グループウェア 8－6
（1994．10．14）

グループウェア向け通信用関数の開発と評価

山元 一永　　宗森 純　　長澤 庸二

鹿児島大学

グループウェア向け通信用関数の開発をおこなった。この通信用関数はグループ管理の機能の他に、静止画の転送機能も持つ。本稿では作成したグループウェア向け通信用関数の概要について述べた後、従来の通信用関数との比較をおこなう。

Development and Estimation of Communication Functions for Groupware

Kazunaga YAMAMOTO　　Jun MUNEMORI　　Yoji NAGASAWA

Kagoshima University

Communication functions for groupware has been developed and estimated. These functions can manage the state of member of same group and transmit image data. We describe these functions and compare them with other commucation functions in this paper.

1．はじめに

近年，グループのための協調作業を支援する環境を提供するグループウェアの研究が進んでいる[1]．グループウェアを実現するためのソフトウェアは，計算機同士でお互いにデータを交換しながら作業を進めていく．例えば，共有データに変更が加えられた場合は即座に他の計算機の同じ部分も変更される．そのために通信部分のプログラムが不可欠である．またグループウェアでは，多数の参加者によって作業が進められるため，参加しているメンバーの管理なども必要となり，通常のデータ通信に比べ様々な機能が要求される．

既存の通信用関数を考えてみると，たいていの計算機のシステムには通信用の関数が用意されている．しかし，一対一の計算機での通信にしか利用できなかったり，一対多の通信に利用できたとしてもグループ管理機能がないので，ソフトウェア側でそれらの通信用関数を使ったプログラムを組み，要求を満たす形にする必要があった．

そこで，グループウェアを実現するソフトウェアから簡単に利用できる通信用関数として，HyperCardの外部関数であるHyperPPCを開発した．

本報告では，開発したHyperPPCの概要を説明し，その評価を行う．

2．HyperCardの通信用関数

2．1　HyperCard

HyperCardとは，AppleComputer純正のアプリケーションであり，プログラミング環境としても利用できる．HyperCardの例を図1に示す．プログラムの記述にはHyperTalkという言語を用いる．HyperCardにより作成されたソフトウェアはスタックと呼ばれ，ソフトウェアであると同時にHyperCard本体に対するデータファイル的な位置にある．

スタックは，ボタンやフィールドを含んだカード，そのカードを含んだバックグラウンド

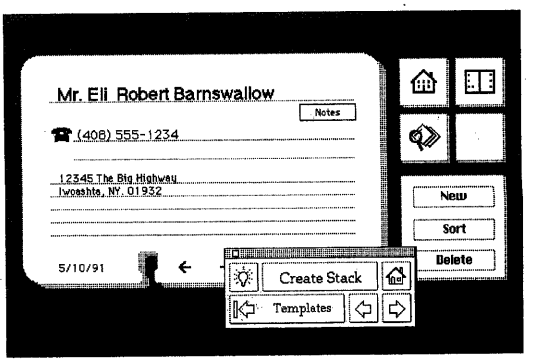

図1　HyperCardの例

などで構成され，それらのオブジェクトは視覚的に作成することができる．プログラムはオブジェクト(ボタン，フィールド，カード，バックグラウンド，スタック)の内部にスクリプトとして記述され，スクリプトは，各動作に対応したメッセージの発生により実行される．また，メッセージは，それを処理するスクリプトが書かれたオブジェクトがない場合は，そのオブジェクトを包括するオブジェクトへと継承されていく．

HyperTalkは実際の英語に非常に近い記述形式となっており，またインタプリタであることなどから初心者でもプログラムが組みやすく，開発効率が高い．さらに，HyperTalkにない命令などは，他の言語でプログラムを書き，それを外部関数という形にすれば既存の命令と同様に実行できるため，容易に言語の拡張をすることもできる．

2．2　HyperCardの従来の通信用言語

（1）AppleEvent

HyperCardに備わっている言語の一つであり，通信手段として利用できる[2]．本来は，データの送受信よりも，アプリケーション間のイベントのやり取りを目的としているが，ネットワークにより接続された他の計算機上のアプリケーションともイベントのやり取りができるため，通信手段として利用できる．送られたイベントは通常はメッセージとしてそのまま実行されるが，そのメッセージをスクリプトで受け

止めて，命令として扱わずにデータとしても利用できる．使用法は簡単で，命令実行前に設定などの準備の必要はない．

このように簡単に利用出来る通信手段ではあるが，欠点としてはグループ管理の方法がないということがあげられる．例えば，接続時にネットワーク上の計算機名を自動的に得る方法がなく，また自分と同じグループなのかどうかを判断することができない．グループ内の計算機がグループから抜け出た場合に確かめるまでわからないことや，グループ内の全ての計算機に送信したい場合でも一度に一台の計算機にしか送信できないため一台ずつ送信する必要があることなどがあげられる．また，送信するデータの内容もテキストデータしか引数として指定できない．

（2）HyperAppleTalk

AppleComputerにより通信用に作成されたHyperCardの外部関数群である[3]．もともとHyperCardにそなわっているAppleEventに比べると，HyperAppleTalkは拡張された関数であるため，使用するのにあらかじめ設定をおこなう関数を実行する必要があるなど，若干手間がかかる．HyperAppleTalkの関数を表1に示す．

利用の手順は，ATPOpen，NBPOpenで計算機をネットワークと接続し，NBPRegisterNameを用いて自分の名前，グループ名を登録する．つぎにNBPLookupNamesを使用して，グループのメンバーリストを得る．そして，得たメンバーリストにより送信先を指定してATPSendRequestでデータの送信をおこなう．データの受信はATPReceiveによりおこなわれる．必要に応じてNBPConfirmNameにより，メンバーときちんと接続がなされているか確かめることができる．

HyperAppleTalkはグループウェアでの使用をよく考えられて作成されており，計算機がどのグループに属しているかの判別，送信先リストを指定しての一度に多数のメンバーへの送信などができるのが特徴である．欠点としては，一度に送信できるデータ量が600バイト程度に限られていることがあげられ，またAppleEvent同様テキストデータのみしか送信できない．

表1　HyperAppleTalkの関数

ATPOpen	計算機をネットワークに接続する
NBPOpen	計算機をネットワークに登録する
NBPRegisterName	ユーザー名、グループ名の登録
NBPLookupNames	グループのメンバーリストを得る
ATPSendRequest	データの送信
ATPReceive	データの受信
NBPConfirmName	接続の確認
ATPClose	計算機のネットワークに対する接続の解除
NBPClose	ネットワークの計算機名の登録の解除

3．HyperPPCの開発

3．1　開発環境

開発環境は，ハードウェアとしてMacintosh Quadra700 (Apple Computer)，16インチカラーCRTを使用し，ソフトウェアとしてはTHINK C (Symantec Corporation)を使用して，前述したHyperCardの外部関数の形で作成した．

実際の通信部分には，MacintoshのOS (System 7)に用意されている関数群Toolboxの中の一つである，PPCToolbox[4]を使用している．また，HyperPPCの名前の由来は，HyperCard，PPCToolboxの二つからきている．PPCToolboxとは，Program-to-Program Communicationsの略で，主にプログラム間の通信に必要な基本的な関数群が用意されている．それらの関数にグループウェアの機能を付加した形でHyperPPCを実現した．

HyperPPCは，C言語で2000行弱のプログラムで，コードサイズにして，約51Kバイトである．

3．2　現在の機能

従来からあったAppleEventとHyperAppleTalkの欠点を補い，各々の長所をいかしたものがHyperPPCである．現在のHyperPPCは表2に示したように全部で九つの外部関数からなる．これらの関数をスタックに

外部関数として加えることで，利用できるようになる．

HyperPPCの主な機能としては，同じグループの計算機との自動接続，接続許可不許可の設定，新たにグループに参加してきたメンバーや，グループから抜け出たメンバーがいた場合のメッセージの発生，さらにAppleEventやHyperAppleTalkは，送信データとしてテキストデータの指定しかできなかったが，HyperPPCは送信データに，テキストデータだけでなく，イメージデータも引数で指定できるようにして，イメージデータの転送も可能としたことなどがあげられる．

HyperPPCの処理の流れは，図2のようになっている．まず，PPCOpenを使ってユーザー名とグループ名を登録する．次にPPCLinkを用いて接続する．これは自動的に同じグループの計算機と接続する．他の計算機と接続されメンバーに変更があった場合は，それを伝えるメッセージが発生するので，それに応じてPPCLookMemberを使用してメンバーリストを得て，グループ管理に利用する．データの送信は，送信先のメンバーを指定してPPCSendで送信し，データの受信はPPCReceiveによっておこなう．すべてのデータの送受信が終わり，終了するときはPPCCloseを実行することにより，ネットワークとの接続がきれる．また，必要に応じて，PPCRemoveMemberにより指定したメンバーとの接続を切ったり，PPCSetInformにより接続要求を受け付けないようにしたりできる．

UNIXのソケット通信[5]との対応を図3に，実際のプログラム例を図4に示す．

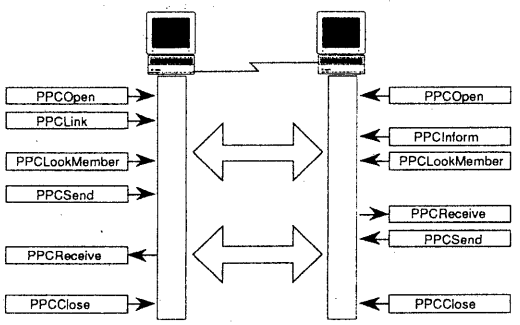

図2　HyperPPCの関数利用の流れ

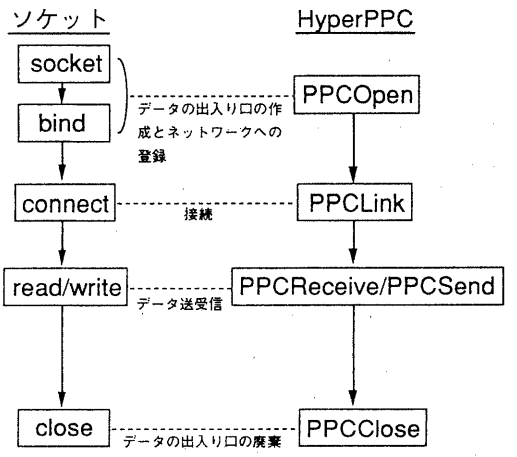

図3　UNIXのソケット通信との対応

表2　HyperPPCの関数

PPCOpen	ネットワークにグループ、名前の登録
PPCLink	同じグループの計算機と接続
PPCInform	他の計算機からの接続要求の受け付け
PPCSend	データの送信
PPCReceive	データの受信
PPCClose	通信の終了
PPCRemoveMember	指定されたメンバーとの接続を切る
PPCLookMember	現在のグループ内のメンバーのリストを得る
PPCSetInform	他の計算機からの接続を許可するかどうかの設定

```
on openStack
  PPCOpen name, group, 4   --登録
  PPCLink   --接続
end openStack

on idle
  PPCReceive "mess", "memberChange"   --受信
  --これによりデータ受信時 "mess"、
  --メンバー変更時 "memberChange"の
  --メッセージが発生する
end idle

on memberChange
  get PPCLookMember()   --現在のメンバーを得る
end memberChange

on closeStack
  PPCClose   --終了処理
end closeStack

on send data, list, type, creator
  PPCSend data, list, type, creator   --送信
end send
```

図4　HyperPPCを用いたプログラム例

4．評価

HyperPPCと従来のHyperCardの通信用関数との比較は表3のようになる．従来のHyperCardの通信用関数と比べて優れている点が多いが，まだ改良の余地はある．

表3　AppleEvent, HyperAppleTalkとの比較

	HyperPPC	HyperAppleTalk	AppleEvent
同時転送	○	○	×
画像転送	○	×	×
グループ管理	○	○	×
手軽さ	△	△	○
大容量転送	○	×	○

次にデータ転送速度の比較結果を図5に示す．1～10000バイトのテキストデータを1台の計算機(Macintosh Quadra700)から別の1～4台の計算機(MacintoshIIfx)に送信し，すべての計算機が受信を完了するまでにかかった時間を示してある．ネットワークはEthernet (10MBPS)である．なお，HyperAppleTalkは一度に転送

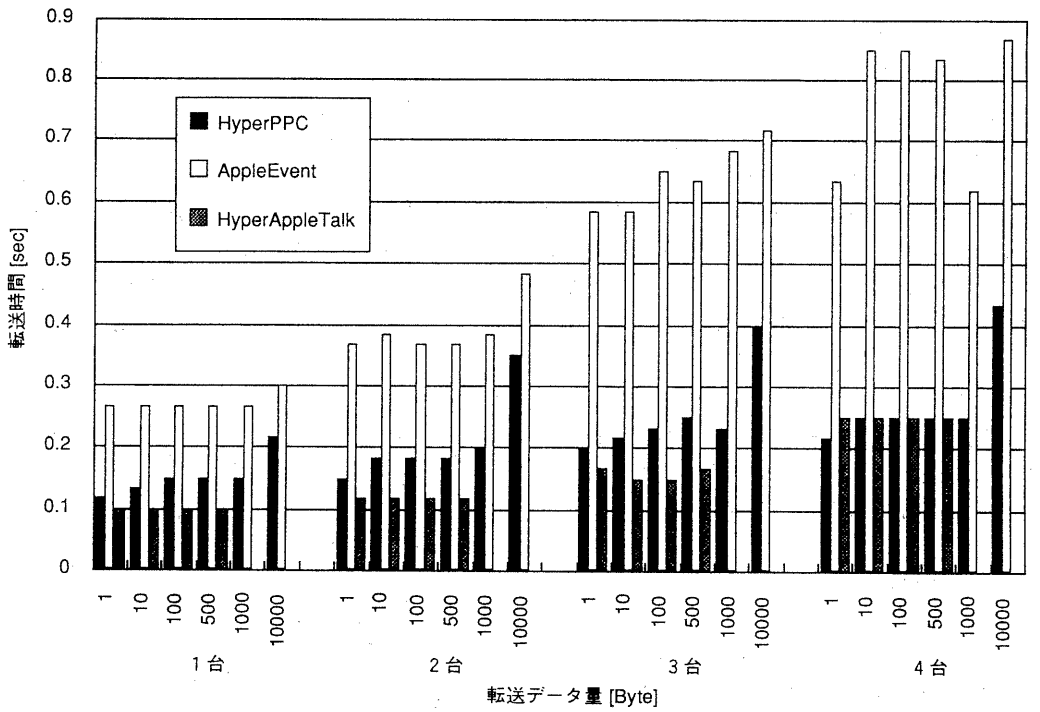

図5　HyperPPCと従来の通信手段との転送時間の比較

できるデータ量が600バイト程度までに限られているので，1000バイト，10000バイトの転送実験はおこなっていない．

HyperAppleTalkと比較してみると，転送速度はHyperAppleTalkのほうがHyperPPCより速い．この原因は，HyperAppleTalkが一度に転送できるデータ量を600バイト程度に制限していることからきている．データ量の最大値を決めておくことで，あらかじめ転送データ用のバッファを確保しておくことができる．一方HyperPPCの場合は，画像などの転送にも利用できるように考えていることから，データ量の最大値を決めることが困難であり，そのようにしても無駄が多い．そのため，まず送るデータのサイズを受け手に知らせてその大きさのバッファを確保してもらい，次にデータを送るという，動的にバッファの確保をおこなう手段をとっている．そのためHyperAppleTalkより時間がかかっている．実際にHyperPPCもデータ量の最大値を決め，あらかじめバッファを用意しておくようにして，実験してみると，HyperAppleTalkと同様のスピードが得られた．

AppleEventを見ると，HyperPPC，HyperAppleTalkに比べてかなり転送速度が遅いが，これはAppleEventがイベントの送受信を目的に作られているため，純粋なデータ転送以外にもいろいろな処理をおこなっているためと考えられる．またAppleEventは，ひとつのアプリケーションに対してデータの送信をおこなうようになっているため，メンバー全員にデータを送信する場合，ループの中で一台ずつ送信しなければならない．そのための時間が余分にかかることも原因のひとつであろう．

5．おわりに

グループ管理とテキストデータの送信に加えイメージデータの送信も可能にした，グループウェア向け通信用関数HyperPPCを開発した．

今後の目標として，次の3点があげられる．

(1) 動画像データの転送
(2) グループウェアに必要な機能の追加
(3) HyperCard以外のアプリケーションからの利用

まず，(1)であるが，グループウェアで扱うデータもテキストから静止画，動画と移り変わってきているので，それに対応させていく．現在は，試験的に静止画を何回も送ることで動画のように見せることはできる．今後はQuickTime[6]などの転送や，MovieTalk[7]の取り込みなども検討していく．

次に(2)では，例えば通信のログを記録したり，またそれを再生することによって，作業の流れを追っていったりする機能などがあげられる．また，グループウェアには共有データを操作する権利である操作権も必要なので，それの管理などもおこなっていきたい．

最後に(3)では，現在の外部関数の形から一つの独立したアプリケーションの形にして，それをAppleEventを用いてコントロールすることを考えている．具体的には，アプリケーションにしたHyperPPCをバックグラウンドで動かし，C言語などで開発したプログラムからAppleEventを用いて，データ送信などをHyperPPCに依頼する形となる．また，逆にHyperPPCの方からデータの要求をして，受け取ったデータを他の計算機に送信することにより，アプリケーションを強引にグループウェアにしてしまうことなども考えている．

参考文献

[1] 松下　温：図解グループウェア入門，オーム社(1991).
[2] Apple Computer, Inc：Inside Macintosh Volume VI (日本語版), chapter 6, アジソンウエスレイ (1993).
[3] Macintosh DEVELOPER'S JOURNAL, No.3, apda[日本語版], p.117, 技術評論社 (1993).
[4] Apple Computer, Inc：Inside Macintosh Volume VI (日本語版), chapter 7, アジソンウエスレイ (1993).
[5] 村井　純，井上　尚司，砂原　秀樹：プロフェショナルＵＮＩＸ，アスキー出版局 (1990).
[6] Apple Computer, Inc：INSIDE MACINTOSH QuickTIme (日本語版), アジソンウエスレイ (1993).
[7] MACLIFE, No.73, p.157, BNN (1994).