インターネットメッセージの統合

山本和彦†

インターネットの代表的なサービスである個人宛の電子メール，メーリングリスト，ネットニュースを系統的に分類し，統括的に操作できることを示す．加えて，多目的メール（MIME）のように構造を持つメッセージを直感的に可視化や作成する方法について述べる．また，プライバシーを強化する枠組を互換性を保ちながら多目的メールと統合する書式を提案する．このように，種々のサービスを統合するインターネットメッセージの概念を提案し，これを実現するMewの実装について述べる．

An Integration of Internet Messages

KAZUHIKO YAMAMOTO†

This paper methodically categorizes electronic mail, mailing-list, and NetNews, which are now the representative services on the Internet, to give systematic operations to them. We discuss methods to view and compose structured messages such as MIME intuitively and easily without any format restrictions. Then a new scheme is explained to integrate MIME and privacy enhancement services maintaining backward compatibility to current privacy programs. We propose a new concept "Internet messages" that integrates various services up above and describe implementation of Mew, which is an interface to Internet messages.

1. はじめに

文字情報を配送するための電子メール[1]（以下メール）やネットニュース[2]（以下ニュース）は，インターネットの黎明期から着実に普及してきたサービスである．両者は文字情報の配送という同じ目的を持ち，しかも，書式が大変似ているにもかかわらず，配送システムは別々に実装されてきた[3),4)]．それに合わせる形で，ユーザインタフェースも別々に発展してきている．

似通ったサービスに，異なるインタフェースが与えられているため，ユーザは混乱していることが多い．配送システムはユーザからは隠蔽されており，配送システムに依存して両者を別のインタフェースにする本質的な理由はない．

インターネットの発達や計算機能力の向上にあわせて，ユーザの要望は多様化しつつある．そこで，画像や音声など文字以外の情報をメールで送りたいという要望を満たすために，MIME（Multipurpose Internet Mail Extentions，多目的メール）[5),6)]が規定された．しかし，従来の文字情報の交換を主目的としたインタフェースは，文字以外のオブジェクトや構造を持ったメールをうまく取り扱えていない．複雑な構造を持つMIMEメールを作成するためには，生成文法などを覚えなければならず，MIMEメールを作ることは容易でない．

インターネットの性格が，組織向けから個人向け，研究目的から商用目的に変化する中で，メールのプライバシーの保護は重要な課題となっている．本文の暗号化や電子署名により，盗聴や改竄を防止する枠組としてPGP[7)]やPEM[8)]が普及しつつある．しかし，簡単に電子署名をメールに施したり，暗号化された本文を平文のように引用できるインタフェースはほとんどない．また，MIMEのプライバシーを強化するための規格MOSS[9)]が規定されたが，このような複雑な機能を単純かつ直感的に扱えるインタフェースは存在しない．

そこで，本稿では，さまざまに分化したこれらのサービスを統合し，統一的なインタフェースを与える方法について議論する．メール，ニュース，MIME，PEM，PGPなどをインターネットメッセージの概念で統一して，複雑な書式を単純かつ直感的に扱う方法を提案する．

本稿は次のように構成される．2章では，メールとニュースを送信，回答，転送の3つの操作で統合できることを示す．MIMEメッセージの構造を単純かつ直

† 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

感的に扱う方法を，可視化と作成方法の面から3章において述べる．4章では，インターネットメッセージに対してプライバシーを強化する枠組を説明する．

以下では，MIME で定義されているフィールド名 Content-Type:, Content-Transfer-Encoding:, Content-Description:, および, Content-ID: をそれぞれ，CT:, CTE:, CD:, CI: と略す．また，我々が提案する PGP と MIME の統合方式を PGP/MIME と呼ぶ．

本稿では，慣用という言葉を，文字情報だけを含む1つのパートからなる MIME メッセージに対して用いる．つまり，慣用 MIME メッセージは，RFC822 メッセージのヘッダに，MIME-Version: と CT: text/plain (そして必要ならば CTE:) を補った MIME メッセージであり，非 MIME インタフェースでも今までどおりに取り扱える．慣用 PGP/MIME についても同様である．

2. メールとニュースの統合

この章では，メール，メーリングリスト，および，ニュースを統合する方法について述べる．まず，問題の背景を 2.1 節で説明し，これらのサービスをインターネットメッセージに統合することを 2.2 節で提案する．

2.1 メールとニュースのインタフェース

メールの書式は RFC822[1]で，配送プロトコル SMTP は RFC821[3]において 1982 年に明文化された．また，ニュースは，書式が RFC850[10]で，配送プロトコル NNTP が RFC977[4] でそれぞれ 1983 年と 1986 年に規定された．このように配送プロトコルと似通った書式は別々に定義され，配送システムも独立に実装されてきた．それに合わせる形で，非常に似通った2つのサービスに対して，別々のユーザインタフェースが独自に発達してきている．

メールとニュースを同時に扱うことができるインタフェースが若干存在するが，2つのサービスを統一的に扱えるわけではなく，単に1つのプログラムがメールとニュースへのインタフェースを別々に与えているだけである．

メールのインタフェースは，メーリングリストを扱うことができるが，これはメーリングリストの配送が SMTP で実現されているからである．しかしながら，ユーザにとってメーリングリストは，メールとニュースの中間に位置するサービスであり，インタフェースの観点からみればメールとして実装すべき本質的な理由はない．

書くための操作に注目してみると，メールやメーリングリストには，発信 (send)，返答 (reply)，メールでの転送 (forward) などがある．また，ニュースには，投稿 (post)，ニュースでの返答 (follow)，メールでの返答 (reply)，メールでの転送 (forward) などがある．

このように，メッセージを送り届けるサービスであるメールとニュース間で，操作が統一されていない．これらの系統的でない操作体系が，メールとニュースを読み書きするユーザを混乱させている．そのため，本質的にメッセージを取り扱う3つのサービスであるメール，メーリングリスト，および，ニュースの性質を明らかにし，系統的な操作を与えることが重要である．

2.2 インターネットメッセージ

我々は，メール，メーリングリスト，および，ニュースをインターネットメッセージ (以下単にメッセージ) の概念で統一することを提案し，送信 (write)，回答 (answer)，転送 (forward) の3つの操作を定義する．まず，3つのサービスの性質を考察し，次にこの3つの操作で統一前の操作がすべて実現できることを示す．

メール，メーリングリスト，ニュースを配送範囲と通信相手の観点から分類した結果を**表1** に示す．ここで，配送範囲とは実際に送信されたメッセージを読む人の数を意味し，通信相手は通信相手を特定できるかできないかを示す．

メーリングリストにおいて通信相手を特定できる場合とできない場合があるのは，配送のシステムに起因している．すなわち，メーリングリストが1段の配送システムで実現されている場合は，メーリングリストのメンバを知ることはそれほど困難ではない．しかし，多段の配送システムで配送される場合は，メンバの特定がきわめて困難になる．また，大規模なメーリングリストであれば，だれがメンバであるかを気にしない場合も多い．

表1 から明らかなように，メール，メーリングリスト，ニュースは，配送範囲においても通信相手においても段階的な位置関係にある．よって，これらをメッセージという1つの概念で統一的に取り扱ってもなんら問題はない．

次に，送信，回答，転送の3つの操作でメッセージ

表1 インターネットメッセージの分類
Table 1 Categories of Internet messages.

	メール	メーリングリスト	ニュース
配送範囲	狭い	中間	広い
通信相手	特定	特定/不特定	不特定

```
To: person@foo.ac.jp
Cc: ml@bar.com
Subject: test
From: kazu@xxx.ac.jp

本文
```

図1 メールとメーリングリストの発信
Fig. 1 Sending a mail and a mailing-list.

```
To: person@foo.ac.jp
Cc: ml@bar.com
Newsgroups: comp.baz
Subject: test
From: kazu@xxx.ac.jp

本文
```

図2 メッセージの送信
Fig. 2 Writing an Internet message.

```
To: kazu@xxx.ac.jp
Cc: person@foo.ac.jp, ml@bar.com
Newsgroups: comp.baz
Subject: Re: test
From: person@foo.ac.jp

本文
```

図3 メッセージへの回答
Fig. 3 Answering to the Internet message.

を取り扱えることを示すために，配送場への識別子という概念を導入する．具体的には，メールやメーリングリストにおけるTo: とCc:，そして，ニュースにおけるNewsgroups: などである．これらは，メッセージを送り届けたい場，すなわち，人の集合へのポインタである．

一般的なユーザは，To: とCc: を同一視することはあっても，To: とNewsgroups: を同じ識別子だと考えることはなかった．しかし，配送範囲と通信相手がメールとは異なるメーリングリストに対して，To: を配送場への識別子として代用していることから分かるように，Newsgroups: も配送場への識別子として同一視することも不自然ではない．

以下では，送信，回答，転送の3つの操作で十分であることを示す．例として，メールへの識別子にはperson@foo.ac.jpを，メーリングリストへの識別子にはml@bar.comを，ニュースへの識別子にはcomp.bazを用いる．

2.2.1 送　　信

まず，第一の操作である送信について考察する．今までメールやメーリングリストの送信の際に，**図1**のようにTo: やCc: にメールとメーリングリストの識別子を同時に書いても違和感はなく，また実際に配送も実現されていた．ならば，**図2**に示すように，ニュースへの識別子を同時に書いてもよいはずである．

インタフェースは，ユーザが作成したヘッダにTo: やCc: があれば，まずSMTPを使ってメッセージを配送し，次にNewsgroups: があればNNTPでメッセージを送信すればよい．To: やNewsgroups: の代わりに新たなフィールドを定義し，インタフェースがTo: やNewsgroups: に変換する方法も考えられるが，新たなフィールドの導入はユーザを混乱させるかもしれない．また，メールのアドレスとニュースグループ名を明確に判断することは難しい．そこで，本稿ではTo: とNewsgroups: をユーザが同一視する方法を提案する．

2.2.2 回　　答

今までニュースでは，メールによる返答とニュースによる返答は別々の操作として定義されてきた．ヘッダ中にTo: やNewsgroups: を同時に書くことが許されると，回答という1つの操作で統一できる．図2のメッセージに対する回答では，**図3**のように，From: の識別子がTo: へ，To: とCc: がCc: へ，Newsgroups: はそのままというドラフトを用意すればよい．これが自分の希望するメッセージの配送範囲を越えているなら，ユーザは適宜フィールドを削ればよい．

2.2.3 転　　送

最後に，転送について考察する．転送という操作は，第三者，正確には第三の配送場へメッセージを配送することである．転送するメッセージを指定すると，インタフェースは第三の配送場への識別子を要求する．もちろん，ここで，To:，Cc:，および，Newsgroups: を入力することができる．

今まで，メールをメールで，あるいは，ニュースをメールで転送するケースはよくあった．しかし，メッセージに対する転送という操作だけで，3つのサービス間のすべての組み合わせを網羅していることは特筆すべきことである．今まで複雑な操作をしなければ，メールをニュースへ，あるいは，ニュースの記事を別のニュースグループへ転送できなかったが，メッセージの転送という操作はプリミティブである．

Resent-To: などを用いてヘッダを変換し転送する方

式では，これまでResent-Newsgroups: というフィールドが定義されていないため，メッセージをニュースへ転送することができない．そこで本稿では，転送の書式として次章で述べるMIME形式によるカプセル化方式を推奨する．本稿では奨励しないが，メッセージの転送という概念は，行頭が"-"で始まる固定的な境界を用いてカプセル化する従来の転送[13]についても適応できる．

3. MIMEの可視化と作成方法

ここでは，MIMEメッセージの直感的な可視化と作成方法について述べる．MIMEは，その名が示すとおりメールの拡張であるが，その書式はニュースにも適応できる．そこで，以下ではMIMEで定められた構造を持つメッセージについて考察していく．MIMEは，Multipurpose Internet Message Extentionsの略であると考えてもよい．RFC1522[6]で定義されているヘッダの拡張方法は，MIME-Version: を持たないメッセージにも適合できるが，ここではMIMEの一機能と解釈して議論する．

まず，MIMEの出現の背景を3.1節で述べ，可視器と作成器の問題点をそれぞれ3.2節と3.3節で指摘する．これらの問題を解決するために，EmacsやMule上で動作するMewというプログラムを実装した．3.4節ではMewの可視器，3.5節では作成器を説明し，両者の評価を3.6節で与える．

3.1 MIMEの出現

メールの配送方法と書式は，規定時の乏しい計算機の能力と不安定な通信路を反映した形で，7ビットのASCII文字列しか配送できない規格であった．しかし，非英語圏ではそれぞれの母国語を伝播するための拡張がさかんに行われた．また，計算機の能力が劇的に向上し，通信路が安定する中で，ユーザのメールへの要望は，絵，動画，音声などの配送といったように多様化を極めた．

そこで，さまざまに地域化されたメールの架け橋となり，用途の多様化を満たし，一部残った脆弱な配送プログラムでも安定して動作する規格が必要となった．このような時代背景の中で，メールの規格が登場してから10年後の1992年に，MIMEが文献11), 12)で定められた．

MIMEは，RFC822の上位互換であり，RFC822の欠点をいくつか克服している．たとえば，非ASCII文字を安全な文字列に符号化してヘッダに挿入する枠組を提供しており，また，本文にASCII文字列だけでなく，非ASCII文字列，メッセージ，絵，動画，音声などを格納することができる．また本文にマルチパートと呼ばれる構造を持たせて，一度に複数のオブジェクトを配送することも可能である．

MIMEの登場から3年が経過したが，現時点ではMIMEがそれほど普及しているとは言いがたい．MIMEを利用しているユーザも，せいぜい日本語をヘッダに挿入している程度で，MIMEの醍醐味であるマルチパートなどはほとんど使われていないのが現状である．

ユーザはメッセージを頻繁に転送するが，転送の形式は固定的な境界でメッセージをカプセル化する旧来の書式[13]のままが多い．MIMEではこの書式を破棄し，境界に任意の文字列を用いることで再帰的な再転送を柔軟に行える形式を規定している．再転送などにMIMEを使用する需要があるにもかかわらず，MIMEが利用されていないのは，使いやすいインタフェースがないためであると考えられる．以下ではMIMEのインタフェースの問題点について考察する．

3.2 可視器の問題点

MIMEメッセージの可視化について考察すると，インタフェースの多くはマルチパートの各パートを上から順に読むことしかできない．ユーザが望んでいるのは，マルチパートの一部を自由に表示するなどの粒度の細かい操作である．

ほとんどのインタフェースは解析したMIMEメッセージの構造を保存しないため，繰り返し同じメッセージを読むと，そのつど構造解析を行う．解析に時間がかかることが分かっているユーザでも，同じメッセージを繰り返し読むときに待たされると不快感を覚える．特に解析が必要であることを知らないユーザは，強い不快感を覚えるであろう．

よって，MIMEインタフェースを実装する際には，各パートを選択的に表示できる可視器を提供し，繰り返しメッセージを読んだ場合に高速に表示することを考慮すべきである．また，新しいアプリケーションやサービスが出現したときに，柔軟に対応できなければならない．

そこで可視器に求められる機能を以下にまとめる．

- 選択性 — 各パートに対する粒度の細かい操作を提供しなければならない．
- 高速性 — 2度目以降の表示は高速でなければならない．
- 柔軟性 — 新しいアプリケーションへ対応できなければならない．

3.3 作成器の問題点

現在では，本文の構造化や文字以外の情報の配送な

どのMIMEの機能はほとんど使われていない．ユーザの多くがMIMEのマルチパートを作成しないのは，各インタフェースが独自の難解な生成文法やコマンドラインオプションをユーザに押しつけているためである．ユーザにとって複雑なインタフェースは，使いにくいばかりでなく間違いを起こしやすい．よって，簡単にMIMEに沿ったメッセージを作成できる方法が必要である．

多くのユーザがヘッダ中の非アルファベットをMIMEに従って符号化しているのは，既存のMIMEインタフェースが，文字列を自動的に符号化しヘッダに挿入するからである．これは，使いやすいインタフェースを提供すれば，MIMEを利用するユーザが増加することを実証しているといえる．

そこで，MIMEの作成器が有すべき性質を以下にまとめる．

- 容易性 — 単純な操作でMIMEメッセージを作成できなくてはならない．ユーザに複雑な操作を押しつけてはならない．
- 直感性 — 覚えにくい文法を定義してはならない．
- 隠蔽性 — MIMEの文法を理解することをユーザに要求してはならない．
- 汎用性 — MIMEの書式を制限なく作成できなければならない．

3.4 Mewの可視器

3.2節の要求を満たすために，Mewの可視器を，復号器，構造解析器，および，視覚器の部分に分けて実装した．配送後のメッセージは，復号器にかけられて元のデータ形式に戻され，Emacsのバッファに保存される．これを構造解析器が解析し，各パートの領域とデータ型を抽出して変数に保存する．視覚器は，メッセージの構造を提示するとともに，指定されたパートを可視化する．これらの全体像を図4に示す．

以下では，Mewの可視器のそれぞれの構成要素について述べるとともに，実現したさまざまなアイディアについて解説する．

3.4.1 復 号 器

ヘッダにはメッセージの配送にかかわる情報が保持されているので，メールゲートウェイで誤動作の原因となる文字を挿入すべきではない．このため非アルファベットを挿入するには，RFC1522で定義された符号方式に従って，安全な文字列に変換する．符号化された文字列はそのまま表示しても理解不可能であるから，表示の際には復号する必要がある．

しかし，メッセージを読むたびに復号を行うのは非効率であるので，復号したメッセージを一時的に保存

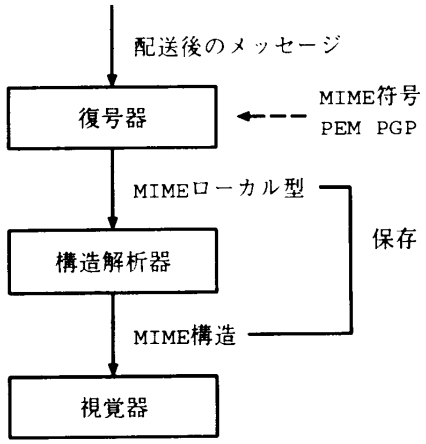

図4 Mewの可視器の内部
Fig. 4 The internal of Mew's viewer.

すべきである．コンテントヘッダについても同様である．また，CT: の1つの値であるmessage/rfc822は，ヘッダと本文の区別があるテキストを再転送するための型である．このヘッダに符号化された文字列が存在する場合には，同様に復号する必要がある．

ヘッダの議論と同じように，配送を安全にするために符号化されるパートがある．符号方式は，各パートのCTE: に格納される．符号化されたパートを表示できるようにするには，CTE: に従って復号化する必要がある．

たとえば，PEMやPGPなどで電子署名／暗号化を施されたパートは，それぞれのアプリケーションで復号する必要がある．これはヘッダでの理由と同様，復号しなければ無意味であるからである．復号前のCT: と復号したパートはデータ形式が違うため，対応するコンテントヘッダを削る必要がある．また，場合によっては，復号後のパートにさらに復号を施す必要がある．

MIMEの再帰的な構造を考慮して，MIMEメッセージを一時的に保存するための形式であるMIMEローカル型を定義した．以下にMIMEローカル型の定義を示す．

- 復号されたヘッダ，および，コンテントヘッダを持つ．また同様に，すべてのmessage/rfc822のヘッダが復号されている．
- 再帰的にすべてのパートが，CTE: に従って復号されている．
- アプリケーションで符号化されたデータが復号されており，かつ，対応するコンテントヘッダが削られている．

Mewの復号器では，CT: message/rfc822を処理する関数 m，シングルパートを復号する関数 S，およ

```
<message> = [ 'message hbeg hend
                ("message/rfc822") (CTE:) (CI:) (CD:) (Mew:) <part> ]
<single>  = [ 'single beg end (CT:) (CTE:) (CI:) (CD:) (Mew:) ]
<multi>   = [ 'multi nil nil \\
              ("multipart/mixed" |
               "multipart/digest" |
               "multipart/alternative" |
               "multipart/parallel " )
              (CTE:) (CI:) (CD:) (Mew:)
               1*<part> ]
<part>    = <message>|<single>|<multi>
```

図5 MIME構造ベクトルのBNF
Fig. 5 BNF for MIME syntax vector.

び，マルチパートを復号する関数 M を実装している．MIMEメッセージは，まず関数 m に渡され，CT: がmessage/rfc822であれば関数 m，multipartであれば関数 M，それ以外であれば関数 S を呼び出す．関数 m，M，S は互いに再帰的に呼びあい，メッセージを先順に復号化しながら，MIMEローカル型に変換する．

復号した後のパートをさらに復号する可能性があるため，復号器の処理時間はメッセージの構造に依存する．MewはEmacs上で実装されているので，復号器で処理して得られたMIMEローカル型はEmacsのバッファに保存される．

3.4.2 文字集合の取扱い

MIMEでは，テキストである1つのパートに対して1つの（符号化を含む）文字集合を与えることができる．残念ながら，1つのテキストに複数の文字集合を指定できないので，MIMEは真の多国語化であるとは言えない．1つのテキストに複数の文字集合を取り込むには，ISO 2022 JP 2[14]などの複数の文字集合を符号化できる文字集合を使用する必要がある．

MIMEの可視器では，文字集合の取扱いについて注意が必要である．RFC822では，本文の文字集合をUS-ASCIIに限定しているが，実際にはさまざまな言語を運搬するように地域化されていることは前述のとおりである．

地域化されたRFC822メッセージがこれほど普及した現在，これらのRFC822に沿っていない地域化を禁止することはほとんど不可能である．幸いにもMIMEはRFC822に対してなんら言及していない．つまり，MIME-Version: を持たないRFC822メッセージに対しては，これまでの地域化が有効である．

そこで，Mewでは以下のような規則に従って，文字集合を決定する．

(1) ユーザにデフォルトの文字集合を選べるようにしておく．
(2) MIME-Version: がないメッセージの場合は，本文をデフォルトの文字集合として扱う．
(3) MIME-Version: があり，CT: がない場合は，US-ASCIIとして扱う．
(4) MIME-Version: とCT: がある場合は，CT: に指示された文字集合を利用する．

Muleでは，1つのバッファに複数の文字集合を収納できるように，内部表現を定義している[15]．よって，Muleでは複数のパートにそれぞれ別の文字集合を持つメッセージを1つのバッファに保存できる．Mewの復号器は，まず内部表現に変換せずにメッセージを読み込む．そして，各パートやフィールドに対して，上記のルールに従って文字集合を決定し，Muleの内部表現に変換する．もちろん，ユーザはデフォルトの文字集合を自由に設定できる．

3.4.3 構造解析器

復号器で復号化する際に，MIMEメッセージの構造を取り出すことは，データの位置が復号化によって変化するため困難である．そこで，Mewでは構造解析器がMIMEローカル型より各パートの領域やデータ型などのMIME構造を抽出し，MIMEローカル型が格納されているバッファに固有な変数として保存する．

MewではMIME構造を表現するために，MIME構造ベクトルを図5のようにBNFを使い定義した．<>はBNFの非終端記号である．シングルパートは<single>で表現され，再帰を許す枠組であるmultipartやmessage/rfc822は，シングルパートとは別にそれぞれ<multi>と<message>で定義されている．[]はLispのベクトル，()はリストを表す．

ベクトル内の第1要素は，どの種の非終端記号であるかを示すシンボルである．第2，第3要素は対応する領域の始めと終わりを表す．<message>の場合はヘッダ領域，<single>の場合はそのパートの領域を

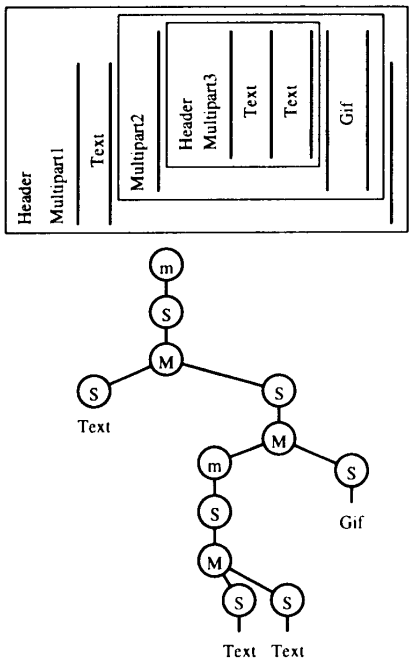

図6 構造解析器よるメッセージの解析例
Fig. 6 An example of MIME syntax analysis.

示す．第4，5，6，7要素は，MIMEが定義しているCT:，CTE:，CI:，CD: を格納する．これらは，複数のパラメータを持つ場合があるので，長さに制限のないリストで表現されている．第8要素は，Mewが特殊なアプリケーションに対して独自に使う情報であるが，これは4.4節で説明する．

復号器と同様，構造解析器は関数 m，M，S からなり，MIME構造ベクトルを構築する．構造解析器における関数の呼びあいの例を**図6**に示す．これは，第1パートがテキスト，第2パートがマルチパートであるメッセージの例である．マルチパートの第1パートは，2つのテキストからなるメッセージであり，第2パートはGIFのイメージが収納されている．MIME構造解析器の解析時間は，ほぼMIMEローカル型の長さに比例する．

3.4.4 視覚器

ユーザが表示コマンドを入力した場合，Mewは現在のメッセージをバッファ内に読み込み，復号器と構造解析器にかける．そして，もし，メッセージがシングルパートの場合は，視覚器が即座にCT: の値に対応した表示を行う．たとえば，text/plainは他のEmacsウインドウに表示し，application/postscriptはghostviewなどのコマンドを起動する．Mewでは，CT: の値に応じて呼び出すコマンドやLisp関数を，ユーザが自由に設定できる．

もし，現在のメッセージがマルチパートの場合は，視覚器は**図7**で示すように，MIME構造を可視化する．この状態でユーザはカーソルを動かし表示のコマンドを入力することで，各パートを選択的に表示できる．

しかし，MIME構造ベクトルを単純に可視化すると問題が生じる．たとえば，MIMEメッセージの中に直接CT: message/rfc822がきており，そのメッセージがマルチパートであった場合を考えてみる．Mewの可視化の枠組では，マルチパートに含まれる各パートを列挙することはできても，直接含まれているシングルパートの構造を表示することはできない．この例では，直接のmessage/rfc822を表示できず，その結果，内部のマルチパートも表示できない．

そこで，Mewでは，「シングルパートと1つしかパートがないマルチパートは同値である」という方針を採用した．つまり，シングルパートと1つしかパートがないマルチパートは，適宜変換できる．Mewでは，可視化の都合上，message/rfc822に直接message/rfc822が含まれている場合，このシングルパートを1つしかパートがないマルチパートとして扱う．他のCT: については，シングルパートとして処理する．

逆に1つしかパートがないマルチパートにおいて，そのシングルパートのCT: がmessage/rfc822以外であれば，シングルパートとして取り扱う方が1つの動作でパートを表示できるため便利である．しかし，Mewでは，1つしかパートがないマルチパートをマルチパートとして扱う．この理由は4.4節で述べる．

3.5 Mewの作成器

ユーザが交換するメッセージの多くは，慣用MIMEである．そこで，Mewの作成器では，従来の慣用MIMEメッセージを作成するための操作を極力排除した方法と複雑なMIMEメッセージを作成できる方法の2つを提供している．複雑なMIMEメッセージについても，操作数は若干増えるが単純かつ直感的に作成できる．

3.5.1 慣用MIMEメッセージの作成方法

慣用MIMEメッセージは，本文がテキストであるため文字集合を正しくMIMEの書式として与える必要がある．Emacs上でMewが動作する場合，ドラフトバッファに7ビット文字しかない場合はUS-ASCII，8ビット文字がある場合はISO 8859 1であると推測する．また，Muleの場合は，内部表現から文字集合を推測する．この文字集合の推測は，メッセージを送信する直前に自動的に行われ，CT: のパラメータとして挿入される．よって，ユーザは自分で文字コードを正しく記述するといった煩わしい作業をしなくてよい．

```
46 M12/06 Kazuhiko Yamamoto  An example of MIME
        1       Text/Plain
        2.1     message/rfc822
                2.1.1   Text/Plain
                2.1.2   Text/Plain      A patch of Mew
        2.2     image/gif       猫の絵
```

図7 MIME構造ベクトルの視覚化
Fig. 7 Visualization of MIME syntax vector.

```
<file> ::= ( filename CT: CD: mark receivers)
<directory>  ::= ( ( dirname CT: CD: mark receivers) 0*<file> )
```

図8 作成器リストのBNF
Fig. 8 BNF for Mew's composer list.

```
To: person@foo.ac.jp
Subject: 猫
Mime-Version: 1.0
----
僕が飼っている猫です.

  ---- multipart --
    0   1/              Multipart/Mixed
    1     00CoverPage     Text/Plain
    2.0   dir/            Multipart/Mixed
B   2.1     cat.gif         image/gif           "A pretty cat"
Q   2.2     cat.ps          application/postsc..
  ---- multipart ----
```

図9 複雑なMIMEメッセージの作成
Fig. 9 Composing a complex MIME message.

3.5.2 マルチパートの作成方法

Mewでは直感的かつ汎用的に複雑なMIMEメッセージを作成できるように，ファイル構造をMIME構造へ射影する方式をとっている．つまり，ディレクトリはマルチパートに，ファイルはシングルパートに変換される．計算機をある程度使った経験があるユーザであれば，ファイルの複製，移動，リンク，および，削除や，拡張子などのファイル名の慣習は理解できている．また，ファイル構造は階層化することが容易である．

各ファイルのCT: はファイル名から推測できる．たとえば，拡張子 ".ps" を持つファイルは application/postscript であるし，"11" のようにファイル名が数字であるファイルは message/rfc822 であると推測できる．

しかし，単純にファイル構造をMIMEの書式に変換したのでは，たとえばCD: のような付加的な情報を織り込むことができない．そこで，Mewでは，図8のような作成器リストを保持しており，ユーザの動作に合わせて内容を更新する．CT: やCD: は，ファイル名と独立に保持されているので，いつでも変更できる．たとえば，ディレクトリのCT: はデフォルトで multipart/mixed が選ばれるが，いつでも multipart/alternative などに変えられる．マークや復号者の指定方法に関しては，4.3節で解説する．

Mewでマルパートを作成した例を図9に示す．ドラフトバッファでマルチパートを作成するためのコマンドを入力すると，本文の下に文字列 "`---- multipart --`" と "`---- multipart ----`" で囲まれたマルチパート部が追加される．マルチパート部では，ファイルの複製，移動，リンク，削除や，ディレクトリの生成，消去などのコマンドが各キーに割り当てられている．

図9において，第1列は，符号化を示すマークが表示されている．これは，作成器リストのマークに対応している．base64なら "B"，quoted-printableなら "Q" が表示される．第2列は各パートの番号，第3列はファイル名である．ディレクトリ，つまり，マルチパートの番号はすべて "0" で終わる．番号0に対応するディレクトリ1は，このメッセージ全体のルートディ

レクトリであり，セキュリティ保全の理由からユーザだけが読める場所に作られる．2行目の00CoverPageというファイルは，ドラフトバッファで記述しているテキストを示している．第4列はCT:，第5列はCD:を表示する．

図9は，全体がマルチパートであり，その第1パートがドラフトバッファで作成されたテキスト，第2パートがマルチパートになっている例である．第2パートの中には，GIFファイルとPostScriptファイルがコピーされており，GIFファイルには“A pretty cat”という説明が付加されている．

送信の直前にMewは以下のようにして，ファイル構造をMIMEの書式に変換する．まず，ドラフトバッファから中央のテキストを切り出し，ファイル“00CoverPage”に保存して，ファイル構造を完成させる．また，マルチパート部を削除しヘッダだけを残す．そして，作成器リストに従ってファイル構造を後順に移動しながら，各ファイルを処理する．対象がディレクトリの場合は，マルチパートの境界が生成され，この境界で区切りながら，ディレクトリ内にあるファイルを処理してマルチパートを完成させる．

ファイルは一時的なバッファにいったん読み込まれ，テキストであれば文字集合が推測される．また，マークに従って，適切な符号化が施される．ファイルの処理が完成したら，ドラフトバッファにコピーされる．このようにMewでは，MIMEの書式を理解していなくても，複雑なMIMEメッセージを最小限の作業で作成できる．

3.6 可視器と作成器の評価

可視器では，視覚器がマルチパートの構造をEmacsのバッファに表示し，ユーザがカーソルを移動させて自由にパートを選択できるため，高い選択性を持っている．また，復号器が復号したメッセージとそのメッセージから構造解析器がMIME構造を抽出してバッファに保存するため，2度目以降の表示は高速である．このキャッシュ用のバッファはLRU方式で管理され，ユーザは大きさを自由に定義できる．また，新たなアプリケーションが現れた場合は，新たなデータ方式からMIMEローカル型を生成できるように復号器を改良するだけでよい．このため，構造解析器や視覚器には何ら影響を与えないという柔軟性を持っている．

作成器では，ファイルの操作という単純な操作だけで複雑なMIMEメッセージを作成できる．ユーザはMIMEの書式やMIMEメッセージを作成するための特別な文法を理解する必要はない．よって，容易性，直感性，隠蔽性を保有していると言える．また，ファイル構造は制限なく入れ子構造をとれるので，作成されるMIMEメッセージにも制限がなく，汎用性を達成している．

4. プライバシーの強化

メッセージは配送経路において，改竄や盗聴の脅威にさらされており，プライバシーを強化することが強く望まれている．ここでは，プライバシーを強化する枠組であるPGPをMIMEに統合する書式やインタフェースについて議論する．

PGPとMIMEを統合する背景と書式を，それぞれ4.1節と4.2節で述べる．4.3節では，作成方法に符号化とマークという概念を導入し，3.4節で述べた方法を発展させる．可視化については，3.5節で述べた方法に従って，MOSSやPGP/MIMEを新しいアプリケーションととらえ拡張する方法を4.4節で述べる．最後に，4.5節でPGP/MIMEの評価を与える．

4.1 PGPとMIME

メッセージに対して，本文の秘匿性，ユーザ認証，本文の完全性，および，内容の非否認性を与える枠組としてPGPやPEMがある．しかしながら，今まで共通の規則を規定していなかったため，PGPやPEMでは文字情報以外を配送することができなかった．また，MIMEはPGPやPEMとは独立に発達して来たので，プライバシーを保護する機能はない．そこで，PGPやPEMをMIMEと統合し，さまざまなオブジェクトを安全に配送するサービスが強く望まれている．

IETFでは，PEMとMIMEを統合する方式を議論し，PEMで保護するオブジェクトとPEMの制御情報をMIMEのマルチパートに射影する方式MOSSを採択した．しかし，MIMEを対象としていないPEMに対して，MIMEの書式を適応したために，結果としてMOSSはPEMと互換性がない．PEMは，輸出制限などの規制によってそれほど普及していないので，互換性を重視しないことはそれほど悪影響を及ぼさないかもしれない．

しかしながら，PGPは世界中に広く普及しており，MOSSのように互換性を犠牲にすることは得策ではない．そこで，PGPと互換性を保ちつつMIMEと統合する方式PGP/MIMEを次節で提案する．

4.2 PGP/MIMEの書式

PGP/MIMEの書式では，PGPとの互換性を保つことと，PGP/MIMEメッセージを書式に制限なく作成できることが重要である．そこで，既存のPGPオブジェクトをMIMEに埋め込む枠組とMIMEオブ

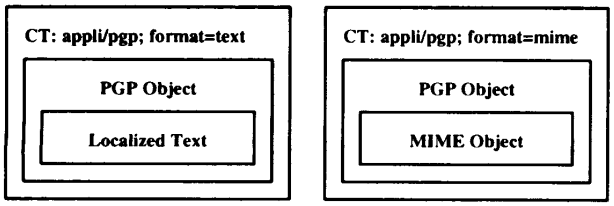

図 10 PGP オブジェクトと MIME オブジェクトの関係
Fig. 10 Structure of PGP objects and MIME objects.

ジェクトを PGP に収納する枠組をそれぞれ定義して，再帰的な構造をとれるようにしなければならない．

我々は，既存の PGP オブジェクトを MIME に埋め込む書式として，CT: application/pgp を定義した．また，その PGP オブジェクトがどんなオブジェクトを収納しているか示すために，format というパラメータを用意した．format の引数が text であれば PGP オブジェクトの中身は，地域化されたテキストである．地域化されたテキストとは，ユーザが生活する地域で主に利用される文字コードからなるテキストである．たとえば，日本では ISO 2022 JP のことを意味する．また，引数が mime であれば，PGP には MIME オブジェクトが格納されている．

図 10 の左側に format＝text，右側に format＝mime の書式を図示する．再帰的な構造を許すためには，PGP の書式自体に中身を判別するパラメータが必要であると考えるかもしれない．しかし，既存の PGP を変更しないことが前提であるので，PGP の書式に新たなパラメータを定義することはできない．

PGP プログラムは MIME インタフェースから呼び出され，署名の検証などの処理後にデータを返す．このデータをテキストとして扱うか，MIME としてさらに処理を行うかは，MIME インタフェースの仕事である．よって，呼び出す前に PGP オブジェクトの中身が何であるか分かっていればよい．また，PGP プログラムは，暗号化，電子署名，電子署名後暗号化というサービスを自動判別するので，MIME にとってこれらのサービスを示すパラメータは不要である．

我々の書式を利用すれば，テキスト以外のオブジェクト，マルチパートの中のいくつかのシングルパート，および，マルチパート全体に暗号化，電子署名，電子署名後暗号化という処理を施すことができる．書式や符号化の方式，および，行末の正規化の詳細については文献 16) を参照されたい．我々が提案する書式は，PEM に対しても適応できるが，MOSS が規定された現在これ以上深く議論することはしない．しかし，4.3 節と 4.4 節で述べる方式は，書式とは独立であるので MOSS にも適応できる．

4.3 PGP/MIME の作成器

PGP/MIME の作成器には，3.3 節で示した容易性，直感性，隠蔽性，そして，汎用性が必要である．そこで，Mew の作成器を PGP/MIME が取り扱えるように拡張しなければならない．Mew の作成器では，マルチパート部の第一カラムに MIME の符号方式をマークで示している．また，PGP での暗号化，電子署名，および，電子署名後暗号化は，一種の符号化であると考えてよい．そこで，Mew では MIME の符号化と PGP での処理を，符号化という概念で統一しマークで表示する方法を採択した．

ユーザは，PGP で処理したいパートに PGP の動作を示すマークを付ける．このコマンドはキーに割り当てられており，ユーザがコマンドを入力すると現在のマークを上書きする．PGP で暗号化するには復号者を指定しなければならない．そこでユーザが暗号化，あるいは，電子署名後暗号化を指定すると，Mew はミニバッファから復号者の識別子の入力をうながす．入力された復号者の識別子は，行の長さの制約のため CD: の部分を上書きする形で表示される．これらのマークはいつでも取り消すことができる．

図 11 に PGP のマークを付けた例を示す．パート 1 のテキストには，電子署名後暗号化を表すマーク "**PSE**" が付けられている．パート 2.0 のディレクトリには，暗号化のマーク "**PE**" が表示されている．ディレクトリへのマークであるから，実際にはマルチパート全体が暗号化される．また，パート 1 とパート 2.0 ともに復号者として kazu が指定されているのが分かる．パート 2.1 の GIF ファイルには，電子署名を表すマーク "**PS**" が指定されている．

Mew の作成器は，メッセージの送信の直前に，ファイル構造を後順にたどって PGP/MIME メッセージを作成する．PGP 用のマークが指定されているファイルは，PGP で処理された後に，ドラフトバッファに挿入される．また，ディレクトリのマークが処理されるのは，マルチパート全体が完成した後である．

Mew は，電子署名や電子署名後の暗号化を表すマークを処理するごとに，ユーザに対して秘密鍵を復号化するためのパスフレーズの入力をうながす．ユーザが入力した文字は画面に表示されない．PGP に対して環境変数やコマンドラインオプションでパスフレーズを与えると盗聴される恐れがあるので，入力されたパスフレーズは対話的に PGP に渡される．盗聴を防止するため Mew は，一度入力されたパスフレーズを再利用することはない．

マークを拡張したマルチパート部を使えば，書式に

```
To: kazu
Subject: 猫
Mime-Version: 1.0
----
僕が飼っている猫です.

---- multipart --
    0   1/              Multipart/Mixed
PSE 1     00CoverPage     Text/Plain              "kazu"
PE  2.0   dir/            Multipart/Mixed         "kazu"
PS  2.1     cat.gif         image/gif             "A pretty cat"
Q   2.2     cat.ps          application/postsc..
---- multipart ----
```

図 11 PGP/MIME メッセージの作成
Fig. 11 Composing a PGP/MIME message.

制限なく直感的に PGP/MIME メッセージを作成できるが，複数のコマンドを入力する必要がある．ユーザが一番多く交換するメッセージは，文字情報であり，PGP においても慣用 PGP/MIME メッセージの利用が多い．よって，慣用 PGP/MIME メッセージに対して，作成の際に入力しなければならないコマンドをできるだけ少なくすることが望ましい．そこで，Mew では慣用 PGP/MIME メッセージを作成するために，上記とは違う方法を提供している．

ドラフトバッファでは，慣用 PGP/MIME メッセージを作成する 3 つのコマンドがキーに割り当てられている．これらのコマンドは，本文を切り出して PGP に渡し結果をドラフトバッファに再挿入し，ヘッダに CT:application/pgp を補う．

電子署名，および，電子署名後暗号化用のコマンドを実行するとパスフレーズの入力を要求される．入力されたパスフレーズは，対話的に PGP に送られる．また，慣用 PGP/MIME メッセージでは，受信者と復号者が同一であるので，暗号化，および，電子署名後暗号化用のコマンドは，自動的に To: や Cc: から受信者を切り出し復号者の識別子として PGP に指定する．

4.4 PGP/MIME の可視器

PGP や MOSS における暗号化，電子署名，および，電子署名後暗号化は，一種の符号化である．よって，MIME の符号化と同様に，ユーザが読む際には復号する必要がある．PGP/MIME メッセージの可視器においても，3.2 節で示した選択性と高速性が要求される．また，柔軟性を示すためには，MIME の可視器を PGP/MIME へ最小限の改良で対応できなければならない．

ローカルのファイルシステムを自分だけが読めることを保証するのは困難である．そこで，プライバシー保護のためには，配送後の PGP/MIME メッセージを PGP で保護されたままの書式で保存する必要がある．しかしながら，ユーザは暗号文をあたかも平文のように，可視化したり回答の際に引用したりしたいと望んでいる．また，自動的に復号化された暗号文が実際に暗号化されていたことや，電子署名の検証の結果を，ユーザは各パートごとに知りたいと思うであろう．

よって，PGP/MIME の可視器では，MIME の可視器への要望に加えて，以下の項目が要求される．

- 透過性： 暗号文を平文のように取り扱えなければならない．しかし，ディスク上には配送されたままの形で保存されなければならない．
- 知覚性： 各パートの検証結果をユーザに通知できなければならない．

Mew の可視器で PGP/MIME メッセージを取り扱うためには，復号器を改良するだけでよい．Mew の復号器は，CT: が application/pgp であるパートをバッファから切り出し，コンテントヘッダを取り除いてから，サブプロセスの PGP に渡す．PGP からパスフレーズを要求される場合は，ユーザにパスフレーズの入力をうながす．そして，結果をバッファに無変換で再挿入する．もし，format の引数が text であれば，デフォルトの文字集合へ変換する．また，mime であれば，さらにそのパートに対して復号化を行う．

復号器は，PGP からの結果を受け取ると，暗号化されていた場合はその旨を，また，電子署名が施されていた場合は検証の結果をオブジェクト内のコンテントヘッダに X-Mew: フィールドとして付け足す．慣用 PGP/MIME には，コンテントヘッダはないが，メッセージのヘッダがその代替を果たす．X-Mew: フィールドは，PGP などのように特殊なアプリケーションのために用意されたフィールドであり，Mew の構造解析器は CT: などと同様に MIME の構造として抽出

```
X-Mew: <1> PGP decrypted. Good PGP sign "<person@foo.ac.jp>" COMPLETE
X-Mew: <2> PGP decrypted.
X-Mew: <2.1> Good PGP sign "<person@foo.ac.jp>" COMPLETE
```

図12 PGPの検証結果の表示
Fig. 12 Visualization of PGP warnings.

```
ftp://ftp.aist-nara.ac.jp/pub/elisp/Mew/mew-current.tar.gz
```

図13 Mewの入手先
Fig. 13 URL of Mew.

する．また，視覚器は，メッセージのヘッダが表示される際に，集めたX-Mew: フィールドをヘッダに追加して，ユーザに検証結果を伝える．

図11で作成されたPGP/MIMEメッセージに対する検証結果の表示例を**図12**に示す．"<>"中の数字がパート番号を表している．検証の結果は，改竄が行われていないか，だれの署名か，および，その公開鍵の有効性の値などが表示される．また，暗号化されていたパートに対しては，復号化されたことを示唆する．

3.4.4項において，1つしかパートがないマルチパートをシングルパートとしてではなく，そのままマルチパートとして扱うと述べた．これは，X-Mew: の情報を保存するためである．

たとえば，中身のシングルパートが署名され，マルチパート全体が暗号化されているメッセージを考えてみる．もし，シングルパートとして扱おうとすると，MIME構造ベクトルからマルチパートに対応する情報を削り，中身のシングルパートだけに置き換えなければならない．するとマルチパートが暗号化されていたという情報が失われてしまう．このような情報の欠損を防ぐために，Mewでは1つしかパートがないマルチパートもそのままマルチパートとして扱う方法を採択した．

悪意をもった者は，受信者を欺くためにあらかじめ不正なX-Mew: フィールドを挿入するかもしれない．そこで，Mewはあらかじめ X-Mew: フィールドを削ることによって，復号器が生成したX-Mew: フィールドのみを構造解析器が集めることを保証する．

4.5 PGP/MIMEの評価

Mewでは，暗号化されたパートは復号化されてキャッシュ用のバッファに保存されるため，あたかも平文のように表示したり引用したりすることができる．また，各パートへの検証結果をコンテントヘッダに保存し，ヘッダに一度に表示するというアイディアで，Mewは警告をユーザへ通知することに成功している．よって，Mewの可視器は，4.4節で掲げた透過性や知覚性を満たしている．

Mewの作成器は，MIMEの作成器にマークを導入することで，ユーザに直感的かつ単純にインタフェースを与えることに成功している．また，ディレクトリにマークを付けたり，復号者の識別子を各パートごとに指定できるなど，作成できるPGP/MIMEの書式には制限がない．

5. 実装状況

現在Mewは，Emacs version 18と19，Xemacs，および，Mule version 1と2上で安定して動作する．Mewのほとんどは，Emacs Lispで記述されており，総ステップ数は11,000を超えている．ニュースを除くすべての機能は，本稿で述べなかった他の便利な機能とともに実装済である．今後は，ニュースをサポートして，完全なインターネットメッセージの統合をめざす予定である．Mewは，GNU Public License 2に従って配布されており，**図13**に示す場所から入手できる．

6. おわりに

本稿では，電子メール，メーリングリスト，ニュース，MIME，PGPなどのサービスをインターネットメッセージの概念で統一し，直感的かつ単純なインタフェースをユーザに与える方法について議論した．

インタフェースを配送プロトコルから独立させれば，To:，Cc:，および，Newsgroups: を配送場へのポインタとして同一視できる．また，これまで繁雑でユーザを混乱させていた操作は，インターネットメッセージに対する送信，回答，および，転送という3つの操作ですべて網羅できる．

我々はインターネットメッセージのインタフェースとして，EmacsやMule上で稼働するMewを実装した．Mewの可視器は，復号器，構造解析器，および，視覚器によって，各パートに対する細かな操作を可能とし，2回目以降の表示を高速に行い，そして，新し

いアプリケーションに柔軟に対応できる．また，Mewの作成器は，ファイル構造をMIMEの書式に変換するというアイデアによって，MIMEや作成文法を覚えることをユーザに押しつけることなく，単純な操作で複雑なMIMEメッセージを制限なく作成できる．

さらに，既存のPGPと互換性を保ちながら，MIMEと統合する方式について提案した．Mewの可視器は，複雑なPGP/MIMEをあたかも平文のように表示し，また，各パートに対するPGPの警告をユーザに通知できる．PGPの暗号化などをMIMEの符号化と統一しマークで表示することによって，Mewの作成器は直感的かつ単純に制限なくPGP/MIMEメッセージを作成できるインタフェースを提供することに成功している．

謝辞 この研究にあたって歌代和正氏，門林雄基氏，櫻井三子氏，佐野晋氏，および，中村素典氏と有益な議論を行う機会を得た．ここに名前を記して感謝する．

参考文献

1) Crocker, D.: Standard for the Format of ARPA Internet Text Messages, RFC822 (1982).
2) Horton, M. and Adams, R.: Standard for Interchange of USENET Messages, RFC1036 (1987).
3) Postel, J.: Simple Mail Transfer Protocol, RFC821 (1982).
4) Kantor, B. and Lapsley, P.: Network News Transfer Protocol, RFC977 (1986).
5) Borenstein, N. and Freed, N.: MIME (Multipurpose Internet Mail Extensions) Part One, RFC1521 (1993).
6) Moore, K.: MIME (Multipurpose Internet Mail Extensions) Part Two, RFC1522 (1993).
7) Zimmermann, P.: The Official PGP User's Guide, MIT Press (1995).
8) Linn, J.: Privacy Enhancement for Internet Electronic Mail, RFC1421 (1993).
9) Crocker, S., Freed, N., Galvin, J. and Murphy, S.: MIME Object Security Services, RFC1848 (1995).
10) Horton, M.: Standard for Interchange of USENET Messages, RFC850 (1983).
11) Borenstein, N. and Freed, N.: MIME (Multipurpose Internet Mail Extensions), RFC1341 (1992).
12) Moore, K.: Representation of Non-ASCII Text in Internet Message Headers, RFC1342 (1992).
13) Rose, M. and Stefferud, E.: Proposed Standard for Message Encapsulation, RFC934 (1985).
14) Ohta, M. and Handa, K.: ISO-2022-JP-2: Multilingual Extension of ISO-2022-JP, RFC1554 (1993).
15) Nishikimi, M., Handa, K. and Tomura, S.: Mule: MULtilingual Enhancement to GNU Emacs, *Proc. INET '93*, pp.GAB-1-GAB-9 (1993).
16) Yamamoto, K.: An Integration of PGP and MIME, *The Internet Society 1996 Symposium on Network and Distributed System Security*, pp.17-24 (1996).

(平成7年 9 月 25 日受付)
(平成7年 12 月 8 日採録)

山本 和彦（正会員）

昭和45年生．平成6年3月九州大学大学院工学研究科情報工学専攻修了．同年奈良先端科学技術大学院大学助手就任．インターネットにおける経路制御，セキュリティ，および，次世代IPの研究に従事．