システムソフトウエアと
オペレーティング・システム　60－5
マルチメディア通信と分散処理　61－5
（１９９３．　７．　８）

マルチメディアメール・ニュースの表示システムの設計および実装

若林 進†　下條 真司‡　西尾 章治郎†

†大阪大学工学部情報システム工学科
‡大阪大学大型計算機センター

文字情報しか扱えなかった従来の電子メール・電子ニュースに対して，これをマルチメディア情報を扱えるように拡張した規格としてMIMEがある．ところが，現在MIMEに対応した電子メール・電子ニュースリーダは，規格の提唱者によるサンプル的なものしかなく，画像の表示をテキストと別のウインドウに表示するなど実用性に欠けている．本稿ではMIMEに対応したマルチメディア電子メール・電子ニュースリーダ'UMIME'の設計および実装について述べる．実装にあたっては，X Window System上のオブジェクト指向ユーザインタフェースツールキットであるET++を使用し，その機能を用いて文字と画像を一つのウィンドウ中に混在させて表示することを可能にした．また，音声データについてはテキスト中にアイコンとして表示し，それをクリックすることにより再生する．

Design and Implementation of a Multimedia Mail/News Reader

Susumu WAKABAYASHI†, Shinji SHIMOJO‡, and Shojiro NISHIO†

†Dept. of Information Systems Engineering, Faculty of Engineering, Osaka University
‡Computation Center, Osaka University

Recently, multimedia electric mail and news have become attractive media for human communication through computer networks, and the standardization for such multimedia mail and news system has been advanced with the open document *MIME*. However, currently available MIME readers for displaying the data specified under the MIME protocol have several deficiencies. For instance, *metamail*, i.e. one of the most popular MINE readers, displays each medium of a multimedia document in a separate window. In this paper, we discuss the design and implementation issues of an advanced MIME reader called *UMIME* (Useful MIME) which resolves such problems. The UMIME is implemented employing object-oriented user interface toolkit *ET++* as well as *X Window system*, and has the function to display multifont characters and illustrate both text and graphic data in the same window. Sound data is embedded in *LipIcon* and can be played back by clicking it.

1 序論

近年ワークステーションが普及し，コンピュータの処理速度の向上，2次記憶装置の大容量化，主記憶の大容量化といったハードウェアの進歩に伴い，コンピュータ上で大容量データを扱うことが可能となってきた．また，ワークステーションに対応したビデオボードの低価格化が進んで比較的容易に入手することができるようになってきており，音声処理のためのDSP(Digtal Signal Processer) などはワークステーションに標準で装備されてきている．このように個々のコンピュータにおいて，従来までの文字情報だけではなく，画像，音声などのマルチメディア情報を扱うことが可能になってきている．さらに，これらのハードウェアの性能向上に加え，データ圧縮技術についても，静止画像のためのJPEG(Joint Photographic Experts Group)，動画像のためのMPEG(Moving Pictures Experts Group) に代表される画像圧縮方式や，ADPCM方式などによる音声符号化などが実用化されてきている．その一方で，ローカルエリアネットワーク (LAN) を相互に接続した広域ネットワークが高速化されてきており，ネットワークを介して大量のマルチメディア情報をやりとりする基盤ができつつある．

これらマルチメディア技術の進歩に伴い，大学・研究機関・企業に世界的に広がっている電子メール，電子ニュースシステム上でも，これまでの文字だけによるコミュニケーションから，静止画像，音声，動画像といったマルチメディア情報によるコミュニケーションを支援するアプリケーションが望まれている．しかし，これまでは，例えばNeXT ワークステーション上だけで動作するNeXTMail[2] のように，特定の計算機上でのみ動作するマルチメディア電子メール・電子ニュースシステムはあったものの，広く異機種間で動作するシステムは普及していなかった．これは，異機種間で使用可能なマルチメディア・データフォーマットの規格がないことなどの問題によるものであった．

このフォーマットの不統一の問題に関しては，1992年6月，Internet の標準化団体であるIAB(Internet Active Board) の発行するRFC(Request for Comments)1341 によって，マルチメディア電子メール・電子ニュースのための規格MIME(Multipurpose Internet Mail Extension)[4] が提案されている．これにより，ネットワークに相互接続された計算機間でのマルチメディア情報の交換が本格的に普及することが期待される．

ところが，MIMEは各種メディアを扱うためのフォーマットを定めているだけであり，実際にそれをどう実現するかはアプリケーション側に依存している．例えば，現在Sun SPARCstation 上で利用できるMIME リーダである`metamail`[5] は，X Window System の端末エミュレータである`xterm` あるいは`kterm` 上でマルチフォントテキストの表示を行なうため表示が非常に貧弱であり，また，テキストと画像が別々のウィンドウに表示されてしまうといった問題点がある．

そこで，本稿では，これらの問題点を解決するために，現在広く用いられているX Window System 上で動作するMIME リーダ'UMIME' の設計と実現について述べる．今回の実装では，オブジェクト指向ユーザインタフェースツールキットET++[10] をGUI(Graphical User Interface) として使用し，各メディアをオブジェクトとしてテキスト中に挿入することで上記の問題点を解決している．

本稿の構成は以下の通りである．まず2章では，電子メール・電子ニュースのマルチメディア化について述べる．ここでは，MIME フォーマットについて説明する．また，従来のMIME リーダの問題点についても述べる．次に3章では，今回実装したMIME リーダ'UMIME' の基本設計について述べる．4章では，'UMIME' の実装について述べる．ここでは設計の概要，ET++に追加したクラスおよび実現方法について述べる．5章では，'UMIME' の実装結果およびその評価について述べる．最後に6章では，本稿の結論，および今後の課題について述べる．

2 メール・ニュースのマルチメディア化

電子メール・電子ニュース配送システムは，大きくメール・ニュースをローカルシステムで読み書きするためのUA(User Agent) と，それを配送するためのMTA(Message Transport Agent) に分けられる．

従来からの電子メール・電子ニュースシステムは，文字情報によるコミュニケーションであったため，伝えることのできる情報量に限りがあった．そのため，より情報量の多い音声や画像といったマルチメディア情報を電子メール・電子ニュース上でやりとりすることの要求が起こってきた．

ところが現在の電子メール・電子ニュースでは，MTA が7bit で表される文字しか通過させないためメッセージ中にはASCII 文字しか含めることができない．そのため画像や音声といったバイナリデータを直接送ることができず，何らかの形でASCII 文字

に符号化する必要がある．

そのための規格がMIMEである．

2.1 MIME

MIMEとは，RFC1341で規定されているInternet上でマルチメディア電子メール・電子ニュースを扱うための規格である．MIMEは，従来の電子メール・電子ニュースのフォーマットに加えて次のような特徴をもつ．

- 非ASCII文字はASCII文字に符号化することによって電子メール・電子ニュースに含めることができる．
- フォーマット付きマルチフォントテキストを扱うことができる．
- 音声や，静止画像，動画像などマルチメディア情報を扱うことができる．
- メール本体を分離して送ることができる．

MIMEではMTAやUAに対する互換性を保つため従来の電子メール・電子ニュースの規格であるRFC822をヘッダを用いて拡張している．また，電子メール・電子ニュースの内容に構造を持たせることができ(図1参照)，UAで符号化されているバイナリデータの復号や，テキストと画像が混在するメッセージの取り扱いを行なうことができる．なお，MIMEではフォント属性を持つテキストを表現するフォーマットとして，`richtext`を用意している．

MIMEで拡張されたヘッダは

1. `MIME-Version:`
2. `Content-Type:`
3. `Content-Transfer-Encoding:`

の三つである．

`MIME-Version:`ヘッダは，MIMEフォーマットのバージョンを示す．`Content-Type:`ヘッダはメッセージの種類を示す．これにより，テキスト以外のメディアを含んでいることを示すことができる．`Content-Transfer-Encoding:`ヘッダは，7ビット文字コード以外のデータを送るときの符号化方式を指定するために用いる．

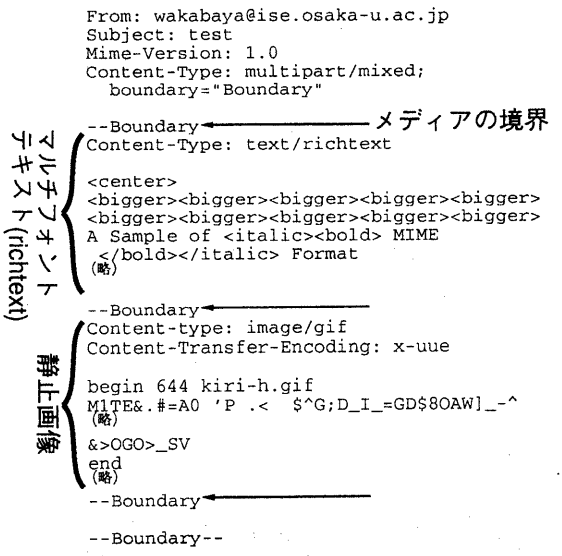

図1: MIMEによりマルチメディア化された電子メールの例

2.2 既存のアプリケーションの問題点

MIMEは，基本的にはRFC822に従いながらそれを拡張しているので，`sendmail`などの既存のMTAでそのまま配送することができ，画像や音声を含まなければ`mail`や`mh`といった通常のUAで読むことができる．ところが，MIMEは各種メディアを扱うための枠組を定めているだけにすぎず，マルチフォント文書の表示，画像の表示などの特殊な機能を実際にどう実現するかはUA側にまかされている．現在MIMEに対応したUAとしては，`metamail`と，NeXT上で動作するNewsBase 3.0がある．

`metamail`とは，MIMEの提唱者であるNathaniel Borenstein氏が作成したMIMEリーダである．しかし`metamail`は，MIMEという規格の実装例として作られているため実用性に欠けている．例えば，次のようないくつかの問題点がある(図2参照)．

- 画像を表示するために外部の表示プログラムを起動するため，テキストと画像が別々に表示される．
- 複数の画像を同時に表示できない．
- いったん表示された画像，あるいはいったん再生された音声はもう一度最初から記事を読まな

いと再生できない.

- テキストの表示に X Window System の端末エミュレータである xterm あるいは kterm を用いているため, richtext をマルチフォントで表示できず, 例えば"<biggrer> MIME</bigger>"は"_M_I_M_E"と表示される.

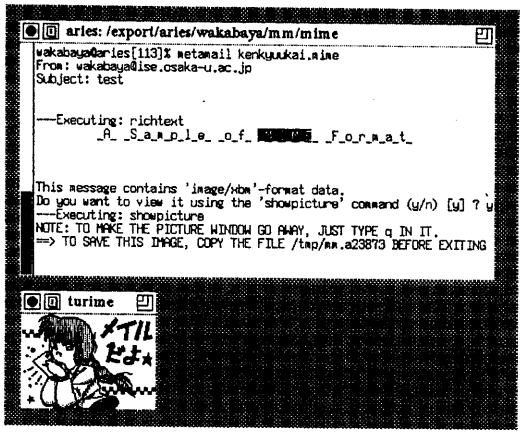

図 2: metamail による MIME フォーマットの記事の表示例

一方, NewsBase 3.0[3] とは MIME に対応したニュースリーダである. ところが, このニュースリーダは NeXT 以外では動作せず, 以下のような理由から移植性も低いため. MIME の機種に依存しないという特徴が活かされていない.

- NeXT は X Window System とは異なる独自のウィンドウシステムを用いている.
- 使用している言語は Objective-C であり, C や C++のように広く使われているものではない.

3 マルチメディア電子メール・電子ニュースリーダ 'UMIME' の基本設計

3.1 目標

前章で述べた従来の電子メール・電子ニュースリーダでの問題点を解決するため, MIME リーダ 'UMIME' を実装した.

その際, 以下の機能の実現を目標とした.

- 一つのウィンドウ中でのテキストと画像・音声の混在した表示
- 自由なスクロール
- richtext のマルチフォントでの表示
- 移植性があるソースコード

なお今回の実装では, MTA である nntpd や sendmail とのプロセス間通信によるメッセージの交換は行なわず, 単に MIME で記述されたファイルの表示を目的とした.

3.2 'UMIME' の設計方針

上述の目標を達成するためにウィンドウシステムとしては, 現在様々なワークステーションで稼働する X Window System を用い, オブジェクト指向に基づいた C++のクラスライブラリ ET++を使用した. それは以下のような理由による.

- C++は多くの計算機上で利用でき, 移植性が高い.
- ET++はオブジェクト指向を用いているため拡張性が高いため, 新しいメディアを追加するためにはクラスを追加するだけでよく, 容易である.
- 異なるメディアを統一的に扱うことができる.

MIME によるマルチメディア電子メール・電子ニュースを表示するためには, テキスト中に画像を混在させることができる必要がある. また, richtext を表示するためにマルチフォントテキストを扱えなければならない.

ET++には *VObjectText* クラスという枠組があり, この中ではテキストや画像といった各メディアを抽象的に表現することができ, テキストの文字と同様に挿入, 削除を行なうことができる. 'UMIME' ではこの仕掛けを利用し, 様々なフォーマットの画像や音声を *VObject* のサブクラスとして表現することでマルチメディア文書を実現している. つまり, テキスト中に様々なメディアを混在させ, スクロールも自在に行なうことができる.

また *VObjectText* は RTF(Rich Text Format) を扱うことができる. RTF は richtext よりも機能が豊富であるため, richtext は RTF に変換することによって表現することができる (表1参照). 静止画像は *VObject* クラスの一種である *Bitmap* クラスのオブジェクトに変換する. 音声は, アイコンを用いて新しいクラスを実装することで対応する.

表 1: richtext と RTF の違い

	richtext	RTF
トークンの数	少ない	多い
フォント情報	記述できない	記述できる
フォントのサイズ	明確には指定できない	指定できる
構文	入れ子	タグ
	(<token>と</token>で囲む)	(\token)

4 'UMIME' の実装

4.1 'UMIME' の設計の概要

'UMIME' の設計の概要を図 3に示す．次の 3 段階で MIME によるマルチメディアメール・ニュースを表示する．

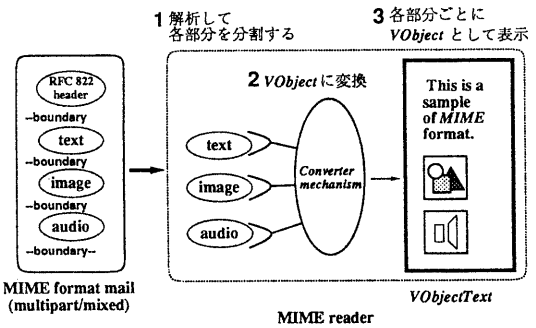

図 3: 'UMIME' の設計の概要

1. MIME による電子メール・電子ニュースを分解
2. 各部分を *VObject* クラスのオブジェクトに変換
3. 変換した *VObject* クラスのオブジェクトを *VObjetText* に読み込む

MIME によるマルチメディア電子メール·電子ニュースはは `Content-Type:` ヘッダの内容に `multipart/mixed` を指定することで表現され，`boundary` パラメータで指定された境界ごとに異なるメディアのデータが格納されている．従って `multipart/mixed` で構造化されているメール・ニュースは，まず `boundary` で指定されている行ごとに分割して各メディアごとにデータを取り出す．このとき，バイナリデータは ASCII 文字に符号化されているのでこれを復号する．

ET++ではファイルなどから *VObject* のサブクラスへ変換するための機構が *Converter* クラスで実現されている．そのため，復号したバイナリデータから *VObject* クラスのオブジェクトへの変換は，この変換機構を用いることにより実現することができる．通常のテキストおよび静止画像の形式の内のいくつかについては，これらを *VObject* に変換する *Converter* クラスのサブクラスが ET++にすでに用意されている．`richtext` および ET++でコンバータが用意されていない形式の静止画像については，ET++が読み込める形式に変換するクラスを作成した．

音声については，対応するクラスが ET++上にないため，まず新たに，音声を表す *Sound* クラスやそれを表示するための *VObject* のサブクラス (*LipIcon* クラス) を実装し，次に音声データを *LipIcon* クラスのオブジェクトに変換するコンバータを実装した．

4.2 ET++へのクラスの追加

4.2.1 コンバータクラス

ET++ではファイルやストリーム上のデータを *Data* クラスのインスタンスとして扱い，AsObject() メソッドにより *Text* や *Bitmap* といったクラスのオブジェクトに変換する．

4.2.2 テキストの表示

ET++において，テキストデータを扱うクラスは図 4のようなクラス階層をなす．

Text	抽象テキストクラス
CheapText	フォント属性を持たないテキスト
GapText	より大きなテキスト
StyleText	フォント属性を持つテキスト
VObjectText	グラフィックオブジェクトを含むことのできるテキスト

図 4: ET++において Text を扱うクラス

ET++の *StyleText* クラスでは，ボールド，イタリックなどのマルチフォントテキストを扱うことができる．*VObjectText* クラスはこの *StyleText* クラスを継承しており，さらに，テキスト中にグラフィックオブジェクト (*VObject*) を混在させることができる．

ET++にはテキスト用のコンバータとして *RtfConverter* クラスが用意されており，RTF の文書を表示することができる．そこで `richtext` の表示については RTF に変換するコンバータクラス *RichtextCon-*

verter を作成し，`richtext` を読み込むときはこれが呼び出されるようにする．

4.2.3 静止画像

ET++は，画像用のコンバータとしてPBM(Portable BitMap), PPM(Portable PixMap), XBM(X BitMap), MacPict, EPSF(encapsuled Postscript), SunRaster などのコンバータをもち，これらのビットマップ画像を *VObject* の一種である *Bitmap* に変換することができる．

今回の試作ではこれらに加えて，画像用のコンバータとしてGIF, JPEG 形式のコンバータを追加した．これらのコンバータは，内部で `pbmplus` コマンドを呼びだし．それにより生成された作業用ファイルを通常の変換機構を用いて *VObject* に変換している．

4.2.4 音声

音声については直接対応する *VObject* がないため，新たに *VObject* を設計した．まず，指定されたファイルを`/dev/audio` に書き込む *Sound* クラスを実装した．次に，*ActionButton* クラスを継承し，*Sound* クラスのインスタンスをメンバ変数に持つ *LipIcon* クラスを作成した (図5参照)．

ActionButton クラスでは，マウスがボタン上で押された場合の動作を決めることができる．そこでマウスのボタンが押された場合に *Sound* クラスの sound_play() メソッドが呼ばれ，音データが再生されるように設定した．また，*ActionButton* クラスは *VObject* クラスを継承しているため，*LipIcon* クラスは *VObject* として *VObjectText* 中に挿入することができる．

最後に，音声データを *LipIcon* に変換するコンバータを実装した．

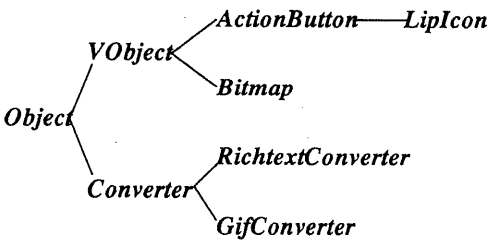

図 5: 'UMIME' で使用したクラス階層

以上のように，各メディアデータを ET++のオブジェクトとして取り扱い，テキスト中にはめ込むことで，自然な形でテキストデータと画像・音声を混在させて表示を行なうことができる．

4.3 実現方法

各種のメディアデータから構成されている MIME による記事を解析するためには，その構造にしたがって記事を分割する必要がある．

表示は以下の手順1, 2 を繰り返して行なわれる．

手順1. MIME による記事が `Content-Type: multipart` で始まっている場合，この記事を `boundary` で指定されている行ごとに分割し，各部分ごとに作業用ファイルを作成する．

手順2. 作成した各作業用ファイルを *Document* クラスのメソッド DoReadData() を用いて *VObjectText* 上に読み込む．DoReadData() の内部では，読み込もうとするファイルのヘッダによりメディアとフォーマットを判断し適切なコンバータを呼ぶことで変換が行なわれる．

上記の手順 1 において，MIME ヘッダおよび RFC822 ヘッダの解析は，`metamail` のルーチンを流用した．記事の対象部分が 7bit コードに符号化されたバイナリデータである場合は，`Content-Transfer-Encoding:`で指定されている符合化方式に従って復号する．

手順1, 2 ではストリームを用いて作業用ファイルの受け渡しをしており，ストリームをファイル上ではなくメモリ上に作成することで，作業用ファイルを作成せず高速に処理を行なうことができる．

5 'UMIME' の実装結果および評価

今回試作したマルチメディア電子メール・電子ニュースリーダ 'UMIME' では，`metamail` での問題点は解決され，以下のような機能を実現することができた．

- 画像もまたET++のオブジェクトとしてテキストの内部に挿入するため，レイアウトやスクロールの問題が生じない．
- 同時に複数の画像を表示することができる．
- 音声をアイコンとして実現している．
- ET++のテキストクラスを利用することによりフォントやそのサイズの変更などを自由に行なうことができる．

- C++, X Window System, ET++, UNIX の組合せにより，高い拡張性，移植性を実現した．
- ET++は，オブジェクト指向に基づいているため，新しいメディアの追加が容易である．
- 静止画像のフォーマットとして，PBM, XBM, GIF, JPEG など多くのフォーマットを用いることができる．

図6に図2と同じMIMEによる記事を'UMIME'で表示した例を示す．`richtext` の表示やテキストと画像の混在の問題点が解決されている．

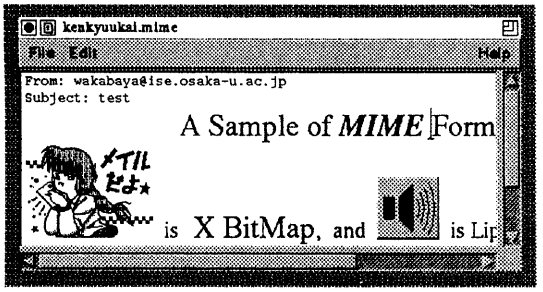

図 6: 今回試作したMIMEリーダの場合

一方，以下のような問題があることも分かった．

1. `richtext`，画像の表示に少し時間がかかる
2. プログラムサイズがかなり大きい (約3.7MB)

1. は，作業用にファイルを生成するためにハードディスクをアクセスする時間が原因であると思われる．この問題点については，ハードディクス上ではなくメモリ上に入出力ストリームを生成することで改善する余地がある．

2. は，ET++が高機能なアプリケーションフレームワークを持っており，ダイアログやイベントハンドリングのためのルーチンなどが実行形式のファイルに含まれることが原因である．現在では，ET++を用いた実装ではこの問題点を解消することはできないが，そのかわりに，共通のフレームワークを用いることで開発期間の短縮や操作性の統一を実現することができる．また，将来的にはET++を共有ライブラリとして実現することで，メモリ上のプログラム間でライブラリ部分を共有することができ，プログラムサイズの問題も解消されることと思われる．

6 結論

本稿ではMIMEフォーマットによるマルチメディアメール・ニュースの表示システム'UMIME'を構築した。'UMIME'では，ET++の *VObjectText* クラスとコンバータメカニズムを組合せたことにより，テキストと画像や音声を統一的に扱って同一ウィンドウ上に混在させることができた．また，MIMEでのマルチフォントテキストである `richtext` をRTFに変換し，ET++の *RtfConverter* クラスを利用して *VObjecxtText* に読み込ませたことにより，マルチフォントテキストの表示が貧弱だった点を解決した．さらに，多くの機種で用いることのできる言語やウィンドウシステムを用いたことにより，移植性を高くすることができた．

以上より，今回構築した'UMIME'では従来利用できたMIMEリーダであるmetamailやNewsBaseでの問題点を解決することができたと言える．

今後の課題としては，日本語への対応と記事の入力インタフェースの開発が挙げられる．これらについて説明すると，まず，ET++はスイス銀行で開発されたため，日本語や中国語などの8bit文字を扱うことはできない．日本語の表示を行なうためには *Char*, *Text*, *Font* クラスなどに改良を加える必要があるだろう．また，'UMIME'ではMIMEによるメッセージを表示することはできるが，テキストや画像をもとにMIMEによるメッセージを生成することはできない．このようなツールを作成するためには，*VObject* クラスの各サブクラスでファイルにデータを出力するメソッドを作成し，これを用いてハードディスク上に符号化したデータを書き込むようにすればよい．

最後に将来への展望としては，マルチメディアメール・ニュースを用いた，マルチメディア文書に対する計算機支援協調作業 (CSCW; Computer Supported Coopertive Work) への応用がある．画像も含めた文書全体に対する協調作業には，MIMEによるメール・ニュースを扱うことのできる'UMIME'のようなシステムが不可欠になるであろう．

謝辞

本研究を進めるにあたり，熱心な御討論を頂いた大阪大学大型計算機センター藤川和利博士に，また，有益な助言を頂いた大阪大学基礎工学部情報工学科春本要氏に深く感謝します．

参考文献

[1] 佐藤 豊: "Japanese subject in MIME format," *fj.sources.d*, Message-ID: <gKLsG.ysato@etl.go.jp> (Jul. 1992).

[2] 下條真司, 宮井計人, 鈴木勝典, G. Arakaki: "マルチメディア・メール／ニュースの夜明け," *UNIX MAGAZINE*, vol. 7, no. 12, pp. 46–52 (Dec. 1992).

[3] 宮井計人, 鈴木勝典, G. Arakaki: "マルチメディア電子ニュースシステム: NewsBase3.0," *ISR Technical Report* (Oct. 1992).

[4] N.S. Borenstein and N. Freed: "MIME(Multipurpose Internet Mail Extensions): Mechanisms for specifying and describing the format of Internet message bodies," *Request for Comments 1341*, DDN Network Information Center, SRI International (June 1992).

[5] N.S. Borenstein: "Multimedia mail from the bottom up or teaching dumb mailers to sing," *in Proc. USENIX WINTER '92* (1992).

[6] D.H. Crocker: "Standard for the format of ARPA Internet text messages," *Request for Comments 822*, DDN Network Information Center, SRI International (Aug. 1982).

[7] B. Kantor and P. Lapsley: "Network news transfer protocol, a proposed standard for the stream-based transmission of news," *Request for Comments 977*, DDN Network Information Center, SRI International (1986).

[8] J.B. Postel: "Simple mail transfer protocol," *Request for Comments 821*, DDN Network Information Center, SRI International (Aug. 1982).

[9] G. Vaudreuil: "MIME: Multi-media, multilingual extensions for RFC822 based electronic mail," *ConneXions*, vol. 6, no. 9 (Sep. 1992).

[10] A. Weinand, E. Gamma, and R. Marty: "Design and implementation of ET++, a seamless object-oriented application framework," *Structured Programming*, vol. 10, no. 2 (Jun. 1989).