

プログラム間データ通信方式の検討
— データ定義プログラム間通信の基礎的技術 —

米 澤 保 雄

帝人株式会社システム技術研究所

現) 愛知技術短期大学電子工学科
情報処理コース

ハードウェアの高機能化に伴ってマルチメディア・データの為の情報処理並びに通信処理の技術開発が重要となっており、特に、アプリケーションレベルにおけるプログラム間のデータ通信では従来のような「送受信双方でのデータ構造の変換手続きのプログラム」の用意を必要としないデータレベルでの互換性保持の処理技術が必要である。本稿では通信されるデータそのものにデータ構造定義を持たせる事によって、プログラム間データ通信の変換前と変換後のデータ長の差異を生じる様な場合等への対処を可能とする通信方式に関して報告する。

A COMMUNICATION CONTROL METHODS
FOR PROGRAMS

Yasuo Yonezawa

System Research & Development Institute
TEIJIN Ltd

Present Address:
Course of Information Processing
Department of Electronic Engineering
Aichi College of Technology

50-2, Manori, Nishisako City, Gamagoori-City
Aichi-Pre, 443, Japan

In order to the development of computer Hardware, the communication for computer Networks were obtained Multi Media data processing capacity. For this development, Parallel data processing & Program Data communication were very important technology. Specially, Data communication of Program for Program were Key technology for Multi Media data processing. In this paper, We were described with the data conversion of two way communication at several programs. This data conversion technique were consisted at data definition program on individually computer linked communication network. According to this program, the difference of data structure depend on several computer were treated as same data structure.

1. まえがき

従来のプログラム間における構造と意味を持つデータ通信を送受信する方式は送信側および受信側共に送受あるいは受信するデータの構造を認識している事を前提としており、かつ送信側および受信側で行われるデータ交換に関しては機種や中央処理演算装置(CPU)の違いにより生じるデータ表現の違いへの対応であるビットオーダやバイトオーダの変換などのデータ交換のみについて行われていた。

従って、プログラム間で送受信されるデータに対してデータ交換を行う際に交換前と交換後でデータ長に差異が生じる様なデータ交換を行う事が出来ないと言う不都合があった。マルチメディア通信の伸展に伴って、各アプリケーション・プログラム間でのデータ通信の複雑性は高度化する傾向にありデータを送受信する際にデータ長に変更がないデータ交換のみならずデータ長に変更が生じる様なデータ交換をも可能とするプログラム間の通信の仕組みを必要としている。

本稿では実際のデータ交換を伴うプログラム間通信の為に以下の方式を検討した。

即ち、プログラム間で単なるビット列としてのデータではない構造と意味を持つデータを送受信するに当たって、送信するデータの構造を定義する手段と、データの各構成要素に対してその送信時にそのデータタイプ(型)に応じたデータの変換を行う変換手続きを定義する手段を持ち、これらの手段によって定義された構造を持つデータを送信する際に各種類のデータに対しその送信用のデータ交換手続きにより変換されたデータの構造の定義を送信し、受信する際には受信した変換されたデータの構造の定義を受信用のデータ交換手続きにより送信されたデータの構造を認識し、その構造に従って受信したデータを受信用変換関数により変換してデータを受信する方式である。

本方式はお互いに通信を行い、同期を取りながら処理を進める必要があるプログラムに利用する事が出来、又、異なる機種、CPU及びコード体系を持つ計算機上で動作するプログラム間の通信に利用出来る。よって、マルチメディア処理及びネットワークにおける各種データの相互伝達処理に効果が期待出来る。

2. 通信方式の基本スキーム

送信するデータの構造を定義したものをデータ構造情報として定義し、送信側が送信を要求した時点において送信を要求したデータ(元データ)と共に元データのデータ構造定義情報も併せて送信する事により、受信側が受信したデータの構造を識別する事を可能にする事を基本とするのが本通信方式である。そして、元データの各構成要素となる各データタイプに対して、データタイプ毎に送信用及び受信用の変換手続きの定義が可能であり、データ

タイプと変換手続きの対応はデータ交換情報テーブルに保持する事、即ち、送信時には元データと元データのデータ構造定義情報とをデータ交換情報テーブルに記述して送信用変換手続きにより変換した後に送信し、受信時には受信したデータ構造定義情報とデータを受信用変換手続きにより変換した後にプログラムに渡す事が出来る。

元データのデータ構造定義情報においては、その各構成要素に対してそのデータタイプと元データの現在のデータ長に加えて最大データ長を合わせて定義する。又、送信用・受信用の変換手続きはその引数としてデータへのアドレス・現在のデータ長と最大データ長を取る事によって、データ長に変動がある場合の変換を可能として、機種・OSやプログラム毎のデータ処理形式の差異によって生まれるデータ通信の際に必要な変換を通信処理内で行う事が出来る。さらに、実際の手順に照らして説明する。

通信手順

ステップ1] データの構造記述

データ交換テーブルとデータ構造定義情報を送受信するデータに合わせて作成する。データ交換テーブルにはデータタイプ毎の送信用と受信用の変換手続きが設定するものであり、データ構造定義情報は送信するデータの構造が記述される。

ステップ2] データの変換処理

データ構造定義情報にその構造が定義された元データを送信する際にその各構成要素は、そのデータタイプの送信用変換手続きに従って変換される。さらに、データ構造定義情報自体が送信用変換手続きによって変換される。

ステップ3] データの送信処理

変換された元データ及び構造定義情報が送信される

ステップ4] データの受信処理

受信側は受信したデータ及びデータ構造定義情報について、先ずデータ構造定義情報を受信用変換情報によって変換する。次に、変換したデータ構造情報から受信したデータの構造を認識し、その構成要素のデータタイプに従って受信したデータを受信用変換情報によって変換して通信を完了する。

以上が本方式の基本的な処理であるが、本方式のスキームから理解される様に、ネットワーク上に結ばれるコンピュータの種類並びにその通信プロトコルの種類に起因する差異などは全てデータ定義上の問題として処理される事の特徴としているので適切なデータ交換処理速度が保障されるCPU上で実行される限りにおいて効果的通信が実現出来る。

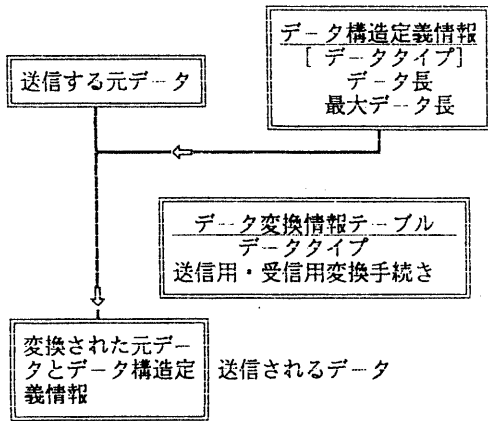


図1：送信の手順

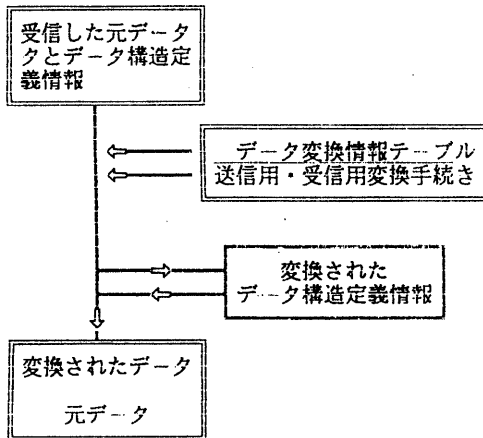


図2：受信の手順

3. 本通信方式の機能

本通信方式は以上述べた様に、送信及び受信時にそのデータタイプに応じた変換情報を呼び出し、この情報によって計算機やCPUの異なる計算機上で動作するプログラム間で通信を行う場合においてもそのデータ表現を変更する変換手続きを用意する事によって通信する相手のプログラムが動作している計算機やCPUの種類などを意識する事なくプログラムを作成する事が可能となり、ネットワークに接続される各種コンピュータ間及びそのプログラム間での通信を可能とする機能を提供する事が出来、アプリケーション開発には効果的である。

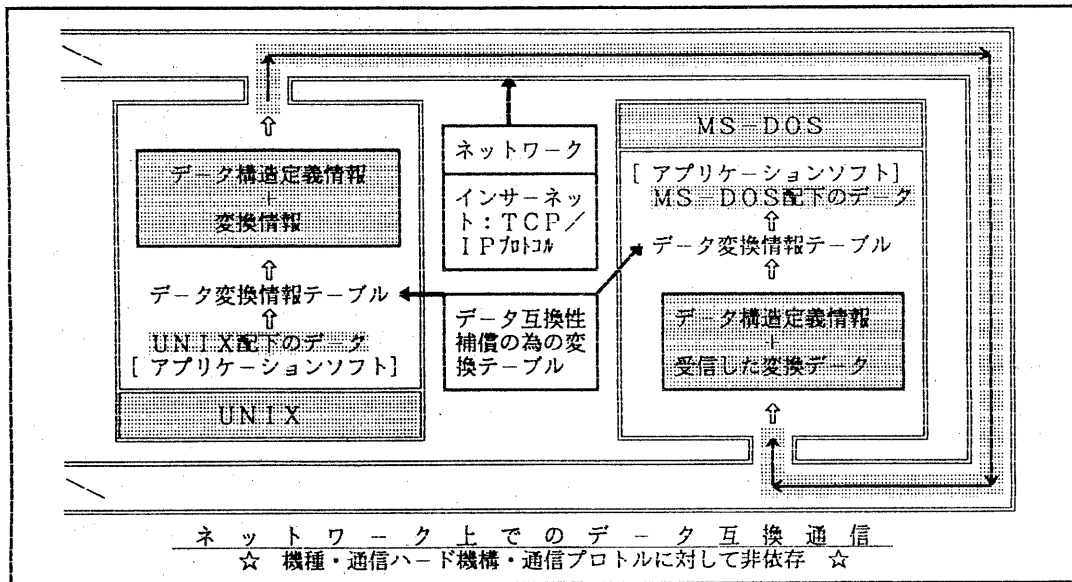
又、データの種類として整数や浮動小数点の様にそのデータ長に変更のないデータタイプだけではなく、文字列の様な計算機のコード体系によってそのデータ長に変更のあり得るデータタイプの変換を互換処理する事を可能とする機能を持っている。

よって、現在のUNIX (WS:ワークステーション) の主流ネットワーク処理方式であるTCP/IPによるクライアント・サーバモデルが提供するネットワーク上に接続されるコンピュータのリモート運用によって実現されている異機種間接続で補えない様な場合、即ち、クライアント・サーバモデルではサーバ側のCPUを用いたデータ処理を主としているものをさらに高度なクライアント側(利用側)のCPUをも使用したネットワーク透過性を実現する事も可能であり、サーバ側で処理された情報データをクライアント側で再処理する為などの通信にも利用可能である。

特に、これからのマルチメディアの情報処理においては複数の種類の情報を効率よく又複雑な処理を混乱なく通信する為の機能を実現する基本的な処理方式と言える。

即ち、マルチメディア通信に係わる情報処理の実現に必要である大容量のデータ通信のキーテクノロジーのデータ圧縮技術にもその圧縮アルゴリズムが本方式のデータ定義テーブルとその変換プログラムに反映されてさえいれば一般の情報におけるコンピュータ間でのデータ構造の差異の変換と同じく処理出来るものである。よって、各マルチメディア情報処理のアプリケーションプログラムは機種間のデータ差異を考慮せずに、又、通信によるデータ圧縮に係わるデータ圧縮及び解凍(伸展)などの処理を意識する事なく作動する事が出来る。

さらには、同じコンピュータ上に同居する異なるアプリケーションに必要なデータ変換にも転用する事が可能であり、アプリケーション毎のデータ変換用のプログラム作成の省力化にも効果を期待出来る。



4. データ種に関する非依存性

本通信方式は各通信相手であるハードウェア及びOSの違いによって生じる差異に関しての互換性を補足するものであり、各機種又はOS配下においてそのアプリケーションソフトが処理出来る互換性を保障する為のデータ定義である。よって、データ種における差異を意識する必要はアナログデータとデジタルデータ等のデータ形式の差異が必要である様な特別な場合を除いては基本的にはない。即ち、送・受信においてA/D変換を必要としない場合である、即ち、送信側及び受信側の双方共にデジタル又はアナログデータの何方か一方のデータ形式がそのアプリケーションソフトでの処理可能データ形式であるならば本方式は有効である。ただし、A/D変換を必要とする通信過程においてはハードもしくはソフトウェアによる変換を各ネットワークされている機種毎に処理する必要が生じる。

5. マルチメディア情報への展開

通常のプロセス間通信の手法には共有メモリ・メッセージパッシング及びリモートプロシージャコール(RPC)などが既存のものとして挙げられる。

- ①共有メモリではプロセス間に共通なデータ領域であるから、あるプロセスが共有メモリにデータを保持した事を他のプロセスに通知する為のプロトコルが別に必要となる。
- ②メッセージパッシングではプロセス間に開設した通信チャンネルを介して通信するものであり、計算機間におけるプロセス間通信を行う事も可能である。
- ③RPCは手続きを他のプロセスの環境で実行させるもので、計算機間における実行も可能である。この時RPCでは呼び出した結果を返事として受け取る事が想定されている。

何れにしても、データ量、通信の頻度、プロセスの生存期間、同時生存数、チャンネル生成の作業量、同期の必要性などを検討する必要がある。

又、マルチメディア情報を扱う事に関しては動画や音ファイルをテキストファイルと同様に扱う事、データ圧縮や伸長機能等が必要である。

即ち、マルチメディア情報を処理するシステムでは先に述べたプロセス間通信をデータ・ファイルのフレキシブルな変換機能を基とした通信方式を具体化しなければならないのである。そして、本データ定義プログラム間通信方式ではこれらの多くの基本問題が解決される。以下に、マルチメディア情報を想定した場合での本通信方式の適応を述べる。

① 通信手法の非依存性

既に述べた様に、本方式では同一機種ないでのプログラム通信では共有メモリ、メッセージパッシング及びリモートプロシージャコールの何れの手法に関しても併用によってそのデータ通信過程の簡略化を果たし、又既存のネットワークプロトコル内(例えばインターネットにおけるTCP/IP上でのクライアント・サーバモデルでは)であればPRCと同等の機能を実現する事が可能である。即ち、クライアント・サーバモデルにおいてのデータ通信上の問題点であるデータ互換性保持機能を付加する事が出来る。

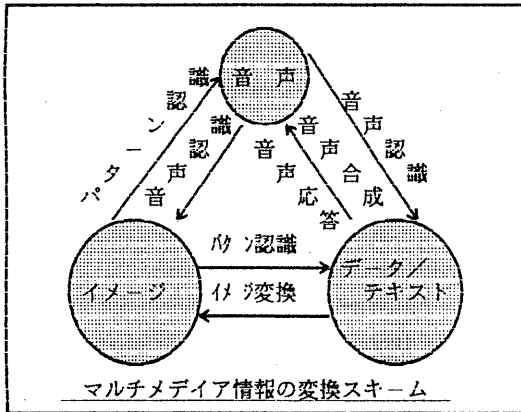
② マルチメディア情報の一元処理

本方式はコンピュータ内のプログラム間又はネットワークされているコンピュータ間のデータ通信のデータ互換性を保持する為の機構であるから、テキストファイルなどを意識する事なく異機種でのデータ処理を補償する事によってマルチメディア情報の通信を簡易化する。さらには、情報量の大きさの為に必要とされる「データ圧縮や伸長機能」に関し

てもデータ定義体及びデータ構造変換テーブルによって容易に本方式の機能のみで対処可能である。特に、マルチメディア情報の対象となる画像（動画を含む）や音声等の情報は通常のテキストファイルに出来る情報とは異なって各自にデータ定義が必要であり又情報量が多い。さらに、これらの通信では同時に伝送されて、アプリケーション処理に必要なレベルでの同期性が確保されなければならない。即ち、マルチメディア情報のリアルタイム通信処理などに於いては運用するアプリケーションによっても異なるが通信プロトコルに依存する機能を除いて以下の要件を満たす必要性がある。

☆マルチメディア情報のデータ通信要件☆

要件1) 音声・画像データをデータ定義し、大容量データを通信する為の「圧縮及び伸長処理」音声データ及び画像データはコンティニューアなデータ（連続する事によってデータとしての意味を持つ）であり、一般的に通常のテキストファイルに比べて大容量である。よって、データ転送速度を高速化する為にはデータ圧縮及び伝送後のデータ伸長が必要である。



上記の様なマルチメディア情報の変換考えられ、基本的には全てをデータ/テキスト情報への変換が通信に取っては効果的である。しかし、現在の技術ではこれらを実施する為にはこれら二つの情報構造の相互変換機能が各データ受渡しに必要となり処理の煩雑化を免れない。よって、データの圧縮/伸長による伝送が現実的であり、これらを情報定義して通信する方式が必要である。

要件2) 同一機のソフトウェア内におけるプログラム間通信データの処理データへの変換
同一機種内におけるプログラム間通信に於いてその通信レスポンスを確保する為には、元データをその物を通信で伝送する（これは先に述べた共有メモリ法で対処されるのが一般的である）より関数又はパラメータなどの形で受渡す事の方が効果的である。よって、そのデータ受渡し後に元データへの変換が

必要である。

要件3) 異機種間通信におけるアプリケーションでの処理データの互換性維持

異機種間のプログラムデータ通信に関しては要件2で述べた同一機種内に比べてネットワーク上でのデータ転送過程が加わる為、さらにレスポンス確保は重要であり、又機種間に存在するデータのコード体系の差異にも対処しなければならない。

☆要件の解決☆

以上述べたマルチメディア情報通信における解決すべき要件の本手法による解決策を次に述べる。

① 圧縮・伸長データの定義とその処理

大容量の伝送の為に圧縮したデータはテキストでは単純に符号化する事で、又、画像データではJPEG規格等に見られる様なデジタル画像符号化によってなされています。元データを符号化する事が情報圧縮の基本であり、この圧縮及び伝送後の伸長は各々のネットワークされているコンピュータで実行されるので、この手続きそのものを本方式に取り込む事が可能である。即ち、本方式における「データ変換情報テーブル及びその変換処理プログラムルーチン」にこれらデータ種（テキスト、動画、音声）に対応する圧縮及び伸長の機能を定義し、機能プログラムルーチンを用意する事によって処理可能である。

② 同一機種内におけるプログラム間通信処理

同一機種内でのプログラム間通信においては異機種間で生じる様なデータコードやデータ長の違いによる互換性維持の為の処理は不必要であるが、マルチメディア情報の様にプロセス間通信による使用メモリの肥大化が予想される場合には、その抑制する為の処理に本方式は適応出来る。プロセス間通信にはメモリの有効活用及び処理の簡易化の為に関数やパラメータを使用する事が多いが、データ通信においてはプログラムで処理されるべきデータコードに変換される必要があるのは言うまでもない。そこで、データとして引き渡される関数又はパラメータにデータ定義をする事が可能である。即ち、処理データ範囲が規定されている場合、データ変換定義情報で圧縮したデータをプログラム間通信に用いて、各プログラム内でのデータ処理は共通メモリに保持されているデータ変換情報テーブルを参照する事によって実施する事になる。この事によって、使用される共有メモリを制限できるばかりか、処理速度そのものにも関わらず大容量の元データをそのものを直接通信する場合に比べて高速化する事が可能である。

③ ネットワークでのプログラム間通信処理

ネットワークを通じてつながっているコンピュータのプログラム間通信を行う場合には先にも述べた様にプログラムが作動するコンピュータのデータ処理構造が異なる事が予想される。この場合に、画一的に管理された情報である事がデータ伝送処理を行う時のブレークスルー技術となる。

ネットワークされているコンピュータ上のアプリケーション・プログラムから見た場合に一元管理されている事、即ち、通信する異なるコンピュータで作動しているプログラムが処理するデータとしてはデータを送った側のハード/ソフト環境におけるデータ構造と受け取った側のハード/ソフト環境におけるデータ構造が互換性を持つ事が可能である。この場合にはネットワーク上の各コンピュータにセットされるデータ変換情報テーブルにそのセットされているコンピュータが扱えるデータ構造を定義する事だけが必要である。実際のネットワーク通信において、通信されるデータの内で大容量である場合には前述の様なデータ圧縮及び伸長処理が付加されるが、通常範囲のデータ量〔元データ量が通信のボトルネックにならない程度のもの〕の場合では元データレベルでのデータ変換情報テーブルによる一元管理が出来る。

おわりに)

本方式は通信するデータ構造を定義する事によるネットワークにおける異機種間通信、複数のデータ種(音声、動画像、テキストなど)の一元管理を比較的簡易な手続きによって実現する事を目的に研究した結果として得られたものであるが、今後、OSへの組み込み及び、異機種間のプログラム関数変換への適応に関して検討を進めマルチメディア情報通信技術として開発を進める予定である。

謝辞)

本研究を行うに当たり、Intelligentics IncのBionet Manager & Marketing Director であるDr. Murray Summers、現群馬大学工学部の宮澤二造氏に有意義なコメントを頂いた、ここに謝辞の意を表します。

参考文献)

- ① Considerations for ISDN Planning and Implementation, by G. S. Bhusri., IEEE Communication Magazine, Vol22 No1, (1984), pp18-32.
- ② Economic Analysis of Integrated Voice and Data Network : A case Study, by I. Gitman & H. Frank., Proc. of the IEEE, Vol66, No11(1978)
- ③ ISO : Addendum to ISO 7498 Covering Connectionless Mode Transmission.
- ④ ISO /TCP7/SC21N1235 "Directory Access and System Protocol"
- ⑤ 「マルチメディアデータベース」 Compute Today, 1986/9.
- ⑥ ISO : Open Systems Interconnection (1984).
- ⑦ JIS : 開放型システム相互接続の基本参照モデル(1985)
- ⑧ A Real Time Electric Conferencing System Based on Distributed UNIX, Proc. Atlanta Usenix Conf(1986), by T. Suzuki et al.
- ⑨ Distributed Interoffice Mail system based on Integrated Document Interchange Protocol by S. Sakata et al., Proc. COMPCON FALL., (1984).