

第 14 部
マルチメディア

第 1 章

はじめに

WIDE Multimedia-WG では、汎用の計算機でマルチメディアを利用できるアプリケーションの「評価」及び、「開発」をおこなっている。本 WG では、広域分散ネットワーク環境で利用されるマルチメディア・アプリケーションを研究対象としており、本報告書では Multimedia-WG の 93 年度の活動をもとに、以下のテーマについて報告する。

- オーディオ・コンファレンス・システムとその改良
- ビデオ・コンファレンス・システムの紹介
- Unicast を用いた映像音声情報及び制御信号の実時間的伝送の実験
- マルチメディア・メール
- マルチメディア情報のモデル化

第 2 章

オーディオ・コンファレンス・システムとその改良

2.1 序論

近年のハードウェアの進歩によって、文字だけではなく音声や画像を入出力することのできるデバイスを備えた計算機が広く一般に普及しつつある。これにともなって音声や画像を利用したアプリケーションの可能性が注目されるようになり、いろいろな用途のアプリケーションが試作されるようになった。本章では、これらのアプリケーションの中から特にインターネット上での音声会話ツールについて述べる。

インターネット上で利用できる音声会話ツールは、既にいくつか試作されて利用実験が行われている。本章では、これら既存のツールの概要を紹介した後、試用実験によって明らかになった音声の途切れの問題に関する考察と実験およびその結果について述べる。

2.2 既存の音声会話ツールの概要

現在、我々がインターネット上で利用できる代表的な音声会話ツールとして、Lawrence Berkley Laboratory の Van Jacobson および Steve McCanne 両氏によって作成された vat(Visual Audio Tool) と、Massachusetts 大学の Henning Schulzrinne 氏らによって作成された NEVOT(NETwork VOice Terminal) をあげることができる。これら 2 つのツールはどちらも X-window ベースの音声会話ツールで、プロトコルに互換性があるため相互のツール間で会話が可能である。以下に、これら 2 つの音声会話ツールの特長を示す。

- IP マルチキャストを利用した多人数での会話
IP マルチキャスト [20] の技術を利用することによって多人数での会話を実現する際の回線利用効率を高めている。
- 音声符号化方式を選択可能
音声入力デバイスから取り込んだ音声の符号化方式として vat,NEVOT とともに複数の方式を用意している。このためユーザは通信回線などの負荷に応じて最適な音質を得るような符号化方式を選択することができる。表 2.1 は vat で利用可能な符号化方式である。NEVOT でもほぼ同様の方式が利用可能である。

- データ転送時に無音部を省略可能

我々の日常会話は有音部 (talkspurt) と無音部 (silence) から構成されている (図 2.1 参照)。vat では、送信時にデータ中の有音部と無音部を識別して有音部のみのデータを伝送することによって、通信回線にかかる負荷を軽減している。

表 2.1: vat で使用する音声符号化形式

名称	符号化形式
pcm	78Kb/s 8-bit μ -law encoded 8KHz PCM (20ms frames)
pcm2	71Kb/s 8-bit μ -law encoded 8KHz PCM (40ms frames)
pcm4	68Kb/s 8-bit μ -law encoded 8KHz PCM (80ms frames)
idvi	46Kb/s Intel DVI ADPCM (20ms frames)
dvi2	39Kb/s Intel DVI ADPCM (40ms frames)
dvi4	36Kb/s Intel DVI ADPCM (80ms frames)
gsm	17Kb/s GSM (80ms frames)
lpc4	9Kb/s Linear Predictive Coder (80ms frames)

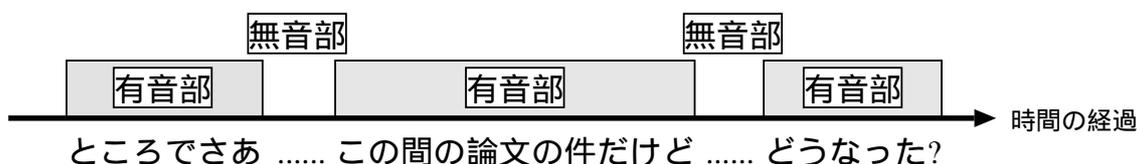


図 2.1: 有音部 (talkspurt) と無音部 (silence)

vat および NEVOT は音声入出力デバイスを持つ様々な計算機に実装されている。vat は現在 β リリースのためバイナリのみでの公開であるが、NEVOT はソースが全て公開されている。以下に 1994 年 4 月現在で vat が実装されている計算機を示す。このうち NEVOT が実装可能なものは Sun と SGI の計算機だけである。

- SPARCstation (Sun Microsystems)
- SGI Indigo/Indy (Silicon Graphics)
- DEC/5000, DEC/3000 (Digital Equipment)
- Snake workstation (Hewlett Packard)

2.2.1 音声会話実験

広域ネットワークの回線容量増大と IP マルチキャスト網の発達によって、従来は構内網内に限られていた vat や NEVOT のような音声会話ツールの利用実験が、近年では広域ネットワーク上でも行われている。広域ネットワーク上での音声会話実験で最も大規模なものは、1992 年 3 月から開始された IETF の会議中継実験である [19]。IETF の実験ではアプリケーションに vat を利用しており、IP マルチキャスト網を通じて世界中に会議内容が中継される。また国内でもネットワーク関係の会議を中心に同じような中継実験の試みがなされている [193]。

我々はこれらの実験の参加者に対するアンケート調査と自らの経験より、現在のインターネットで利用可能な音声会話ツールには以下の 2 つの問題点があると考えている。

1. 会話相手との接続手段の不備
2. 音声の途切れの発生

vat や NEVOT には会話したい相手呼び出したり、他の人の会話に自分が途中参加の意志表示をするための機能が実装されていない。このため、会話をするためには電子郵便のような他のコミュニケーションツールを併用して相手呼び出しや参加意志を伝えなくてはならない。音声会話ツールを使用する際に他のコミュニケーションツールを併用した場合、音声会話ツールだけを使用する時と比べ操作が複雑になり利用者に負担がかかる。従って、音声会話ツールの実用化には相手呼び出しや会話への参加意志を伝えるための機構を組み込む必要がある。

このような機構を音声会話ツールに組み込む際には、talk や phone にある会話相手呼び出す機能を実現する技術を音声会話ツールに応用することが可能である。またこの問題は、インターネットを通じた音声会話そのものの問題ではなく人間とツールとの間のインタフェースの問題なので本章では特に議論しない。

もう 1 つの問題点である「音声の途切れ」とは、話者の言葉が発言の途中で途切れて会話内容を聞きとることが困難になる現象を指す。会話内容を聞きとることが困難になると、円滑に会話を進めることができなくなるのでコミュニケーションツールとして使用することはできない。従って既存の音声会話ツールを実用化するには、音声の途切れが発生しないようにする改良が必要となる。そこで次節では既存ツールの音声通信、処理の方式を紹介し、音声の途切れが発生する原因とその解決手段について述べる。

2.3 音声の途切れ

2.3.1 発生の原因

既存ツールでの音声通信及び音声処理の流れを図 2.2 に示す。図 2.2 で、横軸は時間の経過、縦軸は処理の進行を表している。また、有音部のデータは斜線で、無音部のデータは空白で表している。Ts は音声をパケットに分割する時の単位時間である。

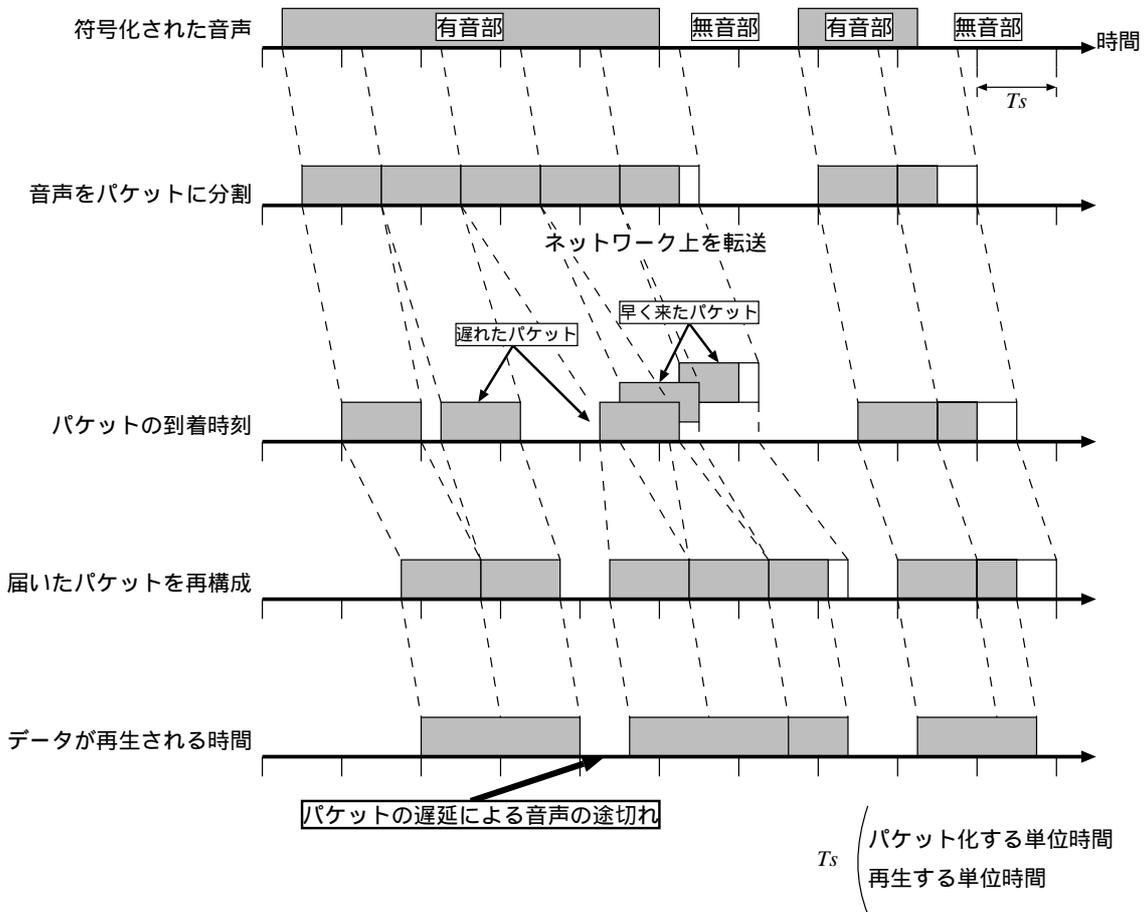


図 2.2: インターネットを通じた音声伝達

発言者が発した音声は、まず符号化されて計算機に取り込まれる。計算機はそれを T_s 時間おきにパケットにして相手に伝送する。受けとられたパケットは元の連続したデータの形に再構成する作業が終了した後、直ちに復号して再生される。以上が発言者が発した音声聞き手の計算機で再生されるまでの処理の流れである。

この一連の処理を実行するためには各段階である程度の処理時間が必要となるため、発言者が発した音声は聞き手側の計算機で遅延して再生される。この遅延時間が常に一定の場合は、発言者の音声は途切れなく再生される。しかしながら既存のオペレーティングシステムでは、各処理を行うプロセスの実行時間が他のプロセスの実行状況によって変動する。また現在のインターネットで用いられているプロトコルでは、音声パケットの伝送時間は他のアプリケーションによって発生しているトラフィックの影響を受けて変動する。従って、音声処理によって生じる遅延時間は実際には一定ではなく変動している。これが既存の音声会話ツールで発生する音声の途切れの原因である。図 2.2では、音声パケットがインターネット上を伝送される時間の変動が再生される音声に影響を与えて音声の途切れが発生する様子を示している。

以上より、音声会話ツールの使用中に音声は再生時に途切れないようにするためには遅延時間の変動が発生しないようにするか、もしくは遅延時間の変動が発生しても音声は再生時に途切れなくなるような工夫をする必要があることがわかる。次節では、再生時の音声の途切れを回避するための方策についてこの2つの側面から考察する。

2.3.2 処理時間の変動補正

音声パケットがインターネット上を伝送される時に生じる遅延の変動は音声パケットの伝送のために必要な回線容量が常に確保されないために発生する。またプロセスの実行時間の変動は、オペレーティングシステムのスケジューリング機構やメモリ管理機構に各プロセスの実行時間を保証するための機構がないために発生する。

従って遅延時間の変動が発生しないようにするためには、回線容量の予約、実時間プロセス制御といったシステムレベルの機構の改良が必要となる。しかしながらシステムレベルでの問題解決は計算機環境全体に与える影響が大きいため、アプリケーションが利用可能になるまでの環境移行には多大な時間が必要である。そこで我々は、システムレベルの改良によって音声の途切れを解消する方式の検討と同時に、アプリケーションレベルで遅延時間の変動を吸収する機構を導入することによって音声の途切れを解消する方式について検討を行っている。

既存の音声会話ツールでは、処理時間の変動を補正するための機構として図 2.2 の各処理段階でのデータの受渡しをバッファを介して行っている。バッファ機構を導入することによってデータ読み書きのタイミングを調整することができるので、処理時間の変動を補正する事が可能である。

しかしながらバッファの容量には限界があるので、送信処理が遅れて送信待ちのデータがバッファ容量の限界を越えた場合にはデータの一部を廃棄しなくてはならない。この時廃棄されたデータは再生側では出力されないため、再生時に音声に途切れが発生する。また受信処理が遅れて次の処理に渡すためのデータがバッファ中に無くなった場合には、次の処理に対するデータ送信を中断せざるを得ない。この結果送信の中断によって再生データの伝達に途切れが生じるため、再生音声に途切れが発生する。そこで我々は、既存のツールで生じる音声の途切れを解消する機構として、従来のバッファ処理に加えて以下に示すような機構を追加実装することを検討している。

- 音声の符号化方式を状況に応じて動的に切替える
- バッファの状態に応じて有音部の再生に遅延を与える

符号化した音声データの大きさは再生時の音声の音質によって異なっている。一般に符号化後のデータを小さくする場合は、標準化および量子化の精度を下げたり符号化する音声の周波数帯域を狭めたりする手法が用いられるため、音質はデータの縮小に応じて低下する。一方で、データの大きさに応じて計算機およびネットワークにかかる負荷は大きくなるため、利用可能な音声の品質は制限される。既存のツールでは現在のネット

ワークの利用状況などの計算機資源の状態と利用したい音声の品質から、利用者が自分で使用する符号化方式を選択しなくてはならない。従って、常に現在の計算機環境を正しく反映した符号化方式を使用した音声伝達をしているとは限らない。我々はこの問題を解決するために、計算機の負荷やネットワークの回線状況に応じて最適な音声符号化方式に切替える機構を既存の機構に追加することを検討している。計算機の負荷やネットワークの回線状況は、音声データの遅延時間の変化と各処理段階のバッファのデータ量によって判定して送信側に伝達する。ただし、符号化方式の切替えによるデータの縮小には限界があるので、この機構だけでは既存ツールでの音声の途切れを解消するには不十分である。

有音部の再生に遅延を与える機構とは、音声パケットの遅延に応じて有音部の再生開始時刻に適切な遅延を与えることによって音声パケットが遅延しても再生時刻に間に合うようにする機構である。図 2.3は、再生開始時刻に新たに付加した遅延によって図 2.2で発生していた音声の途切れが解消されている状況を示している。図 2.3より、この機構を追加することによって、音声の再生時に音声の途切れが発生しなくなることがわかる。またこの機構では、パケットの遅延が著しい場合は再生に大幅な遅延が必要となる。その場合は 1 つの有音部の音声パケットがすべて再構成されるまで再生せずに待つことによって音声の途切れない再生を行なうことができる。

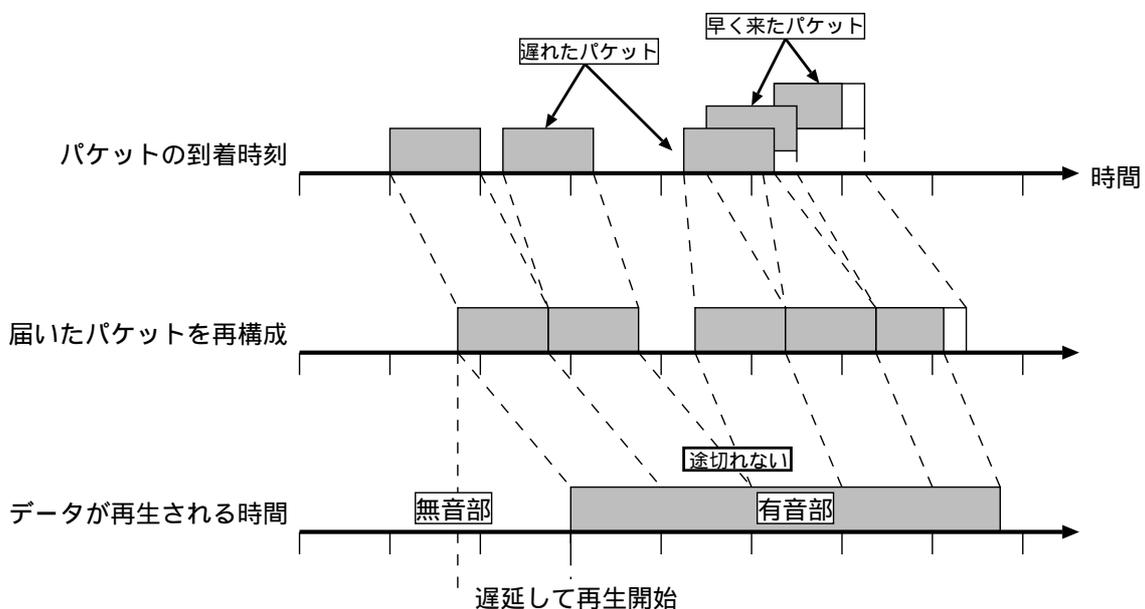


図 2.3: 有音部の再生に遅延を与える

2.3.3 遅延の問題

音声が届かないようにするために音声の再生開始時刻に遅延を与えると会話の進行に遅延が生じることになる。音声の伝達の際に話者と聞き手との間に生じる遅延はコミュニケーションの障害になるので、音声会話ツールの製作時にはできるだけ遅延を生じさせてはならない。しかしながら、コミュニケーションで最も重要なことは相手への意志伝達である。従って遅延を作ることによって会話内容が正確に相手に伝わるのならば遅延が生じるのもやむを得ない処置であると我々は考える。

また、遅延の発生を許容することによって既存のツールでは音声が届いて会話で困難になるような回線容量の小さい回線を使用した場合でも遅延が発生するという制限つきながら会話が可能となる。これは、既存のツールと比較するとコミュニケーションが可能となる範囲が広がるという点で大きな利点になっていると我々は考えている。

2.4 実装及び評価

前節で提案した追加機構を実装した音声会話ツールを作成するための前段階として、我々はまず有音部単位での再生を行なう音声会話ツールを試作した。この試作ツールを用いて、遅延が会話へ及ぼす影響と既存ツールでの音声の途切れによる会話への影響とを比較した。試作ツールは、ソースリストが公開されている NEVOT のプログラムの送受信時のバッファ制御をおこなう部分に変更を加えることによって製作した。また、試作ツールは Sun Microsystems の SPARC Station 上で実行し評価を行なった。

評価は、イーサネット (10Mbps) 及び広域ネットワークで接続された 2 台の SPARC Station 間で会話実験をおこなって、vat と試作ツールとの比較実験を行なった。広域ネットワーク上での実験は奈良先端科学技術大学院大学と九州大学の間で行なった。実験時この 2 組織間の回線で最も回線容量の小さい部分は 192Kbps であった。また、vat については音声の圧縮方式を切替えても音声が届かないか確認した。

この実験の結果を表 2.2 に示す。実験の結果、イーサネットを使用した場合には vat と試作ツールの両方で音声の途切れのない円滑な会話が可能となった。しかしながら、試作ツールでは音声に遅延があるため会話の展開が vat と比較した場合遅くなる。広域ネットワーク上での実験では、vat を使用した場合は音声の途切れが激しく会話は頻繁に中断した。一方試作ツールでは音声の途切れはなく vat と比較すると円滑な会話を行うことができた。

またこの実験で、試作ツールを使用した会話において利用者が遅延の状態を把握できないと会話内容に食い違いが生じて混乱が発生することがわかった。このことから、回線の状態に応じて会話に遅延が発生させる場合には、その時の音声の遅延状態を利用者に何らかの形で提示するような機構を実装する必要があることが必要だと考えられる。

表 2.2: 試作ツールの実験結果

	LAN (Ethernet)		Internet(NAIST ⇔ Kyushu Univ.)	
	vat	試作ツール	vat	試作ツール
再生時の遅延	0.15~0.25sec	over 1sec	0.25~0.5sec	over 1sec
音声の途切れ	little	none	many	none

2.5 今後の課題

本章では、現在インターネット上で音声会話を行うためのツールとして注目を集めている vat と NEVOT について紹介した。また、これら既存の音声会話ツールを実用化する際の問題点を明確にしてその解決策を提示した。また試作ツールを使用した実験によって、回線状態に応じた遅延を挿入する方式の実装によって円滑な会話が可能で音声会話ツールを実現できることを確認できた。今後は今回提示した解決策に基づいて、音声パケット処理の遅延状態に応じた音声再生の遅延時間の決定方法について考察して、今回提示した解決策を用いた多人数での音声会話のための手順を確定させたいと考えている。

また本章では音声のみを媒体とした会話ツールに的を絞って考察してきたが、コミュニケーションの手段として会話ツールを考えた場合、音声だけではなく映像や文字などの媒体も並行して使用可能であることが望ましいと我々は考えている。このようなツールを実装する場合には、今回考察した遅延の変動の問題に加えてそれぞれのメディアの間で再生時に同期をとる必要が生じる。そこで今後は音声だけではなく映像や文字を併用してコミュニケーションする場合についても検討を重ねて、さまざまな媒体が並行して使用可能な会話ツールの実現についても考えていきたい。

もう 1 つの課題は、システムレベルでの機構改良による遅延時間の保証に関する検討である。既存のツールにおけるシステムレベルでの機構改良の試みの 1 つに、ST-II(Experimental Internet Stream Protocol,Version 2)[3] プロトコルを利用した回線容量の確保機構をあげることができる。NEVOT では、この ST-II を利用することによってシステムレベルで音声の途切れを解消する試みがなされている。我々も今後はアプリケーションレベルでの改良検討にくわえて、このようなシステムレベルでの改良に関しても試作実験を行っていきたいと考えている。

第 3 章

ビデオ・コンファレンス・システムの紹介

3.1 NetVideo(nv) ビデオ会議ツールの概要



nv の制御画面・描画面・キャプチャー画面

”nv”は Xerox PARC の Ron Frederick¹によって制作された、インターネット上で UDP/IP を用い映像情報の送受信をマルチキャスト/ユニキャストで行えるツールである。映像情報の通信はユニキャストだけでなく、マルチキャスト対応カーネルの基で IP マルチキャストを用いたグループ間通信も可能である。送信者は映像を端末に取り込むためのハードウェアを必要とするが、受信者は特に専用のハードウェアを必要としない。このハードウェアの性能によりフレームレートは現在のところ低い値となっている。まだ完全にすべてのビデオ取り込みの為にハードウェアをサポートしていないが²今後更に拡張する予定らしい。

映像サイズは 320×240^3 (pixel) であり送定の段階で 8 ビットグレーか 24 ビット YUV 4:2:2 カラーを選択可能である。グレーモードの方が伝送量が少なく済むので、伝送回線が細い場合などに向いている。受信端末では可能ならば 24 ビットカラーで受信画像を表示し、さもなければ色数を落とすなどの処理を行う。フレームレートは映像の差分量や利用できる回線幅などによって左右されるが、平均的には標準環境で与えられる 128kbps

¹frederick@parc.xerox.com

²nv Version 3.2 では Sun VideoPiX, SGI Indigo video, Sony news, DEC JVideo などをサポートしている

³NTSC の場合である。PAL の場合 384×288 (pixel)

の回線幅では 3 から 5 (frame per second) 程度である。機種によってはそれ以上に送出設定値を上げればフレームレートが上がるものもある。

nv は起動後のグラフィカルユーザインタフェースによって各種設定を行えるが、起動時のコマンドラインによっても以下の設定が可能である。

機能	option	標準値
明るさ	-brightness	50
コントラスト	-contrast	50
送出インタフェース	-interface	(primary host name)
最大許容回線幅	-maxBandwidth	128
最大フレームレート	-maxFrameRate	(machine-specific value)
ユーザー名	-name	(username@localhostname)
受信画面サイズ	-recvSize	normal
ウィンドタイトル	-title	nv (version)
TTL ⁴	-ttl	16

また同時に X の標準環境と同様に "-display" option も用いることで、映像取り込みハードウェアを搭載していない端末でも映像を疑似的に受信することが可能であるが、その際は nv 起動時から受信専用設定される。

3.2 送信の設定

送信映像のフレームレートは 0 から最大 15 (fps) に可変出来る。特に設定値 0 はシステム固有のフレームレートを指定しており、例えば SGI Indigo においては最大標準フレームレートは 10 (fps) となる。

"interface" option では送出時のネットワークインタフェースが選択出来る。複数のネットワークに接続されている (multi-homed) 送出ホストでは、マルチキャスト送出に適切なインタフェースを設定すべきである。通常は上位ホスト側にセットされる。

標準設定での送出マルチキャストアドレスは 224.2.1.0/4444 でありマルチキャストアドレスを指定すれば一般的なグループ間同士でのマルチキャスト通信が可能である。しかし、ここでユニキャストアドレス (Class A, B, C) を指定すると UDP によるユニキャスト通信が確立される。

3.3 送受信の制御

受信画像サイズは随時 half, normal, double が選択可能であるが、受信画像サイズを小さくしても、画像をグレーモードにしても送出側伝送情報容量は一定である。ユニキャスト接続時であっても同様である。マルチキャスト送出を中心に考えられており、クライアント (受信側) は双方向な制御をサーバに対して要求することは出来ない。サーバが知る事が出来るクライアントの情報としてはホスト名とユーザ名などが挙げられる。

nv は起動時に映像取り込みハードウェアが利用出来るか否かを判断し、適当なハードウェアが搭載されて利用できる状況にある場合には送出制御画面を含め 3 つの画面を用

意し、さもなければ受信用の2つの画面だけを用意する。受信用画面の一つには、該当のチャンネルに送出されている映像が小さく映し出されるが、これを”underneath 画面”と呼ぶ。任意の underneath 画面をクリックすると、その映像を表示する標準的な大きさの描画面が現れる。ここで現れた画面の中でクリックすると明るさ、コントラスト、濃さなどが設定出来る受信制御画面が登場する。スクロールバー形式の設定項目ではバーを動かせば設定が反映されるが、文字設定の場合には数値入力後のリターンで初めて変更が反映される。受信した画像をキャプチャしたい場合には、制御画面のキャプチャボタンを選択することで、新しい画面が作成されその時点の受信画面にある画像が静止保存される。その画像をファイルに落としたい場合は xwd の画面ダンプツールを用いて保存する。但し、ディザ処理が行なわれていない画像の場合は⁵その処理を後ほど必要とするだろう。この他に送信可能な状況では先ほど述べた3つめの送信制御画面が登場し、最大許容伝送幅や gray 画像を伝送するかカラー画像にするかなどの設定が可能である。すべてグラフィカルな操作が可能で、例えば「送信開始」と指示されたボタンを押した後は、そのボタンが「送信停止」と指示されているなど、効率的な設計をされている。

描画面終了にはウィンドマネージャが提供している quit を用いるか、Ctrl-C, ESC, ‘q’などを描画面中で指示することで終了する。制御画面中で同様の処置を行うとプログラムの実行が終了する。

現在発見されているバグとしては Indigo video での起動時に”sv1LoadMap: bad sv1AddWork”という message がコンソール画面に出されるが、これは Indigo video driver の単なるバグであり無視して差し支えない。

make する際には、メジャー ftp サイトなどからソースを入手した後にプラットフォーム指定の Makefile で make すれば良いが、特に共通して別途必要なのは Tcl 6.7, and TK 3.2 などである。また、SPARCstation では VideoPiX library, SGI Indigo IndigoVideo では”svideo” library, DECstation では PIP Xv library が必要となる。但し最新の配布などは、binary 形式であることがおおい。

3.4 バージョンアップの変更点など

カラー画像をグレー専用モニタに描けるようになったという点で、version 3.2 はそれまでの 3.0,3.1 とは一線を画しているだけで、基本的には 3.0 と 3.2 は通信が可能である。2.x 以下とはパケットが完全に異なっているので相互の通信は不可能である。グレー専用モニタでも表示出来るようになり、利用範囲は格段に向上したと言えよう。同様にすべての画面に Tcl/TK を用い出したのも 3.0 からである。少なくとも画像伝送に関して高画質を要求しない限りは、現在の基本的要求を満たしており、現在のところマルチキャストでの実験などには最適なツールと言えるのではないか。なお、

- ver.1.0 Thu Nov 12 1992 (First public release)
- 3.0 Thu Mar 25 1993 (New major relwase)
- 3.2 xxx May xx 1993 (latest)

⁵静止部分から(通常画面上から)処理が行なわれる

と短期間に急激な変動を遂げていることは、インターネットの急激な成長に似ている。

3.5 考えられる改善点と問題点

3.5.1 IP マルチキャストでの運用に限って

マルチキャストでの国内外会議での利用や、相互連絡などの用途を考えてみるに ST-2 のような資源予約プロトコルを用い回線幅を確保し、伝送信頼性を備える必要があるかと考える。現在の回線幅設定はあくまでも最大値の設定に過ぎず最低値を保証している訳ではない。しかし、映像情報伝送ツールなどで資源予約を行っている、既存のネットワークはパンク状態になることは明かであるので、資源予約機能を追加することには十分な検討を必要とするだろう。

伝送された映像情報をソフトウェアによる MPEG 圧縮などを施しアーカイブ化し、再生することが出来ると便利であろうが、基本的に差分伝送しか行なわず、また独自の画像フォーマットを用いているので、それなりの工夫を要するだろう。既に画像ピクセル数が最大でも通常の 1/2 程度であり、かつ画像品質もかなり劣化しているので、これ以上の劣化圧縮は望まれないだろう。

3.5.2 ユニキャスト運用も含めて

インタラクティブな機能が殆ど提供されていないので、クライアント側での操作は皆無に等しい。画像サイズが小さくても許容出来る場合など、無駄な情報を伝送しなくとも済むような機能を持たせると効率が断然上がるであろうし、クライアントでの送出制限などを行えるように出来ることが望ましい。資源予約を行っても、マルチキャストほど突然酷い状態になる可能性は低いので、最初はユニキャストに限って資源予約機能を追加するのも良いかと考える。

現在は nv だけで通信されることは少なく、vat などの音声会議ツールと兼用される場合が多いが、将来的には統合されたツールが一般的になるのであろう。

3.6 製作者リスト

"nv" main programmer	Xerox PARC, Ron Frederick (frederick@parc.xerox.com)
SGI V.C.B ⁶ Supports	Andrew Cherenon (arc@sgi.com)
SONY NEWS V.C.B Supports	Tatsuo Nagamatsu (nagamatsu@sm.sony.co.jp)
DEC PIP, JVideo V.C.B Supports	Steve McCanne (mccanne@ee.lbl.gov)

第 4 章

映像・音声情報及び制御信号の実時間的伝送の実験



マルチモード画面の様子¹

4.1 はじめに

インターネットという世界的な計算機情報ネットワークの発展の中で、文字情報だけでなく画像や音声情報を含む、マルチメディアと総称される複合された情報の処理及び表現が注目されている。現在のところ、映像や音声情報を予めファイル形式で保存しておき、そのファイルの伝送と再生という手段によって情報サーバから利用者は情報を得ている。その為、情報サーバで多様なサービスを提供しようとする、膨大な容量のファイルを保存しなければならない。そのような中で私達はマルチメディアのような一般的に情報量の多い複合メディアに関して、二つの課題を掲げた。一つは情報の記憶場所と実時間性、もう一つは同一の伝送経路における他の伝送プロセスに対して強引な混雑発生を極力さけて、共存出来る必要性である。

- 計算機情報の記憶場所とマルチメディア情報の記憶場所を分ける外部機器依存方式による実時間情報のサービス
- ネットワークの伝送負荷状況に応じて、伝送する情報量を適宜変化させる協調通信

¹大画面サイズのウインドに小画面を連続的に描画している。

などを目標とするアプリケーション「MW²」の提案及び実装を行なった。

4.1.1 アプリケーションの主な特徴

1. TCP/IP を用いたユニキャスト マルチメディア アプリケーションである。
2. 音声、画像及び制御情報の伝送を同期を考慮しつつ連続的に行なえる。
3. 伝送回線の状況に応じて画像情報の品質を適当に変化させることが可能である。
4. 外部機器などを透過的に扱い情報保存コストを下げる事が可能である。
5. 映像音声情報を外部デバイス (VCR³ など) に依存可能である。

既存の映像/(音声) 情報伝送プログラムとして前記 nv⁴ や ivs⁵などのフリーソフトウェアが提供されている。そこでこれらと比較する。

4.1.2 nv, ivs などと異なる点

MW が優れている点	画質 (表示は 8bit color) Interactive な操作性 (サーバ ↔ クライアント) 外部シリアルデバイスの制御機能 差分検知による特殊表示/機能
MW が劣っている点 (未実装)	IP マルチキャストに対応していない 音声のエンコーディングが未実装 再生/伝送画像の毎秒フレーム数 伝送フレーム数がクライアント数に依存してしまう UDP が未実装である。

4.1.3 制作背景

ある広告代理店が行った大学生向けアンケートによると、男女交際を行う上で二人が離ればなれになった場合、いわゆる「遠距離恋愛」に対して自信があると答えたのは僅かに 10 パーセント以下であった。遠くに居る恋人より近くの隣人こそが、その人のとって最も大切な人になってしまうのであろうか。

実際には近くになくても、そのようには感じさせず普段から相手側の様子が分かるようなメディアがあればそのような問題は解決される。出張やミーティングで遠隔地に滞在しているとき、もし幸せなことにインターネットに接続されている環境が提供されたとすると、少なくとも文字情報の実時間的通信は確保されよう。しかし、文字だけでは表現し難い研究室などの様子や現状を「みたい」ことや、肉声を「ききたい」という衝

² mineo, moma, murai, WIDE という説や nv に一本ずつ線を加えたという名前の由來說がある

³ Video Cassette Recorder/Player

⁴ Xerox PARC, America (NetVideo)

⁵ INRIA Sophia Antipolis, France (INRIA Videoconference System)

動に駆られることは十分に想像出来る欲求である。勿論、国際電話などを用いれば音声での会話は十分に可能であるが、そのためのコストはかなり高くつくであろう。

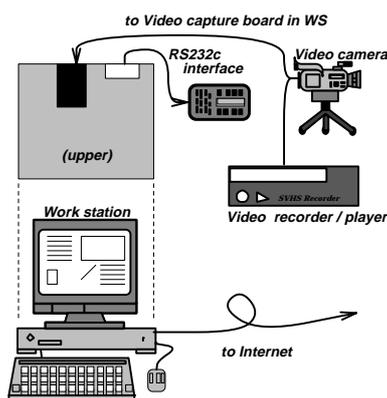
そこで、もしある程度ネットワークに余裕がありそうな場合、その資源を利用してみたいという発想が出てくる。とりあえず自分の研究室にいる友人と会話をしてみようということを試みたいだろう。また、友人の顔をみれば自分が遠隔地にいることさえ忘れさせてくれるかも知れない。

様々な人々に対して物理的距離を意識的距離から解放させることが出来るように今回の制作では特に意識した。よって主な目的として「フとした」時に相手が見れる、大体の様子が分かる事を目標としたので、既存の衛星や専用公衆回線を用いたテレビ会議システムのような実時間に相互映像と音声を広い回線幅を独占利用しながら伝送するものとは本質的に異なり、インターネットのような限られ共有されたネットワーク上で可能な現実的な設計及び実装を行った。これは単に回線幅が広がれば解決されるような問題を対象としているのではなく、回線幅が広がっても利用者が増加すれば同じ現象が発生するジレンマに対する考察でもある。

現実的に実験を行うため、伝送回線やビデオキャプチャーボードなどは既存のものを利用することとなり、キャプチャー能力はハードウェア的に 2(frame/sec) 程度となった。また画像の品質としては VHS 程度の画質で画像サイズは最大 640x480(pixel) となった。

伝送回線	イーサネットなど IP 接続回線
映像取り込み	Sun Video PiX
キャプチャー限界	2 frame/sec (max)
画像サイズ	640x480 (pixel/max)
音声	no special coding

評価機種: Sun workstation ss2 (RAM 64M)



計算機端末と周辺機材図

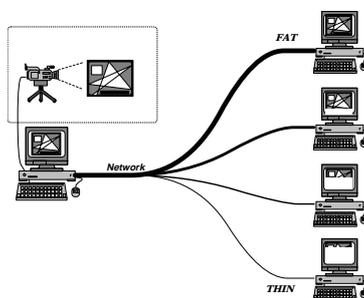
4.2 マルチメディア情報の扱い

高速演算処理装置などの発達に伴い、その高速性能を最も活かすことが出来るといわれる画像処理分野であるが、それはその情報が冗長性を多く含みがちであり、かつ膨大

な容量になる傾向から起因しているようである。つまりは、計算機がかつて無い冗長で膨大な情報を扱わなければならない。

しかし、計算機において冗長性のある膨大な情報を扱わせるのには、資源活用や物理的限界からみて、どうしても無駄なプロセスを生じる場合が多い。それを可能な限り回避すべく圧縮を施して解決しようとする動きが現在のところ大半である。

そこで私達は情報を計算機上の disk だけに蓄積するという計算機特有の考え方に特に着目し、可・不可逆圧縮情報である映像音声情報は計算機上の disk だけでなく、他の適当なデバイスに記録して必要な際に再生出来ても良いのではないかと考えた。今回の提案では以上の様な発想を実現させるために外部記憶媒体として現実的にすぐ成り得る映像音響機器に関わる操作も、シリアルデバイス経由でネットワークから制御出来る仕様を基本としている。



伝送幅と伝送容量概念図

4.3 情報の伝送段階における問題

マルチキャストではその性格上、広範囲に於いて回線の混雑度、利用度を上昇させる場合が多く、またサーバが常にクライアント各々を監視することは難しいという理由などから、今回の設計ではユニキャストを用いネットワーク負荷を極力抑え、気軽に利用出来得ることを意図した。その為に画像品質や、ソース選択、TVチャンネル切替えの機能など双方向操作も可能となり、プライバシーなどを保護することも短期的な解決方法として達成された。

mv などの「マルチキャストで広域に渡って画像を伝達するぞ」という意気込みに対して、MW では「回線が空いている時は高品質な画像を伝送させてもらうが、混んできたら品質を落すなりして他の人に迷惑を掛けたくない、でも使いたい」といった目標を掲げている。

情報の受信において、もし回線に問題が生じれば(回線の急激な混雑など)クライアントは出来る限りサーバに対して現在の回線状況に適した画像品質への変更を要求する事が出来る。この処理はサーバがマルチキャストなどと異なりクライアント個別に行なってくれる。

ネットワークレベルでは、現在 TCP/IP を利用している。映像音声情報のように、若干情報内容が変質しても許容出来る情報体質の場合、信頼性を多少落してもオーバーヘッドを落す事が望まれる。その事を考慮すると、少なくとも TCP より UDP を利用する方

が良いのかも知れないと考えるのが普通である。現在は同期の問題で使い易い TCP を用いているが、今後の課題として認識している。

4.3.1 画像処理について

MW ではクライアントはサーバに対して「インタレース値 (以下 IR)」を指定することが可能である。インタレースとはテレビジョン技術でご存知であろうが「飛び越し走査」という意味である。テレビジョンでは通常 512 本ある画像の走査線をこのインタレースを用いて 256 本にして伝送を行なっている。そのため画像の 1 フレームを 2 フィールドという構成に分割して伝送を行ない円滑な動画を提供している。この技術と同様に、MW ではキャプチャーした画像に対してインタレースを行なう。MW のインタレースではテレビジョンのインタレース値が固定値 2 であるのに対して、MW ではその値が可変であり、その値を前記「IR」値と呼ぶ。例えば IR 値が 256 の場合 512 本の走査線があった場合、伝送されるのは 2 本となる。また、ここでもテレビジョンと異なりテレビジョンでは残りの走査線も伝送するが MW では伝送しない。勿論、その為に情報量が減少して、そのままでは再構成出来ないが、伝送終了後、つまりクライアント側で再生する際に飛び越した線 (line) 分の情報を補間している。この補間する作業に於ては、前方推測方式や後方、両方推測などの概念もあるが、ここでは単純に伝送された情報をリフレインしている。そのため、ある程度 IR 値が大きくなると、まるでモザイクのような画像になる。

更に MW では通常の縦方向のインタレースだけでなく横方向のインタレースも可能としており、これにより目的に合わせた画像圧縮的伝送を可能としている。

ある程度ネットワークが混雑しても映像音声情報などが実時間的に伝送出来るようにクライアント側での実再生時間を測定し、それに適応した IR 値に自動設定を行なう。この情報を基にサーバは伝送を行なうので、実再生時間性をクライアント側では保持しようとする。急激な許容回線幅の変化があり得るネットワークなどに於ては、その変化に対応した IR 値の変更も必要かとは思われるが、画像の質が急激に変動すると視覚的に見にくくなるといった弊害もあるので、現在のところ手動で IR 値の変動値を設定出来るだけになっている。この値を大きく設定すると伝送容量が毎フレームごとにかなり変化するので、ネットワーク初期設定値として設定している基準再生時間に近い実伝送再生時間の前後で、大きな増減を繰り返してしまう。IR 値の変動が大きいので画像も多少見にくい。逆に、変動値を小さく設定するとネットワークに変化が起こっても反応性が鈍く、暫くの間は実時間性がかなり失われる可能性がある。しかし、画像の変化は少ないので、垂れ流し的な映像の場合はこのほうが適している。面白いことに、これは TCP の「適応再転送アルゴリズム」による再転送のためのタイムアウト時間の算出と同様の問題を掲げており、TCP での「Karn のアルゴリズムとタイマバックオフ」概念を利用することも出来るなど、興味深い所である。

4.3.2 回線幅の測定

ところで実再生時間を制御するために前記のようにインタレース機能を最大限に活用するのだが、サーバ、クライアント間における実効許容回線幅の測定が必要である。それ

は RTT(Round Trip Time)ではなく、任意一点における パケット通過時間である。しかし、本来パケットの始点と終点の間全体に渡っての許容回線幅 情報がリアルタイムに得ることが可能ならば便利なのだが、現在のところそれは次の課題である。よって、クライアント側の処理時間状況を基に算出することにした。

4.4 おわりに

このように、扱う情報容量が膨大になる傾向の中で、それらの情報を共有せず、ネットワークを用いない単体機材内で行う「マルチメディア」の限界が既に見え始めているようである。それとは対照的に今後の高速ネットワークに接続された端末では

- 通信コストの減少
- 通信手段の多様化
- 通信速度の向上
- 移動通信性能の向上

などの要素から今後は透過的に、マルチメディア情報のような膨大な情報をネットワークデータベースや外部機器から入手出来るようになり、そして処理するネットワークの特徴が更に注目されることになると容易に予想することが出来る。

将来の計算機ネットワーク上では、映像音声情報を扱うアプリケーションが更に増加することは明かである。しかしその際ネットワークが混み合い、他の利用者に迷惑を掛けるのではないかと気を使ったり、膨大な情報容量に困ることなしに、快適で便利なネットワークが構築されれば幸せである。それは単に回線幅が広がるだけでは達成出来ないのである。

第 5 章

マルチメディア・メール

電子メッセージの規格『MIME(Multipurpose Internet Mail Extensions)』は、現行のシステムに大きな変更を加えることなくマルチメディア・メールの送受信を可能にするものである。MIME は、インターネットの標準規格を検討する協議会ともいえる IETF(Internet Engineering Task Forces) が作成したものである。ここでは、MIME が考案された背景と、その仕様について述べる。

5.1 MIME の必要性

現在、インターネットで運用中のメッセージ交換システムでは、IETF での検討を経て作成された、『Standard for the Format of ARPA Internet Text Messages(RFC822)[194]』の規格に準拠した電子メッセージの配送のみを保証している。RFC822 では、配送に必要なメッセージのあて先、送信者に関する情報などを宣言する「データフィールド」で構成されるヘッダ部と、メッセージの本文にあたるボディ部の記述形式が定義されている。特にヘッダ部のデータフィールドで宣言される配送先アドレスを指定する形式はBNF 表記法の拡張である EBNF 表記法を用いて特に厳密に定義されている。この規格は、インターネットの母体となった ARPA ネット(米国国防省ネットワーク)の時代に定義された規格で、その当時運用されていたメッセージ交換システムの仕様に合わせて定義されたものである。

しかし、計算機ネットワークの主な接続形態は、ARPA ネット時代の配送遅延が大きい UUCP 接続から現在の配送遅延のない IP 接続へと変化しており、この規格は現状にあったものではなくなっている。たとえば、この RFC822 の定義に準拠した電子メッセージでは、(通常 7 ビットで表現される)ASCII 文字集合 [195] のみを使った記述が義務付けられている。この規格を受けて実装され、現在運用されている多くのメッセージ転送システムでは、7 ビットで表現できないデータの転送に対応できない。

このようなメッセージ転送システムのユーザを考慮に入れると、日本のように(8 ビットや、16 ビットで表現される)拡張文字を使用している国では、転送システムでメッセージが 7 ビットデータとして扱われても不都合が起きないように対策をとる必要がある。日本では、このため漢字などの拡張文字を表現する文字コードとして、コード集合を切り替えて表現する ISO-2022 エンコーディング方式 [196] ¹を採用することとなった。

¹一般に JIS コードと称される

また、現在のメッセージ交換システムでは、テキストメッセージの交換だけでなく、プログラムのソースコードのアーカイブや、実行形式のプログラムの交換も行われるようになってきている。こうした利用形態が一般化したことにより、非テキストのデータの取り扱いも要求されるようになった。

このように計算機ネットワーク環境が大きく変化したことで、RFC822 で定義されている仕様では対応しきれなくなっており、新たなメッセージ記述形式の定義が要求されることとなった。

こうして考案されたのが MIME である。MIME は、現行の標準形式である RFC822 に拡張定義を加えて、再定義した形式規格となっている。MIME メッセージ対応のシステムでは、送信するメッセージに含まれる RFC822 に準拠しない部分に対して転送システムから遮蔽するカプセル化処理を施す。カプセル化されたメッセージの外見は RFC822 形式に準拠しているため、配送システムでは正しく扱われる。こうして配送されたメッセージは、ユーザ・アプリケーションとして実装された MIME メッセージ再生システムに渡すことで復号化が行われる。

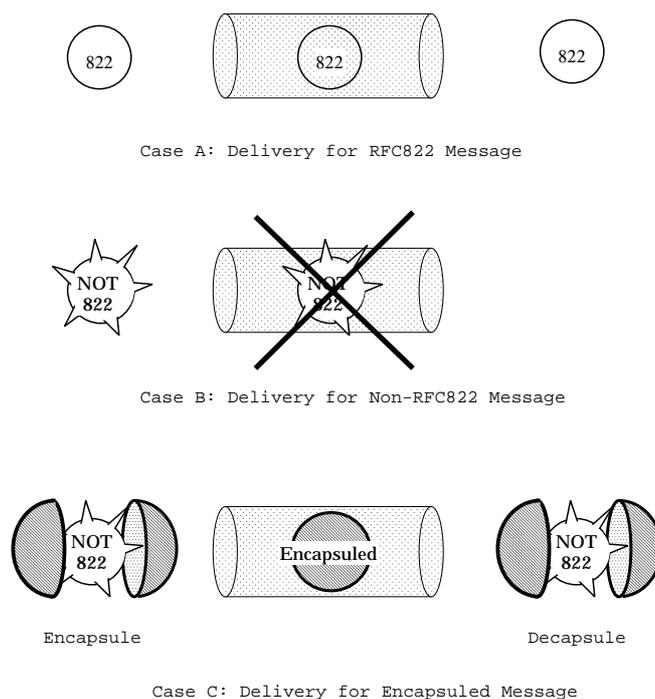


図 5.1: メッセージの MIME 形式化

さらに MIME 形式にされたメッセージは、単に配送システムに対して回避処置を講じるだけでなく、メッセージを構造化されたメッセージにできる。さらに MIME メッセージでは、従来から使われているテキストデータはもちろんのこと、画像データや音声データのような非テキスト型のデータを含むことが可能なため、簡単にマルチメディア・メッセージを実現できる。

5.2 MIME メッセージの拡張フィールド

MIME 形式メッセージの外見上の特徴は、拡張定義されたフィールド群にある。MIME に対応したメッセージ再生システムでは、これらの拡張フィールドから得られる情報に基づいて転送されてきた MIME メッセージの復号化を行う。

MIME メッセージでは、次の 5 つのデータ・フィールドが拡張定義されている。

- Mime-Version: フィールド
- Content-Type: フィールド
- Content-ID: フィールド
- Content-Description: フィールド
- Content-Transfer-Encoding: フィールド

「Mime-Version:」フィールドは、MIME エンコーディングが適用されていることを宣言するとともに、予想される仕様の変化への対応を可能にする目的で設けられた。現在のところ取りうる値は“1.0”のみとなっている。先に述べたように MIME メッセージには、テキスト以外のデータも含むことが可能である。「Content-Type:」フィールドは、メッセージに含まれるデータ型の種類に応じて宣言される。このフィールドの取る値については、後で詳しく述べる。「Content-Transfer-Encoding:」フィールドでは、メッセージの転送時に適用されるデータ変換の変換方式名が宣言される。このフィールドの取る値についても後で詳しく述べることにする。「Content-ID:」フィールドでは、メッセージ中に含まれるデータの参照効率をあげるために、各データを識別するためのユニークな識別子を宣言する、さらにオプションとして、データの内容を簡単に表現するコメントを記述するための「Content-Description:」フィールドが用意されている。

5.3 自動再生

MIME 形式のメッセージでは、データの形式についての記述が義務付けられていることは既に述べた。このことにより、MIME に対応したメッセージ処理システムでは、メッセージの自動再生が可能である。MIME メッセージの再生システムではデータ型 (Content-Type) に対応する再生用アプリケーションが登録されている。再生システムは、ユーザからの再生要求に応じてメッセージからデータを取り出して再生用アプリケーションに渡して自動再生を実現する。自動で行われるのは、データの再生だけでなく、データをファイルとして保管する場合の名前付けの作業も自動化できる (図 5.2)。

5.4 テキストデータへの変換

MIME 形式のメッセージは、配送システム上では、RFC822 形式のメッセージとして認識されることは先に述べた。これは、MIME 対応のシステムには非 ASCII データを NVT

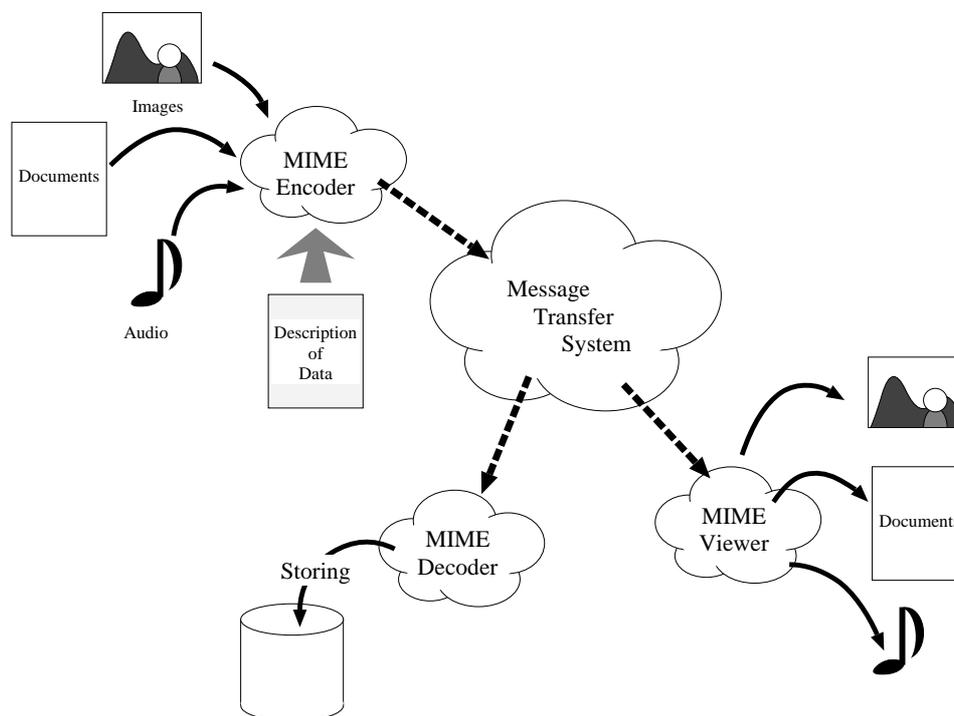


図 5.2: MIME メッセージの自動再生

ASCII 文字集合で記述されたテキストに変換する機能があるためである。NVT ASCII 文字集合とは、ASCII に定義されている文字集合から転送システム上で制御文字として扱われるものを除いたものである。現在の MIME の規格では、任意のデータ型に対して適用できる 2 種類の変換方式が採用されている。

ひとつは、Quoted-Printable 方式である。この方式は、非 ASCII の 1 バイトのデータを「=」につづく 2 桁の 16 進数文字列で表現するものである。この変換方式を適用した場合、1 バイトのデータを表現するのに、3 バイトのデータが必要となるので変換効率が悪い。しかし、主にヨーロッパの国々で使われている 8 ビット系の文字集合のテキストに対して適用する場合には、出現する文字の大部分が ASCII 文字で代用が可能で、非 ASCII 文字データの出現確率が低く、結果としてテキストデータへの変換効率は非常によいものとなる。MIME の規格書では、使用する文字集合が ISO-8859 シリーズ (ISO-8859-*) であるテキスト型のデータにかぎって適用するのが望ましいとしている。また、この変換方式を適用した場合のその他の利点として、テキスト中に出現する不可読な部分がごく一部であるので、この変換方式のデータの復号システムを持たないユーザでもある程度データの内容を推測できるということがある。

もうひとつは、Base64 方式である。この方式では、3 バイト (24 ビット) のデータをまとめて、6 ビット \times 4 のデータとして処理し、4 バイトの文字列で表現する。

5.5 メッセージの論理的カプセル化

MIME 形式のメッセージでは、上記で述べたようなテキスト変換方式を適用することで「物理的な」カプセル化を実現している。さらに、カプセル化されたデータが「どんな種類のデータであるか」、また「どんなデータ変換が適用されているか」という情報を追加して「論理的な」カプセル化も実現している。MIME メッセージ処理システムは、これらの情報をもとにカプセル化されたメッセージを復元できる。

「どんな種類のデータであるか」を宣言するために用意されているのが、「Content-Type:」フィールドである。このフィールドでは、データの表現メディアを示す「コンテンツタイプ」と、データ形式を示す「サブタイプ」が宣言される。さらに、データの再生に必要なパラメータをオプションとして宣言することもある。現在、MIME の規格を定めた RFC1521[174] では、表 5.1 に挙げたコンテンツタイプ、サブタイプの使用が認められている。

コンテンツタイプ	サブタイプ	適用されるデータ
plain	text	通常テキスト
	enriched	リッチテキスト
audio	basic	PCM 音声データ
image	gif	GIF 形式静止画
	jpeg	JPEG 形式静止画
video	mpeg	MPEG 形式動画
application	postscript	PostScript[197] プログラム
	octet-stream	8 ビットデータストリーム
message	rfc822	RFC822 準拠メッセージ
	partial	分割転送データ
	external-body	外部参照データ
multipart	mixed	複数逐次処理データ
	parallel	複数並列処理データ
	alternative	複数データから選択
	digest	複数メッセージデータ

表 5.1: MIME の Content-Type 一覧

「どのようなエンコーディング方式が適用されているのか」を宣言するのが、「Content-Transfer-Encoding:」フィールドである。このフィールドが取りうる値としては、「7bit」、「base64」、「quoted-printable」、「binary」、「8bit」の 5 つがある。「7bit」の宣言は、データ中に NVT ASCII 以外の文字が出現しないことを保証しているだけなので、データに対して特別な復号化の処理は不要である。「base64」、「quoted-printable」の宣言がされたデータは、先に述べた“Base64 テキスト変換エンコーディング”、“quoted-printable エンコーディング”がそれぞれ適用されており、データを取り出すときに復号化の処理が必要であ

る。「binary」、「8bit」はいかなるデータ変換処理も適用されていないことを宣言するものである。この場合、転送システム上でのデータの安全性は保証されない。

5.6 構造化メッセージ

コンテンツタイプ「multipart」は、構造化メッセージを実現するためのものである。multipart コンテンツタイプでは、別のコンテンツタイプをもつデータを複数取り込むことで構造化メッセージを実現することができる。multipart コンテンツタイプには、データの取り出し方により分けられたサブタイプが用意されている。

サブタイプ「mixed」は、単にデータを列挙をするものである。列挙されたデータは出現順に取り出され、再生システムへと渡される。データの再生は、逐次的で前のデータの再生が終るまで、次のデータの取り出しは実行されない。

一方、サブタイプ「parallel」では、列挙するデータの取り出しは逐次的に実行されるが、取り出したデータを再生システムに渡す順序関係はなく、それぞれが独立なものとなる。

これらに対して、サブタイプ「alternative」でもデータが複数列挙されるが、実際に取り出されて再生されるものはただ 1 つだけとなる。これは、データの再生時に、再生システムが扱えるデータタイプが調査され、最初に見付かった再生可能なデータが選択されるようになっているからである。

もうひとつのサブタイプ「digest」は、機能的には前述した「mixed」と同じように複数のデータを列挙するものである。しかし、「mixed」ではデータのコンテンツタイプに指定がなかった場合のデフォルトのコンテンツタイプが「text/plain」であるのに対して、「digest」では「message/rfc822」がデフォルトのコンテンツタイプとなっている点が異なる。

5.7 大量データへの対応

画像や、音声などのデータには、大量のデータが必要である。現在の電子メッセージ用配送システムのなかには、大量のデータ転送を受け入れられないシステムも存在する。

MIME では、「message/external-body」宣言と、「message/partial」宣言を導入することでこれに対応している。

巨大なメッセージを送信する場合、これを配送に支障がない程度の大きさをもつ複数のメッセージに分割し、「message/partial」宣言を使って一連の ID 番号が付けてから送信できる。MIME 対応のシステムでは、これらの一連のメッセージを受けとると、つけられた ID 番号に従って並べ替え、1 つのメッセージとして再構成できる。この機能を利用すれば、一度に大きなメッセージの転送ができないシステムのユーザでも、大きなメッセージを送信できるようになる。

一方、「message/external-body」宣言は、データの参照に必要な情報としてデータの存在する場所、データ提供の手段などをタグとして宣言するものである。MIME 対応のシステムでは、メッセージの参照時にこの宣言部から必要なデータに関する情報を取り出

し、これを使って FTP などのデータ転送システムからデータを取得する。されに取得したデータを宣言部の代りメッセージの一部として挿入する。利用できるデータ提供の手段としては、「AFS(Andrew File System)」、「FTP(File Transfer Protocol)」、「Anon-FTP(匿名ユーザによる FTP)」、「TFTP(認証のない FTP)」、「Mail-Server(電子メールによるデータ転送サービス)」、「local-file」が定義されている。また、この機能を使うことで、転送コストが高い、画像や音声データの無駄な転送も減らすことができる。

5.7.1 多言語ヘッダ

これまで説明してきた MIME の規格は、電子メッセージのボディ部について定められたものである。電子メールに対する MIME の規格には、他に非 ASCII 文字を使用した多言語ヘッダの記述について定義した『MIME(Multipurpose Internet Mail Extensions) Part Two[176]』がある。この規格により、アスキー文字のみに制限されてきた電子メールのヘッダ部の記述にも漢字などの拡張文字の使用が可能になった。

MIME による多言語ヘッダは、WIDE project がパソコン通信ネットワークと共同で実験運用している電子メール用相互接続システムにも導入された。この実験で、MIME の多言語ヘッダが身近なものとなり、その使用が一般化するようになった。

5.8 MIME 対応システム

MIME は、その仕様が検討中の規格であるので、多くのメッセージ処理システムは対応中の段階にある。ここでは、そのなかで代表的なものについて述べる。

5.8.1 MetaMail

metamail は、MIME メッセージを処理する実験用システムとしてベル通信研究所で開発されたアプリケーション・プログラムである。metamail では、MIME で定義されているメッセージ処理にすべて対応しているが、メッセージを配送する機能を持っていないのが欠点となっている。

現在では、電子ニュース処理システムとしては「Gnus」が、電子メール処理システムとしては「VM」、「elm」、「Rmail」などで Metamail を利用して MIME メッセージへの対応がなされている。

Metamail のシステムは MIME メッセージの編集機能を持たないが、多くのシステムで利用できるエディタ「Emacs」用インターフェイス・プログラム、「mime.el」を利用して MIME メッセージの編集も可能である。「mime.el」は、九州工業大学の梅田氏が実装したプログラムである。

5.8.2 MH

MH(RAND MH Message Handling System) は、電子メール処理システムである。MH では、システム内で MIME メッセージ処理を行うので、他のシステムと異なり、単独で

編集を含む MIME メッセージに対する処理が可能となっている。

MH における MIME メッセージの編集機能は、利用方法が複雑で利用されることが少なかったが、“Emacs” 用インターフェイス・プログラム “mhn.el” の実装により、これを簡素化することができた [198]。

5.9 まとめ

このように MIME の規格には、マルチメディア・メッセージを扱うために考案されたさまざまな機能が盛り込まれている。MIME の処理システムの最大の特徴は、メッセージの配送システムに既存のシステムを転用できることである。

しかしながら、MIME のメッセージの編集作業が複雑過ぎるので、MIME メッセージの編集をサポートするアプリケーションに頼る必要があり、その使い易さに MIME の普及がかかっているといえる。

第 6 章

マルチメディア情報のモデル化

6.1 序論

現代社会における情報の価値は大きい。だが、入手した情報は入手したままでは何の価値もなく、それを有効活用してはじめて価値が生まれる。情報は電子化することにより、検索、複製、再編集などの作業による情報の再利用が可能となり、これを有効活用できるのである。

電子化された情報は、電子文書とよばれる。情報の電子化がすすみ、電子文書が身近なものになるにつれて電子文書の構造にも変化が起きている。これまでの平坦な構造を持つ文書の他に、文書を構造化して情報の参照効率を向上させたハイパーテキスト文書などが利用されるようになってきている。また、文書の内容もテキストデータのみから、画像や音声のデータを含むマルチメディア文書が登場しており、これからの電子文書は、マルチメディア文書が中心になっていくと考えられる。その理由として、画像や音声を使ったマルチメディア文書は、単に訴求力に優れているというだけではなく、テキスト文書をも含む多岐にわたる表現が可能であることなどが挙げられる。つまり、情報を最適な手段を使って効率良く伝えることができるのである。

一方、近年の科学技術の飛躍的な進歩により、小型で、高速な計算機を個人でも所有できるようになった。そして、このような個人向けの計算機では、従来の計算機にはなかったグラフィック・ユーザ・インターフェース (GUI)、音声をデータとして記録し、再生するオーディオ・インターフェースなどのマルチメディアに対応した機能を備えているものが主流となりつつある。

また、計算機システムで提供されるサービスの形態の主流は、少数の計算機に機能を集めてユーザにサービスを提供する集中型のシステムから、多数の計算機を通信ネットワークで接続し、個々の計算機の機能を生かすことで運用される分散型のシステムへと移行しつつある。

分散型システムの性能は個々の計算機を結ぶネットワークの性能に大きく依存するので、広域システムでは回線速度の限界の存在によりその大きさに制限を受けていた。しかし、インターネットに代表される広域ネットワーク環境 (WAN) では、その発達に伴ってバックボーンと呼ばれる幹線ネットワークを中心に回線速度の高速化がすすめられ、その利点が活かせるようになった。

このように社会全体の動きとしてシステムの広域分散化と、情報のマルチメディア化が急速にすすんでいるといえる。本章では、こうした社会の動きを受けて情報を広域分散環境で管理するために必要とされる情報のモデル化と、このモデルに基づいて設計したシステムの実装について述べる。

6.2 インターネットにおける情報検索システム

近年の計算機システムのサービスが、従来の集中処理型から広域分散処理型へと移行しつつあることは既に述べた。広域分散型ネットワーク環境でサービスを提供する場合には、各種のデータも分散管理されることが望ましい。ここでは、広域計算機ネットワークの一種であるインターネットにおいて、分散管理されたデータを検索し、提供するサービスについて述べる。

6.2.1 FTP

FTP(File Transfer Protocol)は、ネットワークを利用したデータ転送のためのプロトコルである。FTPプロトコルを使ったデータ提供を行うのがFTPサービスである。FTPによるファイル転送サービスには、認証のためにパスワードを必要とする特定ユーザが利用するサービスと、不特定多数の利用者が認証なしでファイルの提供を受けるサービスの2種類のサービスがある。後者のサービス形態は、特に「匿名FTP(Anonymous FTP)サービス」とよばれ、データ提供サービスとして一般的なものとなっている。また、インターネットのユーザであれば制限を受けずにサービスを受けられるので、データを分散共有するための手段としても利用されつつある。

6.2.2 Archie

Archieは、広域ネットワーク環境におけるデータ検索サービスである。さきに述べたFTPによるデータ提供サービスが一般化するにつれ、FTPのサービスを開始するホストの数も増加した。これにより、ネットワークで提供されるデータの総数も増加し、情報源としての価値が上がったが、それと同時にデータの適切な取得が困難になってきた。このような広域ネットワークに点在するデータファイルの取得を容易にするために、FTPサービスを提供するホストをあらかじめ登録しておき、提供されるファイルのリストを作成して提供するのがArchieのサービスである。Archieサービスの利点として次のようなことが挙げられる。

- データファイルの存在するホストの場所を探索できる。
- データの更新時を比較することで最新版を入手できる。
- 必要なファイルを最も近いホストから入手できる。

6.2.3 wais

wais(Wide Area Information Servers)は、広域計算機ネットワーク上で分散管理されている情報を対象にした検索とデータ取得の機能を実現するシステムである。WAISシステムでは独自の通信プロトコル(Z39.50)で、クライアント-サーバ間のデータ交換を行う。ユーザはキーワードを指定して必要なデータの場所を検索し、さらにその結果をもとにデータの取得することができる。この機能は、ソース(source)とよばれるデータベースを検索して実現されている。ソースは各WAISサーバから取得でき、得られたソースをもとにして別のサーバにアクセスして情報を得るといった操作も可能となっている。ソースには、データを提供するサーバ名、提供されるデータの種類、ソースの管理者などの情報が記録されている。

また、WAISでは画像や音声などのマルチメディア情報も扱える。画像や音声による情報は、ソースにデータファイル名として登録されており、データを取得する場合には、ファイル名の拡張子に応じて再生プログラムの実行が可能となっている。

6.2.4 Gopher

Gopherは、WAISと同種のサービスを提供するものである。WAISがデータを提供するサーバをソースファイル内で静的に記録されていることに対し、データを管理するサーバも動的に検索できるという利点がある。

また、検索できる情報の種類がWAISより豊富で、画像や音声の再生をするなどの機能の他に、外部プログラムの起動や、他のホストへの間接アクセスなどの機能も備えている。

6.2.5 WWW

WWW(World Wide Web)は、gopherと同様のサービスを提供するが、GUI(グラフィカル・ユーザ・インタフェイス)にハイパーテキスト文書(HTML)を導入して情報を提供する点で、Gopherのシステムからユーザ・インタフェイスをさらに高度化したサービスといえる。また、WWWのサーバは、FTPサーバ、Archieサーバ、Gopherサーバなどの既存の情報提供サービスへのゲートウェイとしても機能するので、ユーザにはこれらを統合したシステムに見せることができる。図6.1にWWWのサービスを利用するためのユーザアプリケーション「Mosaic」の画面の様子を示す。

6.3 現行システムにおける問題点

ここでは、広域分散環境でマルチメディア情報を提供する場合に現行のシステムで問題となる点と、新たに要求される機能について考察する。

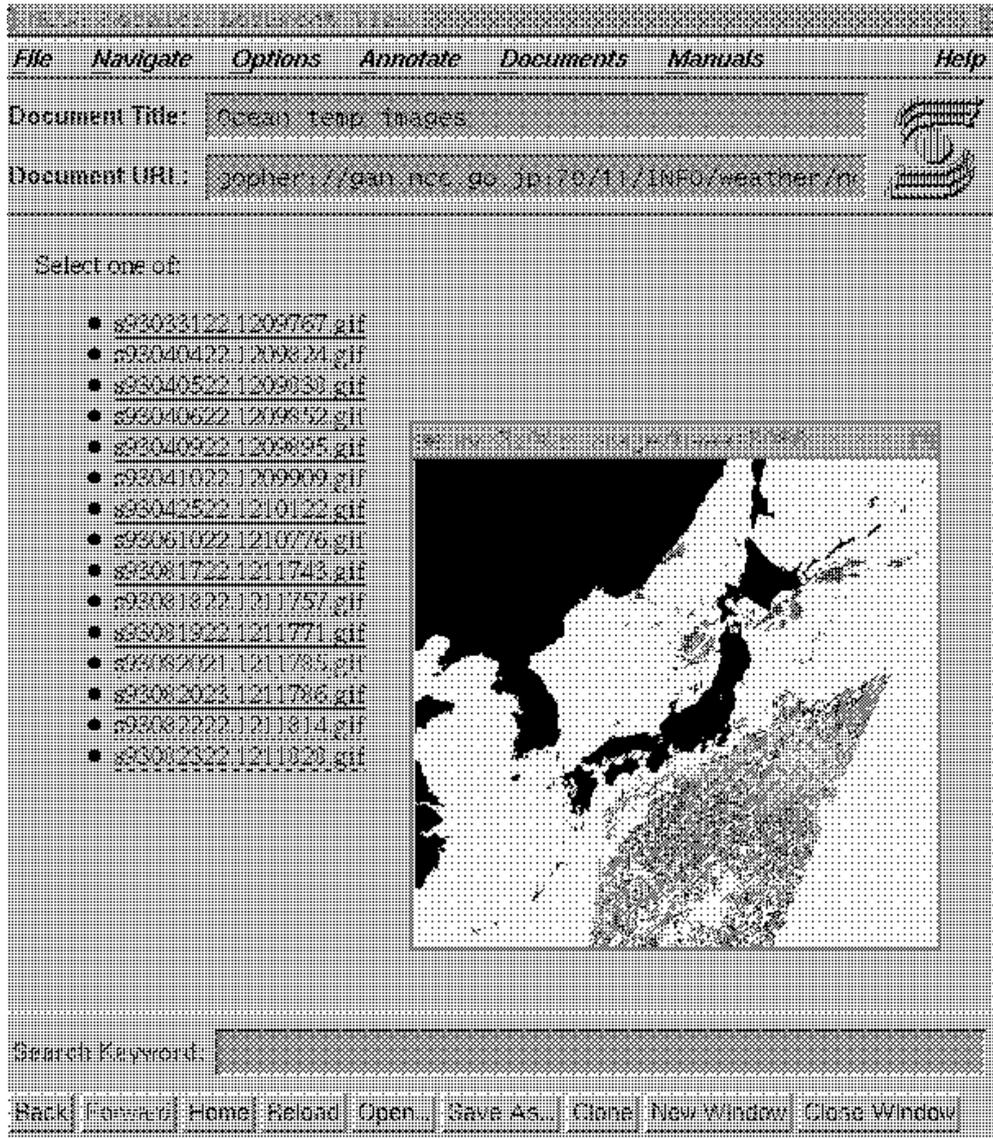


図 6.1: WWW を利用するアプリケーション「Mosaic」

6.3.1 システムの構成上の問題

広域分散環境を提供する計算機ネットワークであるインターネットでは、ネットワーク上の不特定ユーザが利用できる情報提供サービスが開始されている。インターネットで運用されているこうした情報提供サービスでは、情報の所在を示すシステム、情報に対応するデータを示すシステム、データを提供するシステムというようにそれぞれが独立した形態で運用されている。したがって、こうした情報提供サービスを利用するためには、まず最初に情報の所在を調べ、その結果を使ってその情報に対応するデータを検索し、さらにそのデータの提供を受けるという段階的な手順を踏まなくてはならない。

近年では計算機での扱いが可能な表現メディアが増加したことに伴ない、画像や音声などのテキスト以外の表現メディアを含むマルチメディア情報の利用も増加しており、統括的なマルチメディア情報の管理が必要とされている。マルチメディア情報の提供では、データ形式やデータ特性の異なるデータの管理が重要である。しかし、現在インターネットで運用されているシステムは、テキストベースのデータを管理するシステムとして設計されているので、適切な対応ができないものが多い。

また、本来情報の伝達は一方通行であってはならない。しかし、現行の情報管理システムでは、情報を提供する側、提供される側という立場が明確に分けられており、ユーザ間で情報を交換するという目的を果すことができない。

6.3.2 データ転送システムの問題

インターネットのユーザ間で情報交換の手段として主に利用されているのは、電子メールシステムや、電子ニュースシステムなどのメッセージ交換システムである。しかし、現在運用されているメッセージ交換システムでは、メッセージデータの転送をうけもつ部分(メッセージ配送システム)が7ビットからなる文字テキストデータを処理するように設計されていることが問題となる。

異機種ホスト間の接続により構成されるネットワークでは、データが互換であるシステム(オープンシステム)であることが第一に要求される。現行のメッセージ配送システムは、インターネットの前身にあたる ARPA ネットが運用されていた時代に設計されたものがベースになっている。当時、ネットワークのカバーする地域がアメリカ合衆国国内に限られており、また可読テキスト以外のメッセージを交換することがなかったため、システム間で交換する電子メッセージに使用できるデータは、ASCII 文字集合 [195] に含まれるテキスト文字のみという制限が加えられていた [194]。これを受けて制定された配送システムの標準仕様では、ASCII 文字集合に含まれるテキスト文字は7ビットのデータで表現できることから、7ビットのデータ転送に対応できればよいとされた。

現在では、日本を含む英語を母国語としない多くの国々でメッセージ交換システムが運用されるようになっている。これらの国々のメッセージ交換システムでは、その母国語に応じて定義された拡張文字も使用されることが多くなっている。こうした拡張文字の表現には8ビットのデータを要する場合があり、(7ビットで表現しきれない)非 ASCII

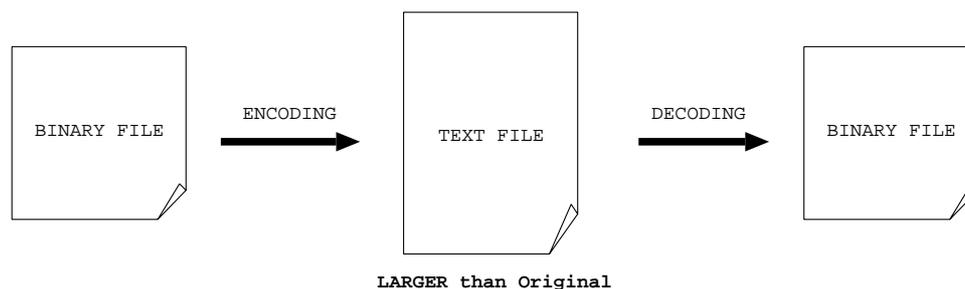


図 6.2: テキスト変換におけるデータ量の増加

文字のデータの交換も要求されるようになった。8ビットのデータ転送を実現するには配送システムそのものの置き換えが必要であるが、システムの規模が大きいインターネットでは、置き換えの手間と移行時の混乱を考えるとその実現は難しい。日本ではこれに対応するため、ISO規格であるISO2022で定められたエンコーディングを用いたコード系 (ISO-2022-JP[196]) に統一することで7ビット系配送システムの問題を回避している。しかしながら、8ビット系文字コードのみを用いるヨーロッパの国々ではこの方法では回避できない。

7ビット系の配送システムが問題となるもうひとつの理由として、交換されるデータの種類の变化が挙げられる。現在のインターネットでは、(バイナリ形式の) 実行可能プログラムや、プログラムのソースコードのアーカイブ等のデータ交換が頻繁に行われるようになり、これに伴って非テキストデータの転送への対応も要求されるようになった。現行のシステムでは、非テキストのデータを一旦ASCII文字集合の文字のみで表現されるように変換(テキスト変換)してから転送し、転送後にデータの復元をする方法が広く用いられている。この方法を導入することで非テキストデータの交換が可能になったが、テキスト変換時にデータ冗長度が高くなり、その結果転送するデータ量が大幅に増加するという現象が発生する(図6.2参照)。

さらに、現行の配送システムでは1メッセージあたりの転送データ量に制限がある。現在日本国内で運用されている電子メール、ネットニュースなどのシステムでは、1メッセージあたりのデータ量がテキストにして数十行程度と制限が加えられているものが多い。これは、間欠リンク回線(一定時間ごとに接続が行われる回線)で接続された(UUCP接続ともいう)ネットワークのなごりである。日本国内ではJUNET協会[199]の一員としてネットワークに参加している組織のなかには、UUCP接続でインターネットに接続されているものも多く、これらの組織宛てに巨大なメッセージを一度に送信すると接続時間内にデータ転送を完了することができず、そのまま放置しておくシステムが作動不能に陥る危険もある。この数十行という量はメッセージがテキスト文書の場合であれば十分な量であるといえるが、先に述べたような非テキストデータを含むメッセージの交換では、転送するデータ量がテキストメッセージよりも大量であるために、システムの許容範囲を超える事態が頻繁に起きてしまう。

システム相互で交換されるメッセージに非テキストデータが含まれるようになり、その種類が多様化したことで新たに発生した問題もある。メッセージに含まれるデータの形式についての記述が不足していることである。転送されてくるメッセージに、非テキストデータが含まれる場合には、受信者はテキスト変換されているデータをデコードして取り出し、場合によっては、データをその記述形式に対応した再生システムに渡して再生する必要がある。このようなメッセージでは、メッセージを受信したユーザがそのデータの記述形式についての知識を持っていない場合には、データの再生を行うことができない。

6.3.3 マルチメディアメッセージの交換

これまで述べてきたのは、システムがこれまで通りのテキストデータ中心の利用形態をとる場合についてである。ここでは、複数の表現メディアを扱うことのできるマルチメディア対応のメッセージ交換システムで要求される機能について考察する。

まず、マルチメディア対応のメッセージを交換システムでは、データの転送を従来のようにメッセージ単位に行うのではなく、それぞれのメッセージに含まれる表現メディア単位にして行う必要がある。その理由として、表現メディアごとのデータ特性の違いがまず挙げられる。たとえば、転送するデータに符号圧縮を適用して転送する場合に、テキストデータと画像データとでは、データの冗長性の性質が異なるので、テキストデータに有効な圧縮アルゴリズムを適用した場合には、画像データに対しての効果は期待できない。つまり、テキストデータと画像データの双方を含むメッセージを効率よく転送しようとする場合には、メッセージをメディア別に分離してそれぞれに最適な圧縮アルゴリズムを適用しなくてはならないことになる。

もうひとつの理由として、データ量の違いがある。一般に、画像データや音声データのデータ量は、テキストデータと比較すると非常に大きい。画像や音声のデータがメッセージに含まれる場合には、テキストデータのみが含まれる場合に較べて大量のデータ転送が必要となる。したがって、画像や音声のデータは必要になってから送るようにすれば、無駄なデータ転送を減らすことが可能である。

メッセージにして伝えようとする情報があると、文章、絵、あるいは表やグラフなどその表現の仕方は無数に存在する。これらのすべてを統合してメッセージとして送信した場合に、メッセージを受信したユーザは、複数の表現のなかから状況に応じて最適なものを選択することでメッセージを効率よく理解できると考えられる。このとき受信者が選択するのは、「意味レベル」では同一の情報を表現する表現(プレゼンテーション)の1つである。たとえば、「今年は、米が不作である」ということを伝えるのに、「荒れた田畑の写真」と、「今年の収穫量を示した表」を使うとすると、どちらのデータも表現メディアが同じ画像データであるにも関わらず、受信者がそれぞれのプレゼンテーションから受ける印象は異なるものになる。しかし、どのプレゼンテーションが情報をより効率的に伝えることができるのかを事前に判断することは困難である。また、メッセージが個人に宛てたものではなく、ニュースの記事など公共性の高いメッセージである場合に

はもっと困難である。不特定多数の相手に最適なプレゼンテーションをあらかじめ選択してメッセージを作成することは不可能であるからだ。このようなメッセージは、「対話型メッセージ」とよばれハイパーテキスト文書はその一例である。

6.4 マルチメディア・メッセージ交換システム

ここではこれまで述べてきた現行システムの問題点とマルチメディア情報の表現に要求される機能を考慮に入れ、マルチメディア情報の表現を実現するシステムをモデル化を行う。

6.4.1 意味指向表現モデル

マルチメディア情報を扱うシステムでは、次のような要求をみたしている必要がある。

- 表現メディアごとのデータ処理
- 対話による表現メディアの選択とデータの取得

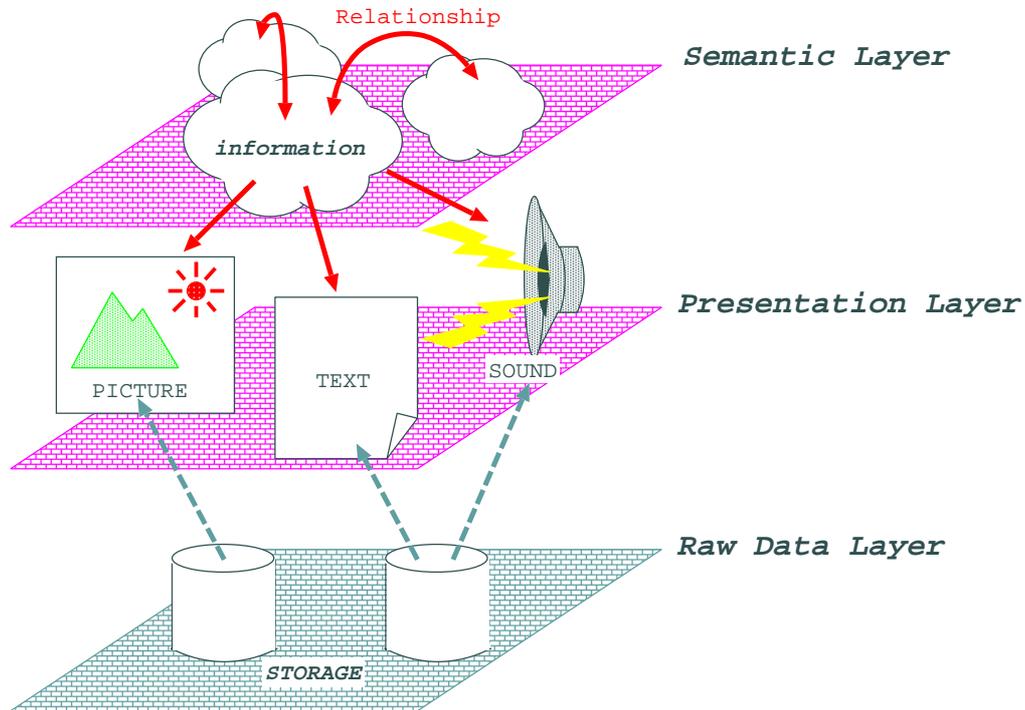
以上のことを念頭において、これらの問題点を解決し、要求をみたすシステムのモデル化を試みる。

表現メディアごとのデータ処理、および対話的なデータ参照を可能にするためには、メッセージ本体と、データは切り離すことが必要であると考えた。つまり、受信者に配送されるメッセージには、データの実体は含まれることはなく、表現メディアへの参照のための ID のみが含まれるようにするのである。メッセージの受信者は、データへのポイントが記述されたメッセージをもとに適切な表現メディアを選択し、別に用意する参照システムにアクセスしてデータの取得を行う。

ところでデータの記述形式は、表現メディアで一意に定まるものではない。たとえば静止画像というメディアを例にとっても、GIF、JPEG といった複数の標準形式が存在することは既に述べた。しかし、ユーザがこうしたデータ形式について考慮しなければならないのは再生システムを選択する場面においてのみである。そこで、モデル化するシステムでは、メディア再生システムとデータの管理システムを別々に用意することにした。

こうして構築されるメッセージ管理システムは、図 6.3 に示すような階層構造をもつモデルで表現される。最上位の階層である「意味層 (semantic layer)」では、メッセージで表現される「情報 (information)」が管理されることになる。また、その下の階層である「表現層 (presentation layer)」で「情報」の表現に必要な各種表現メディアによる「表現 (presentation)」が管理される。モデルには、さらに下層の表現メディアに使われるデータの実体を管理する「生データ層 (raw data layer)」が存在するが、この部分は情報を提供するシステムによってユーザに対して遮蔽されており、ユーザにとってはあたかも表現メディアが最下層の操作の対象であるかのように見える。

このシステムモデルでは、情報の管理・表現が「意味層」の「情報」を中心に行われている。こうした特徴からこのモデルを「意味指向情報表現モデル (Semantic Oriented Information Representation Model)」とよぶことにする。モデルでは、マルチメディア



Semantic Oriented Information Representation

図 6.3: 意味指向情報表現モデル

メッセージを複数の表現メディアデータからなる集合体としてではなく、表現メディアを使う「表現」の集合体として扱っている点が特徴となっている。また、ユーザが各メッセージの表現を要求する場合に、表現メディアを使った結果得られる「表現」という抽象的な概念をその処理単位とすることも特徴といえる。

6.5 メッセージ交換システム的设计

実装するメッセージ交換システムは、前述した「意味指向情報表現モデル」に基づいて実装する。

システムは大きく分けて、「メッセージ配送部 (Message Exchanger)」、「メッセージ参照部 (Message Viewer)」、「データ・サーバ部 (Data Server)」の3つの部分で構成される(図 6.4参照)。

「メッセージ配送部」は、マルチメディア情報を含むメッセージの配送を受け持つ部分である。本システムにおいて配送されるメッセージは、データの実体をメッセージの外部に持ち、表現したい情報ごとに識別子 (情報-ID) が宣言されている。

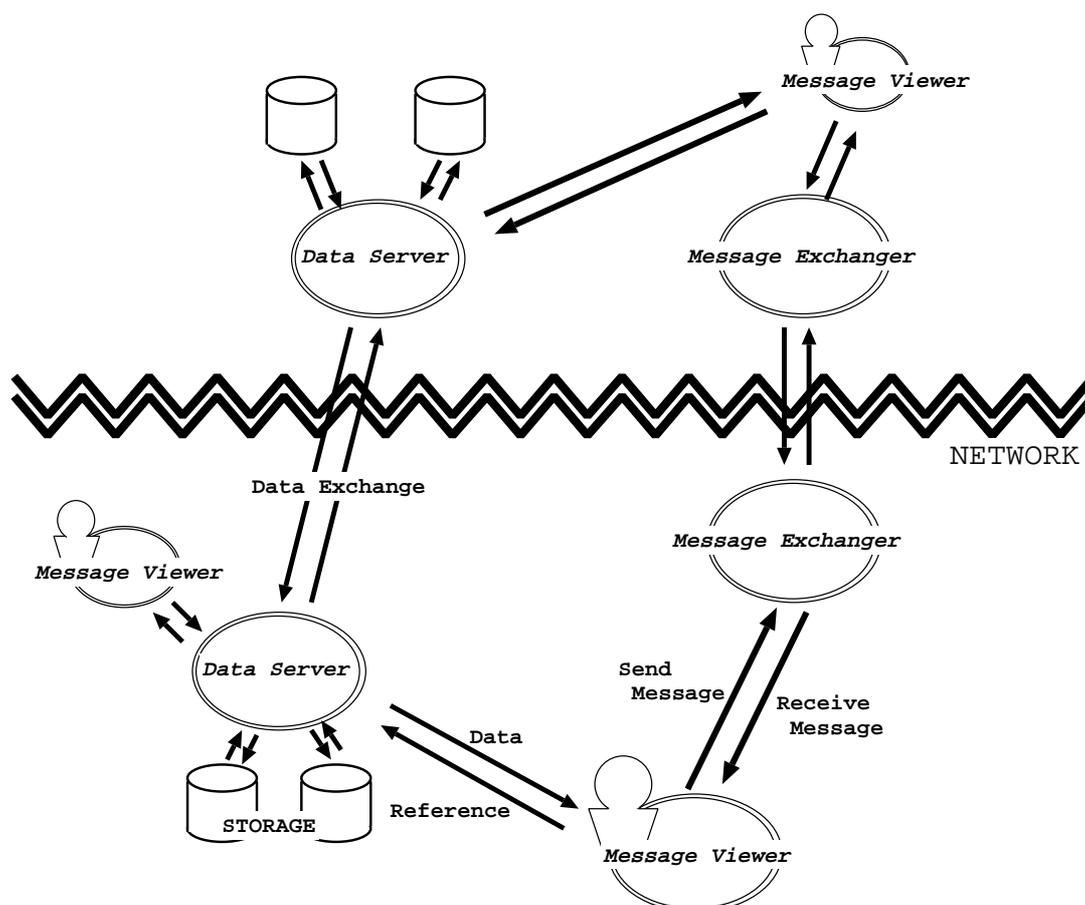


図 6.4: システム概観

「メッセージ参照部」は、ユーザからの要求に応じてメッセージを選択し、その再生する部分である。この部分が、「意味指向情報表現モデル」における「表現層」を実現している。「メッセージ参照部」は、ユーザによって選択された「表現」に必要なデータをメッセージ中に含まれる情報-ID を使って「データ・サーバ部」と通信することでデータの取得して、そのデータを再生システムに渡して「表現」を実現する。このときユーザは、表現に必要なデータについて直接関与する必要はなく、端末で実現される表現を選択することのみが要求される。「メッセージ参照部」は、メッセージ交換システムにおけるユーザ・インタフェース部としての役割をもっている。

「データ・サーバ部」は、「メッセージ参照部」から要求される「表現」の実現に必要なデータを提供する。その役割は、「意味指向情報表現モデル」における「生データ層」と「表現層」と結ぶものである。「データ・サーバ部」におけるデータの管理は、データの実体のファイル単位ではなく、情報-ID ごとによって行われており、情報-ID と表現メディアを指定されると、必要なデータを再生システム的环境にあわせて提供する仕組みとなっている(図 6.5 参照)。

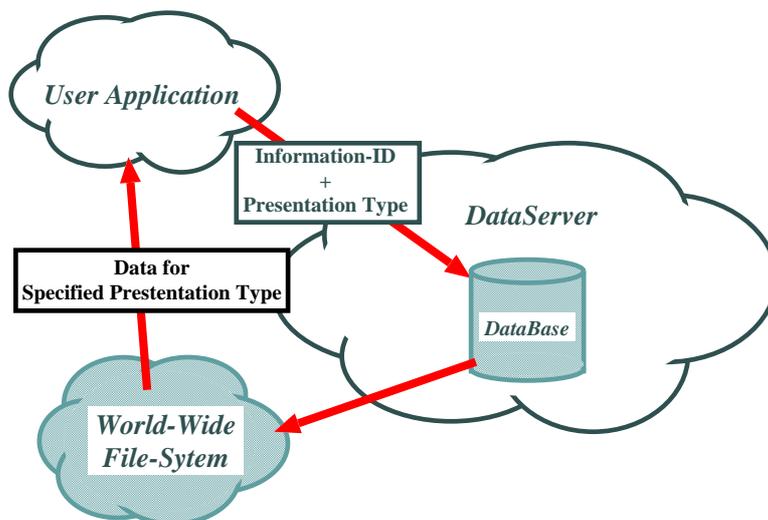


図 6.5: データサーバへのアクセス

ユーザによるメッセージの参照要求に対して、システムは以下のように動作する。

1. ユーザは、「メッセージ配送部」により配送されたメッセージを「メッセージ参照部」経由で参照する。このとき、メッセージに含まれるメッセージで提供されている「表現」の中から好みの「表現」を要求する。
2. 「メッセージ参照部」は、要求された「表現」に必要なデータを情報-ID と表現メディアを指定して「データ・サーバ部」に請求する。このとき、「データ・サーバ部」は「表現」の実現する表現メディアのデータの中から、再生システムが扱えるデータ型のデータを選び出して提供する。
3. データを受けとった「メッセージ参照部」は適切な再生システムを自動起動して、「情報」の「表現」を実現する。

6.6 システムの実装

実装したシステムでは、メッセージの記述フォーマットに MIME 形式 (前章を参照) を採用し、既存の配送システムを「メッセージ配送部」として利用している。また、「メッセージ参照部」には MIME 対応のメール処理システムに「データ・サーバ部」との通信機能を追加するための改造を加えて実装した。

「データ・サーバ部」は、インターネット上のユーザからの利用を考慮し、サーバプログラムとして実装した。「データ・サーバ部」のインターフェイスは、異機種システム間の通信を容易にするために、テキストベースのコマンドインタプリタとなっている。「デー

「データ・サーバ部」へのデータ請求は、コマンドインタプリタが受け付ける以下にあげる簡単な命令で実行できる。

SET コマンド データ提供の条件を設定する。設定が可能な条件としては、以下のようなものがある。

1. 表現 (データの再生) が可能なメディア名
2. デフォルトとして提供される表現メディア
3. 各表現メディアにおけるデータフォーマットの優先順位

GET コマンド データ提供の請求を行う。引数として情報-ID と表現メディア名をとる。さらにオプション引数としてデータ型も指定できる。データ型を指定した場合、上記の SET コマンドによる指定の手順が省略できるので、短時間の接続で利用する場合の便宜を計るための機構である。情報-ID に対して提供可能なデータがあった場合には、そのデータ型とサイズを報告し、データの転送を開始する。指定された情報-ID が登録されていなかったり、提供可能なデータがない場合には、エラーとして報告する。

ここで指定する表現メディア名には、MIME システムからの移行を考慮に入れ、MIME の規格で表現メディアを表す「コンテンツ・タイプ名」に合わせた。

6.6.1 データ・サーバ内部での処理

本システムでは、「データ・サーバ部」の内部でデータと表現メディアとの関連づけを行っている。ここでは、この関連づけの処理について述べる。

「データ・サーバ部」はその内部に情報-ID と、これに対応するデータの一覧表をデータベースに保存している。データの一覧表は、図 6.6 に示すようなデータ構造を持っている。

また、情報 ID によるデータの検索の対象になるのは、指定されたデータの記述形式だけでなく、指定されたデータへの変換が可能なデータ形式も含まれる。この場合には、データ変換を行った後にデータの提供が行われる。

6.7 メッセージの記述形式

本システムでは、マルチメディア・メッセージの記述に MIME 形式を導入している。「データ・サーバ部」からのデータ取得の宣言は、データがメッセージ外部に存在することから、MIME の外部参照の宣言に拡張を加えて用いる (図 6.7 参照)。例に挙げたメッセージ中で宣言されている MIME の宣言フィールドのうち、access-type、x-infoid、x-dtype の各オプションで独自の拡張を加えている。

「access-type=x-dataserv」は、データの取得にデータサーバの利用を宣言するものである。この宣言により「メッセージ参照部」はデータサーバへのデータ請求を行うこ

```
struct MediaDesc {
    DataFmt dataFmt;          /* データの記述形式 */
    DataID  dataId;          /* データファイルの内部表
現 */
}

struct InfoDesc {
    int    filledCells;      /* 登録済の表現メディアの
数 */
    struct MediaDesc mediaList[MAXCELLS]; /* 表現メディアのリスト */
}
```

図 6.6: 表現メディアの一覧表を実現する構造体

とができる。「x-infoid」と「x-dtype」は、「データ・サーバ部」からデータ提供を受けるときに使われ、「情報-ID」、「データフォーマット」がこれらのフィールドで宣言されている。ここに示したメッセージ例では、(3種類の表現をする)各パートがあるが、データの取得に使う x-infoid は、同一のものである。しかし、データサーバからデータを取得する時点で x-dtype を共に指定すると、異なる型のデータを取り出すことができる(図 6.8、図 6.9参照)。

```
To: testusers
From: Shingo Fujimoto <shingo@cs.uec.ac.jp>
Subject: TEST message
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="----- =_aaaaaaaaaa0"

----- =_aaaaaaaaaa0
Content-Type: message/external-body;
    access-type="x-dataserv";
    x-infoid="shingo9402081053bigbird@cs.uec.ac.jp";
    x-dtype=jpeg

Content-Type: image/jpeg
Content-ID: <4314.760727377.1@cs.uec.ac.jp>
Content-Description: Data in JPEG format

----- =_aaaaaaaaaa0
Content-Type: message/external-body;
    access-type="x-dataserv";
    x-infoid="shingo9402081053bigbird@cs.uec.ac.jp"
    x-dtype=gif

Content-Type: image/gif
Content-ID: <4314.760727377.2@cs.uec.ac.jp>
Content-Description: Data in GIF format

----- =_aaaaaaaaaa0
Content-Type: message/external-body;
    access-type="x-dataserv";
    x-infoid="shingo9402081053bigbird@cs.uec.ac.jp";
    x-dtype=plain

Content-Type: text/plain
Content-ID: <4314.760727377.3@cs.uec.ac.jp>
Content-Description: TEST format

----- =_aaaaaaaaaa0--
```

図 6.7: システムで交換されるメッセージの例

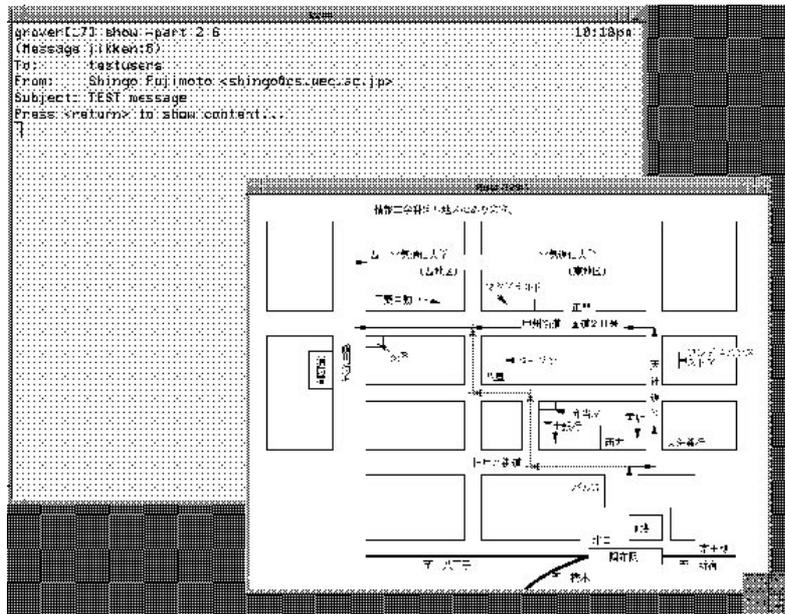


図 6.8: 実行例 (画像データとして表示)

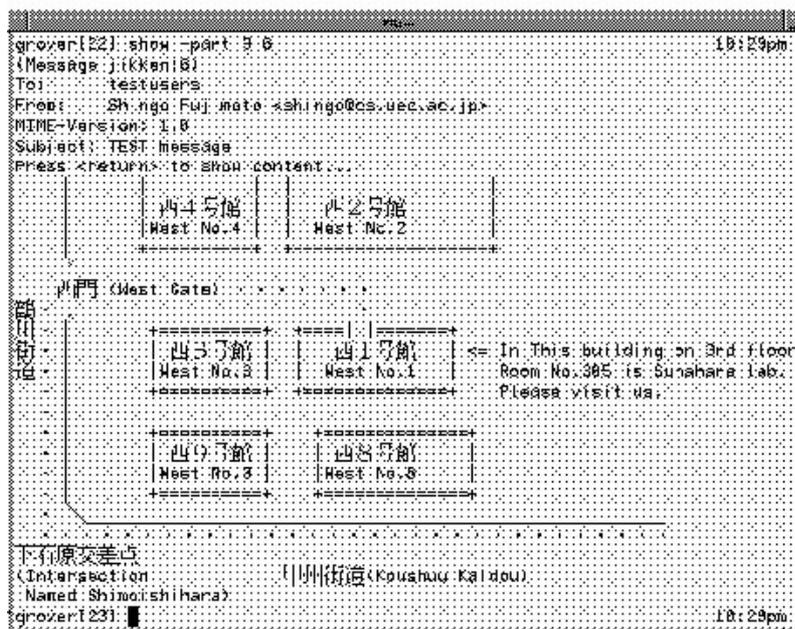


図 6.9: 実行例 (テキストデータとして表示)

メッセージの形式が MIME 準拠であれば、「メッセージ配送部」として既存の配送システムをそのまま利用でき、さらに「メッセージ参照部」として MIME 対応のメーラ (電子メールシステムのインタフェース) に改造を加えることで対応できる。

このようにして、「データ・サーバ部」の実装とメーラのインタフェース部分の改造という最小限のコストで、「意味指向情報表現モデル」を実践するマルチメディア対応メッセージ交換システムの実装が実現できた。

6.8 評価

本章では、既存の従来型メッセージ交換システムと、MIME 対応のメッセージ交換システムとの比較、および従来の MIME に対応しているシステムと今回実装したシステムとの比較を行い、実装したシステムの有効性について考察する。

6.8.1 既存のシステムとの比較

- マルチメディアへの対応

既存のシステムでは、テキスト以外のメディア用データは、単なるデータとして送受信されることになる。一方、MIME に対応したシステムでは、マルチメディアのデータを含むメッセージの送受信が可能となっており、各メディア用データ自動再生も可能である。

- データの転送

データの記述形式が多く、互換性が少ない画像データの送受信では、複数種のデータがメッセージ内に含まれていることが望ましいので、必要なデータのみを選択し、不要なデータの送信を抑制することが可能な外部参照データへの対応はマルチメディア・メッセージ交換システムには必要不可欠な機能である。既存のシステムでは、こうした必要のないデータの送信を抑制することができないが、MIME 対応のシステムであれば、外部参照データの宣言を行うことで、データは必要とされた場合のみ転送されるようにすることができる。

6.9 MIME 処理システムとの比較

今回の実装では、メッセージの記述形式に MIME を拡張定義したものを使用している。そのため、ユーザインターフェイスに変化はないが、データの管理や転送などの目に見えない部分で改良がされている。

- マルチメディアメッセージの参照

実装したシステムでは、対話的な表現の選択も受け入れられるようになっている。しかし、メッセージ中の適切な表現を対話的に選択するという機能が MIME の規格のなかには定義されていないので、MIME 用のインターフェイスが提供されているだ

けのアプリケーションでは、対処できていない。MIME の規格に表現を選択する機能が採用された場合には、「データ・サーバ」によるこの機能を活かすことができる。

また、「データ・サーバ部」では請求された記述形式のデータが存在しない場合でも、データの記述形式の変換プログラム等を用いて必要なデータを作成することもできる。この機能により、サーバ側に(あらかじめ)用意されていない記述形式のデータを(可能であれば)記述形式を変換して提供できる。

- 情報の管理

情報の管理作業には、データの登録、データの提供、データの削除などがある。本システムにおいてこれらの作業は、「データサーバ部」により「情報-ID」で行なう。既存のシステムでは、データの登録がファイル単位で行われていたので、データを提供する場合には、データ取得の制御もファイルあるいはファイルシステムで実現する必要があった。一方、このシステムでは表現に結びついた表現メディアがまとめられているので、これらのデータの整理をする場合には、処理の対象を把握しやすく、分散管理される情報の管理作業における負担を軽減できる。

6.10 まとめ

本研究では、マルチメディア情報の効率的な表現法について考察し、マルチメディア情報による効率的な表現を可能にするシステムのモデルとして、「意味指向情報表現モデル」の提案を行った。

また、「意味指向情報表現モデル」に基づく応用システムとしてインターネット上での稼働を想定したマルチメディア・メッセージ交換システムを設計し、実装を行った。

実装したシステムでは、マルチメディア情報の管理と提供について従来のメッセージ交換システムにはなかった新たな機能を実現できた。

6.11 今後の課題

メッセージ交換システムの実装では、現行システムからの移行コストを軽減することを最優先としたため、MIME 規格の範疇から出ることができなかった。今後は、MIME 規格により大規模な拡張を提案することで、より利用価値のあるマルチメディア情報管理システムをめざしたい。

また、広域分散環境ではデータファイルを取得する手段として、ネットワークにまたがる分散ファイルシステムを利用できる。本研究との関連は薄いですが、分散ファイルシステムは広域分散環境における情報の管理を簡素化する技術のひとつである。分散ファイルシステムの研究は現在盛んに行われているので、その成果は積極的に取り入れていきたい。