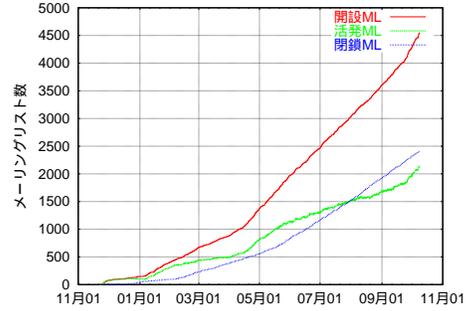


図 17 メーリングリスト数の推移

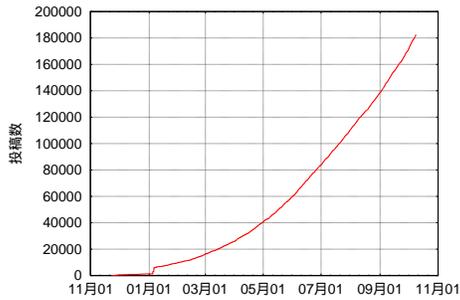


図 16 投稿数の推移

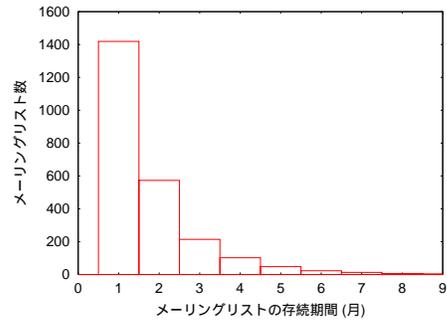
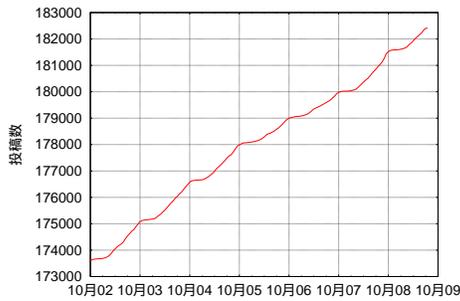


図 18 メーリングリストの存続期間の分布



行なっており<sup>5)</sup>、メーリングリストの投稿にみられるフラクタル的なふるまいのモデル化を行なっている。

### メーリングリスト数の推移

メーリングリスト数の推移を図 17 に示す。9 か月の間に、約 4,500 のメーリングリストが作られ、そのうちの 2,500 近くが消滅している。これは、1 か月投稿のなかったメーリングリストは自動消滅するという QuickML の特徴が顕著に現われた結果といえる。

### メーリングリストの存続期間の分布

消滅したメーリングリストの存続期間の分布は図 18 のようになる。1 か月で消滅するメーリングリストが大半を占め、グラフの右に進むにつれて数が少なくなる。縦軸を対数軸とするとグラフは直線となり、メーリングリストは指数分布に従って、ほぼ同じ割合 (約 28 日の半減期) で消滅していることがわかる。

### メーリングリストのメンバー数の分布

メーリングリストメンバー数の分布を図 19 に示す。点線は最小自乗法で求めた  $f(n) = 803 \times 0.82^n$  ( $n$  はメンバー数) の関数である。メンバー数の分布はほぼ指数分布に従っている。

図 20 は我々の職場で運営されている、Majordomo を用いた 180 個のメーリングリストの分布を比較したグラフである。データ数が 180 と少ないため統計的に有意ではないが、QuickML はメンバー数の少ないメーリングリストが多いのに対し、Majordomo はメンバー数が多いメーリングリストが多いという傾向が見られるようである。

### メンバー数と存続期間の相関

図 21 はメーリングリストのメンバー数と存続期間の相関を表したグラフである。色の濃淡はメーリングリストの数を表し、濃い色ほどメーリングリストの数が多い。たとえば、メンバー数が 4~7 人のメーリングリストの存続期間は 10~

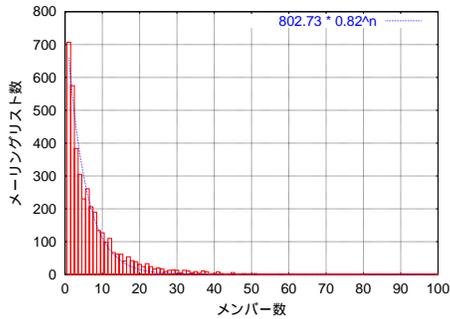


図 19 メンバー数の分布

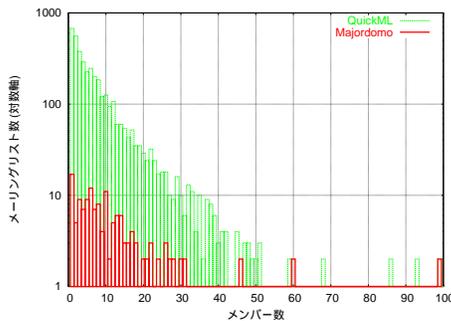


図 20 メンバー数の分布の比較

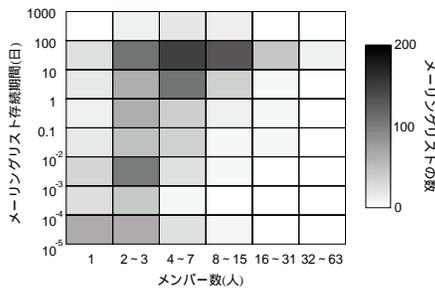


図 21 メンバー数と存続期間の相関

100日に多く分布している。メンバー数が多くなるにつれてメーリングリストの存続期間は長くなる傾向がわかる。

### 3. おわりに

インターネット上の新しいコミュニケーションツールとして、“Wiki Wiki Web”とお手軽メーリングリスト“QuickML”を紹介した。これらはまだまだ完璧なシステムではないが、機能の融合や拡張などによりさらに便利なシステムとしていきたいと考えている。たとえば、QuickMLのメッセージをWikiの掲示板データに変換したり、掲示板への投稿をQuickMLと共有することが考えられる。Wiki, 掲示板, メーリングリストはそれぞれ特徴があるのでそれらを生かしつつさらに有用なシステムを構築していきたいと考えている。

### 参考文献

- 1) Dan Bernstein. Variable envelope return paths, 1997. <http://www.jp.qmail.org/qmaildoc/RFC/RFCVERP.html>.
- 2) Dan Bernstein. qmail: the internet's mta of choice. <http://www.qmail.org/>.
- 3) Mailman Cabal. Mailman, the GNU mailing list manager. <http://www.list.org/>.
- 4) D. Brent Chapan. Majordomo: How I manage 17 mailing lists without answering “-request” mail. In *LISA VI The Systems Administration Conference*, 1992.
- 5) Ken'ichi Fukamachi. *If Fluctuations in Conservative Equations and Internet Communications*. PhD thesis, Tokyo Institute of Technology, 1996.
- 6) John C. Klensin. RFC2821: Simple mail transfer protocol, 2001.
- 7) Bo Leuf and Ward Cunningham. *The Wiki Way*. Addison-Wesley, 2001. (邦訳: 『Wiki Way』, ソフトバンク, 2002)
- 8) Paul Mockapetris. RFC1034: Domain names – concepts and facilities, 1987.
- 9) Wietse Venema. The Postfix home page, 2002. <http://www.postfix.org/>.
- 10) 深町賢一. fml パイプ. オライリー・ジャパン, 2001.
- 11) 古瀬幸広, 廣瀬克哉. インターネットが変える世界. 岩波書店, 1996.