

エラー検出機能をもつグループウェア向け 通信用関数HyperPPCの開発

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。さらに発想一貫支援グループウェア郡元に適用し、エラー検出機能の付加も行なった。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と、付加したエラー検出機能について述べる。

HyperPPC : Communication Functions for Groupware that can detect Error

Kazunaga YAMAMOTO

Jun MUNEMORI

Yoji NAGASAWA

Kagoshima University

Communication functions for groupware has been developed. These functions can manage the state of member of same group and transmit image data. These functions has been used in a groupware for new idea generation support system, GUNGEN, and we also add ability of error detection to these functions. We describe summary of these functions and additional ability of error detection.

1. はじめに

近年、グループによる協調作業の環境を支援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ため、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。

パーソナルコンピュータにおける既存の通信関数を考えてみると、通信用の関数が用意されているが、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ものが多く、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、発想一貫支援グループウェア郡元[2]に適用した。そして、郡元での使用経験からエラー検出機能を付加した。

本報告では、開発したHyperPPCの概要と、発想一貫支援グループウェア郡元に適用した結果、および付加したエラー検出機能について報告を行う。

2. 発想一貫支援グループウェア郡元への適用とその問題点

作成したHyperPPCを発想一貫支援グループウェア郡元に適用し、KJ法の実験で実際に使用した。KJ法の実験は、テキストベースの雑談の実験[3]も兼ねていることもあり、参加者それぞれが異なった部屋や階で実験をおこなうこともある。そして、実験中に一台の計算機がエラーを起こして、データの送受信ができなくなった場合(図1)は、実験を中断することになる。その場合、計算機がエラーを起こしたことを他の正常な計算機を使用している参加者に伝えて、ただちに処置をとる必要がある。そのようにしないと、実験のデータが正しくとれないばかりか、エラーを起こした計算機の影響で他の正常な計算機までもエラーを起こしてしまうことがあった。そのため実験中にエラーが起こった場合、担当者が各階を走り回ることになる。そのようなことから、エラー検出機能がHyperPPCに求められた。HyperPPCで他の計

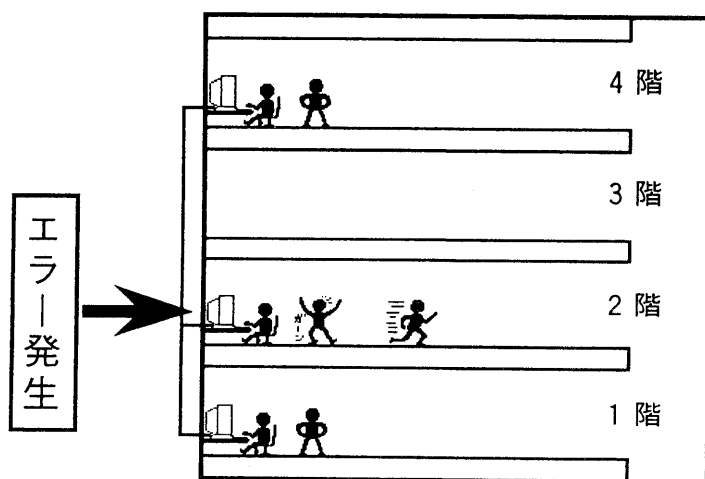


図1 KJ法実験でのエラー発生

算機がエラーを起こしたことが検出できれば、プログラムの処置をすることができる。さらにはエラーの状態からの自動復帰などにも応用できるためである。それらを考慮してエラー検出機能を通信用関数HyperPPCに付加した。

3. エラー検出機能を持つ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Symantec C++ (Symantec Corporation)を使用して、前述したHyperCard (Apple Computer)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 7.1)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一对一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言語で約2000行のプログラムで、コードサイズにして、約51Kバイトである。

3. 2 現在の機能

従来からあった通信用関数AppleEvent[5]と

HyperAppleTalk[6]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1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HyperCardの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

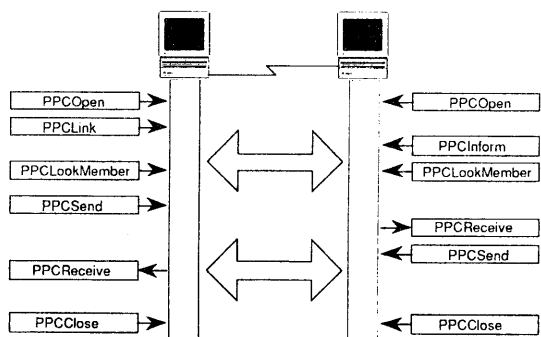


図2 HyperPPCの処理のながれ

表1 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

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、終了するときには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。

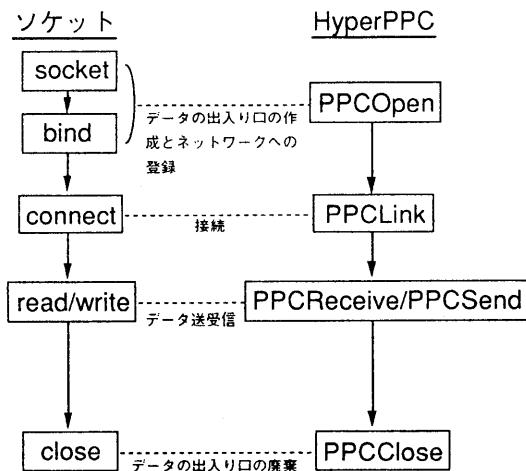


図3 UNIXのソケット通信との対応

3. 3 エラー検出機能の付加

エラー検出の方法として、まず、グループ内の計算機では常に一定の時間間隔で信号のやり取りを行なうようにした。信号のやり取りの様子を図5に示す。そして当初、他の計算機からの信号が1つでも、一定時間より大幅に遅れた場合、エラーが起これたと考えるようにした。

```

on openStack
  --最初に一度だけ行なう処理
  PPCOpen name, group, 4 --登録
  PPCLink --接続
end openStack

on idle
  --アイドル処理
  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
  --メンバー変更のときのメッセージ
  get PPCLookMember() --現在のメンバーを得る
end memberChange

on closeStack
  --終了時に一度だけ実行する処理
  PPCClose --終了処理
end closeStack

on send data, list, type, creator
  --データを送信するとき
  PPCSend data, list, type, creator --送信
end send

```

図4 HyperPPCを用いたプログラム例

しかし、それでは自分の計算機がエラーを起こして信号を受信できないのか、それとも他の計算機がエラーを起こして信号を送ってこれないのか判断がつかない。そこで、郡元での実験は主に4人で行なわれることから、グループ内の他の計算機すべてから信号が送ってこない場合は、自分の計算機がエラーを起こして信号を受信できないためと判断し、グループ内の1、2台の計算機から信号が送ってこない場合は、それらの計算機がエラーを起こして信号を送信できないと判断することにした。そして信号の遅延があった場合は、メッセージが発生することでプログラムに伝えるようにした。そのメッセージを利用すれば、プログラムの通信のエラーに対する処理ができるようになる。実際のプログラムは、メッセージを受けとめて、次のような処理を行なうことが考えられる。

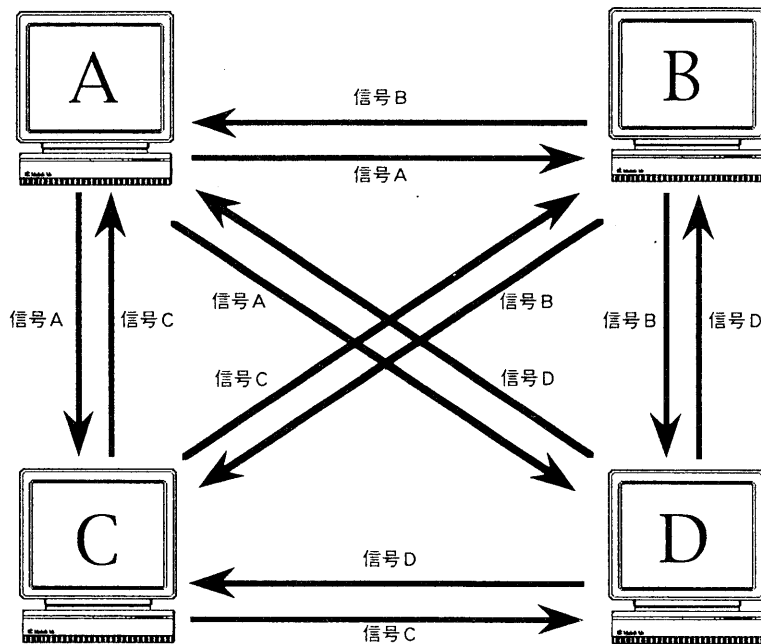


図5 一定時間間隔で信号のやり取りを行なう

- (1) 信号が遅れている可能性を考えて、もうしばらく待つ。
 - (2) HyperPPCを初期化し直す。
 - (3) 計算機をリスタートする。
- 実際のプログラム例を図6に示す。

4. 通信用関数の今後の展開

通信用関数の今後の展開として、次の4点を考えている。

- (1) 動画データデータの転送
- (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
- (4) エラー検出機能の充実

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

on SignalLate name

```
--信号受信が遅れているときに発生するメッセージ
--nameに信号が遅れている計算機名が入る
if name is not myMachine then
  --他の計算機がエラーの時
  answer "他の計算機にエラーが発生しています。"
    &return&"しばらくお待ちください。"
else
  --自分の計算機がエラーの時
  answer name&"からの信号が遅れています。"&return
    &"どうしますか?" with "そのまま待つ"
    or "初期化する"
    or "リスタートする"
  if it is "初期化する" then
    --初期化の処理
    PPCInitialize
  else if it is "リスタートする" then
    --リスタートの処理
    Restart
  end if
end if
end SignalLate
```

図6 エラー検出メッセージを用いたプログラム例

QuickTime Conferencing[6]などの利用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、その管理などもおこなっていききたい。

(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。

最後に(4)では、今回付加したエラー検出機能について、実際に使用してさまざまな機能の充実をはかっていくことを考えている。現在は、自分の計算機がエラーを起こしているのか他の計算機がエラーを起こしているのかの判断を、グループ内の他の全ての計算機からの信号を受信できるかできないかで判断しているが、この方法は手間がかかる上に確実ではない。また接続する計算機の台数が多くなってくると、信号の遅延などでグループ内の計算機の信号すべてが届くのに時間がかかるようになることも考えられる。そこで今後は、信号を自分の計算機自身に送って受信し、自分の計算機がエラーを起こしているかどうかの判断に使用することを考えていく。図7にイメージ図を示す。しかし現段階では自分の計算機自身と接続するとき二重に接続してしまうなどの問題点が残っている。

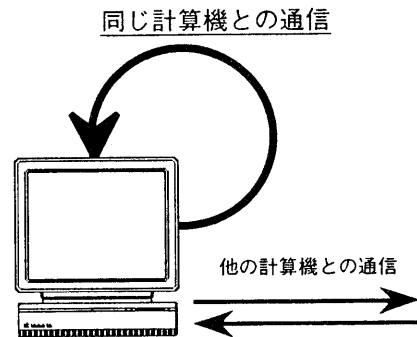


図7 同じ計算機に信号を送る

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、エラー検出機能も付加した。今後は4.で述べた今後の展開を検討していく。

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] 由井 隆也、宗森 純、長澤 庸二：発想一貫支援グループウェア郡元の開発と適用、情報処理学会、グループウェア研究会、8-7(1994).
- [3] 五郎丸 秀樹、宗森 純、長澤 庸二：グループウェアにおけるテキストベースのコミュニケーションについて、情報処理学会、システムソフトウェアとオペレーティング・システム、マルチメディア通信と分散処理合同研究会、17(1994).
- [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] Apple Computer, Inc：Inside Macintosh Volume VI (日本語版), chapter 6, アジソンウエスレイ (1993).
- [6] Macintosh DEVELOPER'S JOURNAL, No.3, apda[日本語版], p.117, 技術評論社 (1993).
- [5] 村井 純、井上 尚司、砂原 秀樹：プロフェッショナルUNIX、アスキー出版局 (1990).
- [6] MACLIFE, No.80, pp.216-219, BNN (1995).