

# 発言と行動の管理に基づいた協同作業支援電子メール PilotMail†

市村 哲†† 松下 温††

今や電子メールはコミュニケーション媒体としてオフィスなどで広く用いられるようになってきた。ところが既存の電子メールは、一方向にメッセージを転送するための手段を提供しているにすぎず、人間の対話や協調活動そのものを支援する機能を備えていない。この問題に対し、人間の意図を正確に伝えることができるような電子メールシステムを構築するという方向からアプローチを行った。われわれは、メッセージを受け取った受信者がどのように振る舞えばよいのかを送信者がナビゲートできる仕組みを電子メール上を実現し、PilotMail システムと名付けた。PilotMail を使用することにより、送信者は誤りなく意志を受信者に伝えることができ、また、電子メール上での人と人の情報交換作業が迅速かつ柔軟になる。システムは、オブジェクトを転送する機能と、クラス階層化管理機能をサポートしており、Objectworks/Smalltalk 上で実装された。さらに、会話の遷移モデルに基づく会話管理機能とプロシジャを転送できる機能を融合し、より人間の協調構造に密着した対話支援を可能にした。PilotMail はグループの協同作業を支援することが目的であり、これまでにわれわれが行った利用実験から、協同作業促進のために有効であると評価された。本論文では、設計思想、特徴的機能、システムの適用・評価について論じる。

## 1. はじめに

近年協同作業を支援することの必要性が高まるにつれグループウェアに関する興味が強まっている<sup>1)~3)</sup>。Winograd はグループウェアの特徴として、実現に用いるコンピュータシステムの構造ではなく、人間の協調活動の構造に設計の焦点があることを挙げている<sup>4),6)</sup>。すなわち、人間がコンピュータをどのように利用していくべきかという問題のほか、人間と人間をどのようにすれば密に融合できるかという問題が存在しているのである<sup>6)</sup>。この問題に対し、われわれは、人間の意図を他の人に正確に伝えることができるようなコンピュータシステムを構築するという方向からアプローチした。

電子メールは、紙を用いず、しかもワープロソフトを使って作成した原稿や書類あるいはコンピュータプログラム等を瞬時に相手に送るといった使い方ができることから、広く普及してきた。ところが既存の電子メールは、一方向にメッセージ(データ)を転送するための手段を提供しているにすぎず、人間の対話や協調活動そのものを支援する機能を備えていない<sup>4),5),7),8)</sup>。加えて、非同期分散型のコミュニケーションの場合、宿命的に、送信者はメッセージの受取人の行動を予測することができない。特にビジネスでは、「発言と行

動は同一のものである」という考え方が根底になくはならず、契約と行動の両方を管理することが重要である。

われわれは、以上のような視点から、メッセージを受け取った受信者がどのように振る舞えばよいのかを送信者がナビゲートできる仕組みを電子メール上を実現し、PilotMail システムと名付けた<sup>9),10)</sup>。これにより、発言によって意図を伝達するという従来のコミュニケーション形態を超えて、受信者の振る舞いをガイドすることが会話を構成する重要な要素の一つになる。すなわち、送信者はより正確に意志を受信者に伝えるための手段を得ることができ、また、グループ内の情報交換作業が迅速かつ柔軟になる。システムは、オブジェクト(データ+手続き)を転送する機能と、半構造化メッセージ機能をサポートしており、Objectworks/Smalltalk<sup>11)</sup> 上で実装された<sup>12),13)</sup>。さらに、会話の遷移モデルに基づく会話管理機能とプロシジャを転送できる機能を融合し、発言と行動がより高度に調和した対話環境を実現した<sup>14)</sup>。PilotMail はグループの協同作業を支援することが目的であり、これまでにわれわれが行った利用実験から、協同作業促進のために有効であると評価された。本論文では、主に、設計思想、特徴的機能、システムの適用・評価について論じる。

## 2. 関連研究との対比

グループウェアに対する研究は、電子メール・電子掲示板の普及をうけて発展してきたと言っても過言で

† PilotMail: A Group Oriented E-mail Based on Management of Speaking and Acting by SATOSHI ICHIMURA and YUTAKA MATSUSHITA (Department of Instrumentation Engineering, Faculty of Science and Technology, Keio University).

†† 慶應義塾大学理工学部計測工学科

はない。CSCW (Computer Supported Cooperative Work) 研究の視点からは、電子メールの中に人間の協調構造を取り込もうとすることに研究者の関心が集まっている<sup>15)</sup>。

Winograd と Flores は会話の遷移モデル (図1参照)<sup>4)</sup> を提唱し、このモデルに基づいて設計された Coordinator という製品を開発した<sup>5)</sup>。人間のう会話話を単なるメッセージの集積ではなく、構造を持った情報処理機構として捉える必要があり、その中では、会話の行方 (将来とり得る遷移状態や結果) が意識されなければならないと唱えている。しかしながら、Coordinator は受信者の行動の結果をチェックする機能をサポートしているものの、受信者の行動そのものを管理するための機能を持っていない。発言と行動の両方を管理することを目的としているにもかかわらず、思い違いや誤解のために発言と行動が同一の意味を持たなくなる可能性を考慮していない<sup>14)</sup>。ひとたび誤解が生じるとそれを解消するためのコストは膨大なものになり、チームにとって多大な損失がもたらされる<sup>9),10)</sup>。そこでわれわれは、受信者が送信者からの要求された行動を実行するのは図1中の状態3のフェーズであるから、誤解の発生を予防するためにはこの状態3における受信者の行動そのものを管理する必要があると考えた<sup>14)</sup>。われわれはこの点に着目して会話の遷移モデルに基づく会話管理機能と PilotMail の持つプロシジャ転送機能を融合し、行動の管理をも含めた対話の流れの管理を可能とした。

Malone らは情報交換の場における「情報の氾濫」という問題を解消すべく Lens 電子メールシステムを開発した<sup>16)</sup>。Lens システムでは、半構造化メッセージタイプを導入し、あらかじめルールベースに自動処理手続きを登録しておくことにより自動的に情報を分類することができるようになってきている<sup>17)</sup>。また、オブジェクト指向データモデルの採用により、メールの使用目的に応じたクラス階層化を行ってメールを管理している<sup>18)</sup>。以上のように Malone らは電子メールの抱える数々の問題点を指摘し、それに対する興味深い解決策を提案してきた。しかしながら、対話の流れを支援することを目的としておらず、われわれや Winograd が問題視している会話のサポートの必要性に関して論じていない。また PilotMail では、各フィールドのメッセージ自体がデータタイプを持ったオブジェクトであり、受信者が実行するためのプロシジャを送信者が任

意のフィールドに記述し、オブジェクトの振る舞いをメッセージ中に含めることができるという機能を提供している<sup>13)</sup>。

われわれは、メールのクラス階層化管理機能と、半構造化メッセージの中に自動処理手続きを記述できる機能と、会話管理機能を融合することを目指し、これをオブジェクトを転送するという統一された手法で実現した。

### 3. 対話支援メールシステム

われわれはグループのメンバー間のインタラクションを支援すべく PilotMail 電子メールシステムを開発した。人と人のコミュニケーションを成功させるためには、まず、送信者の意図が受信者に正確に伝わらなくてはならない。そこでわれわれは、電子メールという非同期コミュニケーション媒体を用いた場合に発生するコミュニケーションギャップを分析し、その中から誤解の発生原因とそれに対する防止策を探った。以下に誤解の発生原因とそれに対する PilotMail での対処策に関して議論する。

a) 受信者の行動が予測できない：従来の非同期分散型のコミュニケーションの場合、送信者はメッセージの受取人の行動を予測することができない。特にビジネスでは、「発言と行動は同一のものである」という考え方が根底になくなくてはならず、契約と行動の両方を管理することが重要である<sup>9)</sup>。たとえば、行動の約束をしても実際にその行動を起こさなければ、会話は完了したことにならない。そこで、送信者の意図を受信者に正確に伝えるためにも、また、発言と行動を調和させるためにも、メッセージを受け取った受信者がどのように振る舞えばよいのかを送信者がナビゲートしてやることが相互の誤解を抑える手法となりうる。PilotMail システムでは、Smalltalk の生成するオブジェクト (データ+手続き) を転送する機能をサポー

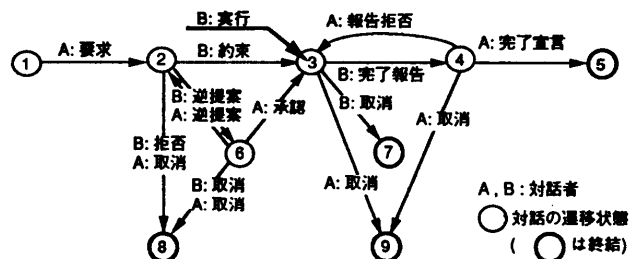


図1 会話の状態遷移モデル (文献4) から一部引用)

Fig. 1 The speech act theory.

トしており、発言（データ）と行動（手続き）を調和させるための手段を提供しているといえる。さらに、受信者によって評価・実行されるプロシジャを転送することができるので、送信者の意図に従って受信者が忠実に振る舞うことが保証される。

b) 対話の不連続性：従来の電子メールを用いたコミュニケーションにおいて、人は対話をしようとするのではなく情報の要点を簡潔に伝達することに専念しており、大量の語句の羅列を1通のメールに詰め込んで会話を成立させようとしている。さらに非同期コミュニケーションでは、対話が断続的にならざるをえず、会話に流れがなくなってしまう会話の進行状態や方向性が見失われやすい<sup>5)</sup>。一方、面と向かって会話する場合や電話で会話をする場合、相互理解を深めるためあるいは誤解を解消するために何度も会話を交わしたり行動の確認をとるといふ努力をしており、このように対話に流れを作ることがコミュニケーションを密なものにするための極めて有効な策となっている。われわれは、電子メールの利用形態に潜在するこれらの問題点を解決すべく、対話の進行を「構造化された状態へ移」であるとする Winograd と Flores の会話の遷移モデルに基づく会話管理機能と、元来 PilotMail の持つプロシジャ転送機能を融合した<sup>14)</sup>。

c) メッセージの伝達ミス：日常会話においても、話し手の言葉を聞きもらしたり、あるいは話し手が要点を伝え忘れることが多分にある。特に非同期コミュニケーションでは、送信者は要点の書き忘れがないようにしなければならないし、受信者は文面から要点を的確に読みとらなければならない。そこで、情報の要点が欠如することなく送信者から受信者に伝えられる仕組みが必要となる<sup>10)</sup>。PilotMail では、Lens システムのようにメールを構造化しており、メールは情報の内容に適合した構造（フィールド群）を持っている。これにより、受信者はそのメールの内容を正確に把握しやすく、送信者にとっても要点の書き忘れがなくなる。また、オブジェクト指向データモデルを採用し、メールの使用目的に応じたクラス階層化を施して管理した。クラスが階層化していることで、クラスは親クラスの機能・属性を受け継ぐことができ、たとえば、新たにメールのクラスを定義するときには親クラスとの差分のみの機能・属性を付加するだけでよい。

d) 情報整理の難しさ：正確に伝達されたメッセージであっても、そのメッセージを整理するときに失敗してしまう可能性がある。たとえば、重要な会議が数日

後に開催されるという通知がきても、それをスケジュールに組み込むのを忘れたならば、また、組み込んだとしても通知の内容を把握できなければその通知は無駄におわってしまうのである。すなわち、伝達される情報の形式は整理されやすいものであるのが好ましいといえる<sup>10)</sup>。PilotMail では、オブジェクト指向データモデルの採用により、メールの使用目的に応じたクラス階層化を施してメールを管理した。またメールを構造化することにより、その後の情報の整理および返答がしやすくなっている。例えば、「メールの特定のフィールドがルールのある条件を満足したならばあらかじめ指定したアクションが動作する」というような自動処理手続きを設定するための、PilotMail と協調動作するルールベースを構築することも容易にできる（未実装）。すなわち半構造化技術により、自然言語処理をすることなしに、コンピュータが情報の意味を理解できるようになっている。

以上のように PilotMail では、受信者へのプロシジャ転送機能とメールの半構造化技術を基盤技術として人と人の対話を支援している。プロシジャ転送機能によって、送信者の意図に従って受信者が忠実に振る舞うことが保証され、送信者が会話の行方や受信者の行動を予測できるようになり、信頼して処理を受信者に任せられるようになる。また、メールを半構造化することにより、コンピュータを用いたコミュニケーションを半自動化し、情報交換作業を迅速かつ柔軟なものにした。

#### 4. PilotMail システムの使用例

この章では、PilotMail の特徴的機能であるプロシジャ転送機能、会話管理機能、および、オブジェクト転送機能を説明するために、簡単なシナリオを挙げて PilotMail の使用方法について論じる。

##### 4.1 プロシジャ転送機能と半構造化技術

ここでは、簡単な具体的使用例を挙げて PilotMail が提供するプロシジャ転送機能および半構造化技術に関して説明をし、それと共に、実装されたユーザインタフェースにも言及する。ここに挙げる例は、ユーザ ichimura がグループミーティングを招集しようとしてそのミーティングの参加者に向けて通知を行う例である。ただし、グループの各メンバは自分専用のスケジュールマネジメントシステムを持っており、PilotMail システムと各人のスケジュールマネジメントシステムの間にはインタフェースが存在してい

る (このスケジュールマネージャは Smalltalk の Schedule クラスとしてシステム内に実装されている)。このスケジュールマネージャは、個人の仕事のスケジュールを管理するためのツールであるが、PilotMail システムとのインタフェースを備えることによりチームのメンバー間のコーディネーションのためにも利用できる。

階層化されて管理されている構造化メールのクラスの中から MeetingAnnouncement 構造を選択すると図 2 のデフォルトテンプレートが得られる。MeetingAnnouncement 構造を選ぶためには、まず、PilotMail ブラウザの上部左のカテゴリリストから Meeting-Support カテゴリを選ぶ。Meeting-Support カテゴリには、ミーティングに関連した数個のクラスがクラス階層に関係なく集められており、MeetingAnnouncement クラスもこのカテゴリに属している。テンプレートの左部はメールのフィールド名であり、それぞれの右部に、対応するデフォルトプロシジャが表示されている。図中の DATE: TIME: フィールド、PLACE: フィールドはそれぞれミーティングの開催日時、開催場所が記入されるべきフィールドであり、また、PARTICIPANTS: TOPIC: フィールドはそれぞれミーティングへの参加者、ミーティングの議題が記されるフィールドである。以上のフィールドは MeetingAnnouncement 構造に特有のフィールドである。一方、これら以外のフィールドは上位のクラスから継承されたフィールドであり、親クラスの属性を受け継いでいる (ただし親クラスで定義されている MAILCLASS: フィールドの属性は、MeetingAnnouncement クラスの中での再定義によって上書きされている)。

プロシジャの記述言語は Smalltalk を基盤としているが、図中の input は新たなデータ入力を促すためのクラスメソッドであり、例えば Date クラスのエディタがカレンダーのようなものであっても良いし、Place クラスの入力エディタが地図のようなものであってもよい。ユーザは自分の好みでクラスごとに専用の入力エディタを指定できる。

もしデフォルトのテンプレートを変更したい場合は、図 3 のように自由に変更できる。図では、ミーテ

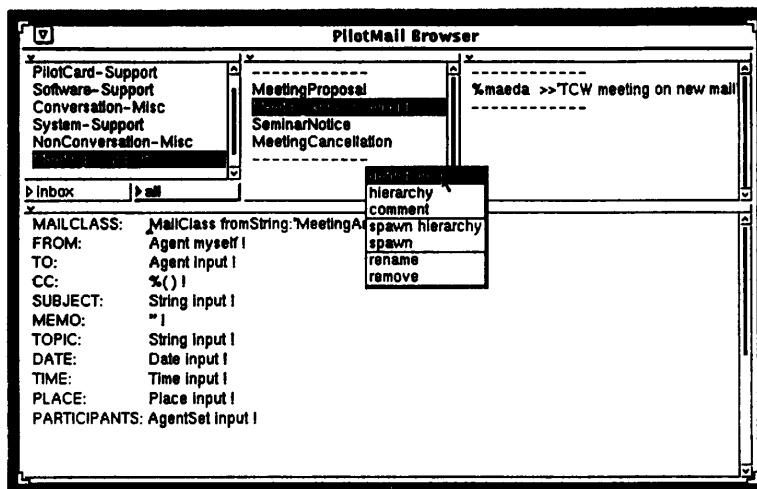


図 2 デフォルトテンプレート  
Fig. 2 The default template.

```
MAILCLASS: MailClass fromString:'MeetingAnnouncement' !
FROM: Agent myself !
TO: AgentSet input !
CC: %() !
SUBJECT: String input !
MEMO: "" !
TOPIC: SUBJECT !
DATE: Date today addDays: 3 !
TIME: Time input !
PLACE: Place input !
PARTICIPANTS: TO !
NOTIFY: TO doIt: [
    Schedule notify: SUBJECT
    date: DATE
    time: (TIME subtractHours: 1)] !
```

図 3 訂正後のテンプレート  
Fig. 3 The modified template.

ィングの開催日 (DATE: フィールド中のメッセージ) を 3 日後に設定し、さらに、NOTIFY: フィールドを追加して、ミーティングの 1 時間前に受信者のスケジュールマネージャのノーティフィケーション機能が起動するように設定している。Agent および AgentSet クラス (これらのクラスのインスタンスはユーザ識別子を保持している) の doIt: メソッドが受信者へプロシジャを転送するための命令である。

プロシジャの変更が終了した後、送信者はプロシジャの実行を行うが、上のフィールドから順に評価される (プロシジャの実行によりデータの入力を促すエディタが各フィールドごとに起動する) (ただし doIt: メソッドのブロック引数内は評価されない)。そして各フィールドにメッセージを入力した後、実行結果を見てそれが正しければメールを送信する (図 4 参照)。

電子メールで転送されるメッセージは図 5 のようなプレーンなテキストであり、そのフォーマットは UNIX 標準の書式であるので既存の電子メールを用

いてメールを転送することが可能である。送信者側の PilotMail システムは、オブジェクトを再構築するための文字列を自動生成し、受信者側の PilotMail システムは、その文字列の内容に基づいてオブジェクトを再構築する。オブジェクトを再構築するための文字列を生成する機能は、Smalltalk に標準で備わっている storeOn: メソッド<sup>19),20)</sup>を利用して実現されている。

メールの受信者は図4で示されるメールと同様のメールを受け取る(ただし、PilotMail ブラウザ上部の三つのリストサブビューには、新たに到着したメールのカテゴリ、クラス、インデックスがそれぞれ表示される)。受信者が依頼された処理を了解すればブロック内の手続きの評価を行い、これにより各自のスケジュールマネージャにノーティフィケーションの指示が伝わる。受信者に対しミーティングに参加するように発言によって指示するという従来のコミュニケーション形態を超えて、PilotMail が提供するコミュニケーション環境では、メールの受信者のスケジュールに予定を組み込むという行動までもが会話を構成する重要な要素の一つになっているのである。

#### 4.2 プロシジャ転送機能と会話管理

電子メールを用いたコミュニケーションに対話の流れを作るため、PilotMail では会話の状態遷移モデル(図1参照)に基づいた会話管理機能を提供している。ここでは、プロシジャ転送機能と会話管理機能の融合によって可能となる、行動の管理をも含めた対話の流れの管理法について例を挙げて説明する。この例は、ユーザ ichimura がグループミーティングを開催しようとして、メンバの都合を尋ねると同時に参加の依頼を行う例である。会話の状態遷移モデルを表す図1中の状態1において MeetingProposal 構造を選択するとデフォルトテンプレートが得られるが、図6は送信者によって ACTION: フィールドにプロシジャが記述されたものである。例では、ミーティングへの参加が可能であると回答したメンバに対してのみミーティングの1時間前にスケジュールマネージャのノーティフィケーション機能が起動するように設定されている。図中の 'CONVERSATION new' は対話の流れを管理するインスタンスを生成させるための命令、'CONVERSATION open' は会話管理マネージャを起動するための命令である。また、action: メソッド

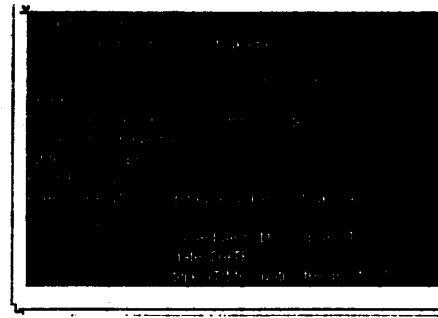


図4 データ入力後のテンプレート  
Fig. 4 The evaluated template.

```
##('MAILCLASS' (MailClass basicNew instVarAt: 1 put: 'MeetingAnnouncement'; yourself) ) #'FROM'
((Agent basicNew instVarAt: 1 put: 'ichimura'; yourself) ) #'TO' ((AgentSet new) add: (Agent
basicNew instVarAt: 1 put: 'okada'; yourself); add: (Agent basicNew instVarAt: 1 put: 'hiraiwa';
yourself); add: (Agent basicNew instVarAt: 1 put: 'tsukada'; yourself); yourself) ) #'CC' ((Ag
entSet new)) ) #'SUBJECT' 'CW meeting on PilotMail System' ) #'MEMO' '' ) #'TOPIC' 'CW
meeting on PilotMail System' ) #'DATE' (Date readFromString: '21 October 1991') ) #'TIME'
(Time readFromString: '1:00:00 pm') ) #'PLACE' (Place basicNew instVarAt: 1 put: '26-205';
yourself) ) #'PARTICIPANTS' ((AgentSet new) add: (Agent basicNew instVarAt: 1 put: 'hiraiwa'; y
ourself); add: (Agent basicNew instVarAt: 1 put: 'okada'; yourself); add: (Agent basicNew instVarA
t: 1 put: 'tsukada'; yourself); yourself) ) #'NOTIFY' '[ Schedule notify: SUBJECT date: DATE t
ime: (TIME subtractHours: 1) ]'' ) )
```

図5 電子メールで転送されるメッセージ  
Fig. 5 The message sent through E-mail.

```
MAILCLASS: MailClass fromString: 'MeetingProposal' !
FROM: Agent myself !
TO: Agent input !
CC: %(!)
SUBJECT: String input !
MEMO: '' !
TOPIC: String input !
DATE: Date input !
TIME: Time input !
PLACE: Place input !
PARTICIPANTS: AgentSet input !
ACTION: 'Schedule notify: SUBJECT
date: DATE
time: (TIME subtractHours: 1)' !
CONVERSATION: Conversation new action: ACTION !
CONV-MANAGER: TO doIt[ CONVERSATION open ] !
```

図6 初期会話テンプレート  
Fig. 6 The template for initiating a conversation.

によって対話の流れを管理するインスタンスに登録されたアクションは、状態3において実行される(このアクションは任意の会話遷移状態において変更可能である)。

メールが転送され、状態2において受信者が 'CONVERSATION open' プロシジャを実行すると、図7に示される会話管理マネージャが起動される。受信者は、現在の会話の遷移状態(図では二重丸となって表示されている)を視覚的に確認した上で、送信者からの要求を実行するか、逆提案をするか、拒否するかを決定することになる。図7のように要求を実行すると決定した場合、会話の状態は3に推移し、プロシジャ

で指定されているとおり、受信者のスケジュールマネージャにノーティフィケーション動作の予約がなされる。

このほか、会話マネージャは、約束どおりに仕事が進行しているか、進行中の案件にはどんなものがあるか、やり残した仕事は何かなどといった情報にアクセスするための機能を備えている。各ユーザの会話マネージャには現在進行中の対話がすべて登録されており、利用者は対話の進行状況を随時参照できる。

#### 4.3 プロシジャ転送機能とマルチメディアメール

郵送されてくる会議の案内状には、簡単な地図、イラスト、写真などがよく含まれているが、電子メールでもマルチメディアを交換できるようにしようとする研究や開発が盛んに行われている<sup>21)</sup>。これらのことから、マルチメディアメールの実現は、人間の意図や気持ちを正確に伝えるためにも有効であると思われる。PilotMail はオブジェクト転送を基盤としており、メールにマルチメディア情報を付随させて送ることが容易にできる。さらに、従来のマルチメディアメールとは異なり、PilotMail のプロシジャ転送機能により、マルチメディア情報のプレゼンテーションの方法を送信者が指定することができるので、より柔軟なマルチメディアデータ転送が可能である。

先ほどの会議通知の例を用いて、図3のメールに会議場への案内図をつける場合を想定する。PilotMail ブラウザのメニュー（マウスの中ボタンを押すと現れる）から utility コマンドを選択すると図8のユーティリティブラウザが現れ、イメージ画転送コマンドを選ぶことで、このブラウザの下部に表示されるフィールドが図3のメールに追加される。IMAGE: フィールド中の fromUser 命令は Smalltalk に標準で組み込まれているメソッドであり、ワークステーション・スクリーン上の任意の画像を切り出して、それをもとに Smalltalk の Image 型のインスタンスを生成するための命令である<sup>11)</sup>。送信者は、X-Window<sup>22)</sup>上で動作する任意のアプリケーションを使って地図を

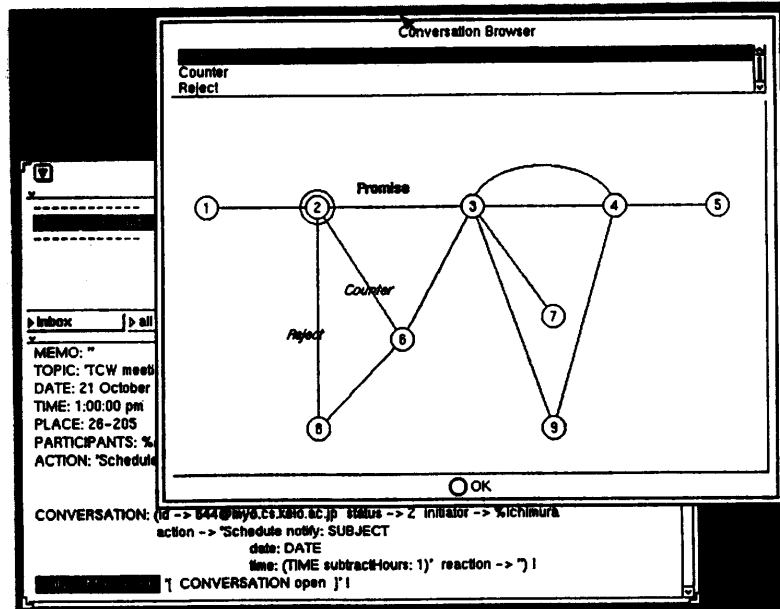


図7 会話マネージャ  
Fig. 7 The conversation manager.

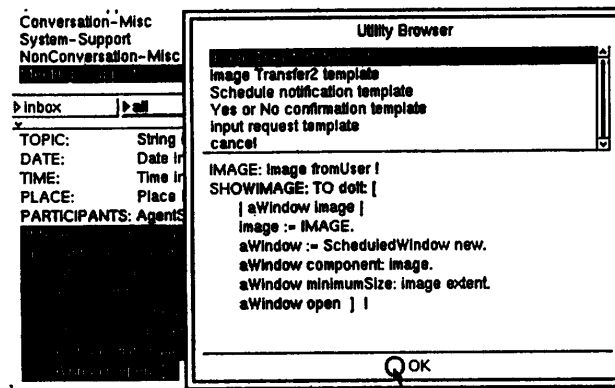


図8 イメージ画転送のためのテンプレート  
Fig. 8 The template for forwarding image data.

描画し、その地図を切り出してメールに付属させることができる。送信者によって生成された Image オブジェクトは受信者に転送されるが、SHOWIMAGE: フィールドに記述されているプロシジャによって、転送されたイメージ画は受信者のスクリーン上に表示される。このとき、送信者がプロシジャで指定したように、地図が Smalltalk の標準のウィンドウとして現れる。

## 5. プロシジャ記述言語

PilotMail システムのプロシジャ記述言語は、Smalltalk<sup>19),20)</sup> が基盤となっている。この Smalltalk の言

語仕様の一部を変更することにより PilotMail プロシジャ記述言語は記述できる。以下に PilotMail プロシジャ記述言語に特有の仕様を示す。

Agent クラスが Object クラスのサブクラスとして存在し、そのインスタンスはユーザを特定する ID あるいはグループを特定する ID を保持するものである。また、Agent クラスのインスタンスはリテラルとして記述することができる。Agent 定数は %ichimura のようにユーザ識別子に % を前置したもので、PilotMail システムが Agent クラスのインスタンスとして認識し、Smalltalk の解釈できる式に変換する。

AgentSet クラスが Set コレクションのサブクラスとして存在し、そのインスタンスは Agent クラスのインスタンスの集合を保持する。また、AgentSet クラスのインスタンスはリテラルとして記述することができる。AgentSet 定数は % (%matsushita %okada %tsukada) のように要素を % (と) でくくって書き、AgentSet クラスのインスタンスになるが、要素として書けるのは Agent 定数だけである。AgentSet 定数は PilotMail システムによって AgentSet クラスのインスタンスとして認識され、Smalltalk の解釈できる式に変換される。

PilotMail の使用例中に登場した、AgentSet クラスあるいは Agent クラスのインスタンスメソッドである doIt: はその引数にブロック式をとるキーワード・メッセージである。そのブロック式はメールの送信者が記述するが、そのブロックを評価・実行するのはメールの受信者 (Agent クラスのインスタンスによって特定されるユーザ) である。このメソッドによって、受信者へのプロシジャの転送 (未評価オブジェクトの転送) が可能となる。

## 6. PilotMail システムの利用評価

われわれは、グループの協同作業を支援する目的で PilotMail を開発してきた。したがって、システムの利用評価をするにあたってまず留意したのは、共通のゴールを共有し、協調して仕事を進めているグループを実験対象とするということである。現在、われわれの研究室において、同じ研究グループに所属する学生を対象とした評価実験を行っている。ここでは、現在までに与えられた評価を基に、問題点や今後の改良策に関して議論を行う。

われわれの実装したプロトタイプシステムの中には、受信者の振る舞いをガイドするためのサンプルプ

ロシジャがいくつか用意されており、ユーザは utility ブラウザに登録されているこれらのプロシジャを任意に組み合わせてメールの中に含めることができるようになっている。受信者のキー入力を促すためのプロシジャ、YES か NO を選択させるためのスイッチプロシジャ、選択項目から一つの項目を選ばせるためのリストプロシジャ、メールを受けとったことを送信者に通知させるためのプロシジャなどがサンプルプロシジャの例である (前述のシナリオに登場したスケジュールマネージャに対する操作もこのサンプルプロシジャの一つである)。被験者らは utility ブラウザ中のサンプルプロシジャを好んで使用し、メールの交換を行った。また、これらのサンプルプロシジャがわれわれの予期していなかった使われ方をされることもしばしばあった。たとえば、スイッチプロシジャやリストプロシジャは投票のためにも用いられた。このことから、PilotMail は柔軟かつ様々な形態をとって拡張利用されてゆく可能性があると確認された。ところが、Smalltalk を使用したことのない人は、簡単なプロシジャであれ、utility ブラウザ中に登録されていないプロシジャをキーボードから入力しようとはしなかった。しかしこれでは、Smalltalk 動作環境で PilotMail が稼働しているという利点を十分引き出すことはできない。このことから、utility ブラウザに重点をおいて改良を加え、より細かな記述ができるツールにしてゆく必要があると感じた。

マルチメディアメールに対する興味が以前から被験者らにあったということもあるが、PilotMail をマルチメディアメールとして用いるという使い方は、被験者から特に好評を得た使い方の一つであった。被験者らは utility ブラウザ中のイメージ画転送プロシジャを適宜メールの中に含め、画像転送を行った。たとえば、バグ情報の交換の際に、エラーメッセージやその症状が現れているスクリーンのスナップショットをバグ発生レポートの一部としてメールに含ませてプログラマに報告した。プロトタイプシステムでは音声や動画のサポートはしていないが、マルチメディアメールの実現は人間の意図や考えを正確に伝えるために有効であると確認された。

Winograd の会話の遷移モデルに基づいた会話管理機能に対しては、対話の流れを管理するという概念に慣れていないということもあり、被験者らはこれをどのような場面で使用したらよいかかわからないという状況がしばしば生じた。Coordinator のように、す

すべての会話を遷移モデルに従って行うように強制するという方法も考えられるが、すべての対話形態にこの遷移モデルを押しつけるということに関しては以前から様々な批判意見が出されている<sup>21,23)</sup>。事実、われわれの行った実験結果からも、どのような状況でこの会話管理機能を使えばよいのかを十分検討する必要があると感じた。しかしながら、会話管理機能が非同期コミュニケーション環境において対話の流れを作り出すために用いられたケースは多くあり、人と人のインタラクションを親密にするために有効であると認められた。

プロシジャの転送機能に対し、被験者らの間で最も危惧されたのはセキュリティに関する問題であった。グループ以外の悪意を持ったユーザがプロシジャを転送してきて、それを不用意に受信者が実行した場合には、確実にセキュリティは破られる。しかし、Pilot-Mail は協同作業を促進するために設計されたシステムであり、この意味から、一般的な汎用メールシステムとは性質を異にし、ワーキンググループ内および組織内でのみ用いられる、いわば、協同作業支援専用の電子メールシステムである。ゆえにグループ以外の人からのプロシジャは原則的に実行するべきではない。しかしながらグループ以外の人からグループのメンバになりすましてプロシジャを転送してきた場合まで考えると、暗号化技術の導入が必要になる。プロシジャを記述したのが同じグループのメンバであることを確実に認証でき、1対多通信にも適した暗号化方式であることが要求される。そこで、従来のコピー鍵方式<sup>24)</sup>の欠点を改良し、グループの構造変化にも柔軟に対応できる MCK (Modified Copy Key Management Method) 暗号化方式<sup>25)</sup>を導入することを現在検討している。

しかし、グループのメンバからのメールであれ、記述されたプロシジャが間違っていた場合、受信者側の作業環境において誤動作が発生するということであり、破壊的なダメージが受信者の作業環境にもたらされる可能性もある。この問題を解決するためには、Smalltalk のすべての命令を実行できるという Pilot-Mail の根本的な仕様を変更しなければならず、一般ユーザには直接 Smalltalk のコードを書く権限を与えてはならないということになる。しかし前述したように、Smalltalk を使用したことのない人は簡単なプロシジャであれプロシジャをキーボードから入力しようとはしなかったという実験結果から考えると、むしろ

一般ユーザは Smalltalk のコードに直接触れる必要性はないともいえる。一般ユーザが直接プロシジャ記述言語にアクセスできないようにし、utility ブラウザのようなツールのみが利用できるようにユーザインタフェースを設計するということに対応しなければならないであろう。

## 7. おわりに

本論文では、協同作業を支援する目的で開発された電子メールシステム PilotMail の設計理念、インタフェース、利用評価を、具体的使用例をまじえて述べた。また評価の結果、協同作業支援に効果を発揮した点および改善すべき問題点が明らかになり、今後の研究の指針を発見できた。評価に際しては研究室の学生を実験対象としたが、実際のビジネスの場において使用してみる必要があるだろう。また、現在、PilotMail システムとわれわれの開発したオブジェクト指向データベースを融合するプロジェクトも進んでおり<sup>26)</sup>、より応用価値の高いシステムにしていきたいと考えている。

グループウェアの課題である人間と人間をどのようにすれば密に融合できるかという問題に対して、われわれは、人間の気持ちや意図を他の人に伝えることができるようなシステムを構築するという方向からアプローチした。人間の気持ちや意図というものをコンピュータシステムとうまく結び付けることができるかどうか、グループウェア成否の大きな岐路となるだろう。

謝辞 研究の推進にあたりご指導いただいた岡田謙一講師ならびに同研究にたずさわった塚田晃司氏、伊藤陽子氏に感謝いたします。

## 参考文献

- 1) 松下 (編著): 図解グループウェア入門, オーム社 (1991).
- 2) 石井, 大久保: コンピュータを用いた人間の共同作業支援技術について, マルチメディア情報と分散協調シンポジウム 論文集, pp. 27-36, 情報処理学会 (1989).
- 3) Engelbart, D. and Lehtman, H.: Working Together, *BYTE*, Vol. 13, No. 13, pp. 245-252 (Dec. 1988).
- 4) Winograd, T. and Flores, F.: *Understanding Computers and Cognition*, Addison-Wesley (1986).
- 5) Winograd, T.: Where the Action Is, *BYTE*, Vol. 13, No. 13, pp. 256 A-258 (Dec. 1988).



- 6) Malone, T. W. and Crowston, K.: What Is Coordination Theory and How Can It Help Design Cooperative Work Systems?, *Proc. CSCW '90*, pp. 357-370 (1990).
- 7) Shepherd, A., Mayer, N. and Kuchinsky, A.: Strudel—An Extensible Electronic Conversation Toolkit, *Proc. CSCW '90*, pp. 93-104 (1990).
- 8) Woo, C. C.: A Tool for Automating Semi-Structured Organizational Communication, *Proc. COIS '90*, pp. 89-98 (Apr. 1990).
- 9) 市村, 平岩, 松下: PilotMail: グループ内会話モデルに基づいたメールシステム, 情報処理学会研究報告, 91-HI-35 (1991).
- 10) 市村, 松下: 誤解の発生を抑制するメールシステム, 人工知能学会第5回全国大会, Vol. 2, pp. 535-538 (1991).
- 11) Objectworks/Smalltalk User's Guide, Fuji Xerox Information Systems Co., Ltd. (1990).
- 12) Ichimura, S. and Matsushita, Y.: A Human Interaction-Based Mail System, *Proc. 6th JWCC*, pp. 153-160 (July 1991).
- 13) Ichimura, S. and Matsushita, Y.: A Mail System for Automating Communication Based on Object-Forwarding Mechanism, *Proc. ISCOM '91*, Vol. 2, pp. 638-641 (Dec. 1991).
- 14) 市村, 松下: グループ会話支援メールシステム PilotMail の実装と拡張利用, 第43回情報処理学会全国大会論文集, Vol. 1, pp. 259-260 (1991).
- 15) 市村, 松下: コンピュータとグループコミュニケーション—グループウェアからのアプローチ, 情報処理学会研究報告, 91-OS-52 (1991).
- 16) Malone, T. W. et al.: Semistructured Messages Are Surprisingly Useful for Computer-Supported Coordination, *ACM Trans. Office Information Systems*, Vol. 5, No. 2, pp. 115-131 (1987).
- 17) Pollock, S.: A Rule-Based Message Filtering System, *ACM Trans. Office Information Systems*, Vol. 6, No. 3, pp. 232-254 (July 1988).
- 18) Lai, K.-Y. and Malone, T. W.: Object Lens: A Spreadsheet for Cooperative Work, *Proc. CSCW '88*, pp. 115-124 (1988).
- 19) Goldberg, A.: *SmallTalk-80: The Language and Its Implementation*, Addison-Wesley (1983).
- 20) Pinso, L. J. and Wiener, R. S.: *An Introduction to Object-Oriented Programming and Smalltalk*, Addison-Wesley (1988).
- 21) Borenstein, N. S. and Thyberg, C. A.: Power, Ease of Use and Cooperative Work in a Practical Multimedia Message System, *Int. J. Man-Mac. Stud.*, Vol. 34, No. 2, pp. 229-259 (1991).
- 22) Young, D. A.: *X Window Systems Programming and Applications with Xt*, Prentice Hall (1989).
- 23) 中村ほか: グループウェア: 電子メールの普及を受けて模索始まるグループ・コミュニケーションの姿, 日経コンピュータ, pp. 48-65 (July 1989).
- 24) Kent, S. J.: Security Requirement and Protocols for a Broadcast Scenario, *IEEE Trans. Commun.*, Vol. COM 29, No. 6, pp. 778-786 (June 1981).
- 25) Nakamura, H., Takagi, K., Okada, K. and Matsushita, Y.: A Group-Oriented Key Management Method-GCK, *Proc. 5th JWCC*, pp. 229-236 (1990).
- 26) Ichimura, S. and Matsushita, Y.: A PilotCard-Based Hypermedia Network Integrated with a Layered-Architecture-Based OODBMS and an Object-Forwarding Mail System, *ACM Proc. Computer Science Conference*, pp. 431-438 (Mar. 1992).

(平成3年10月23日受付)

(平成4年5月14日採録)



市村 哲 (正会員)

1966年生。1989年慶應義塾大学理工学部計測工学科卒業。1991年同大学大学院理工学研究科修士課程修了。現在、同大学院理工学研究科博士後期課程計測工学専攻在学中。グループウェア、分散処理、ヒューマンインタフェースなどの研究に従事。人工知能学会会員。



松下 温 (正会員)

1939年生。1963年慶應義塾大学工学部電気工学科卒業。同年沖電気工業(株)入社。1968年イリノイ大学大学院コンピュータサイエンス学科卒業。1989年より慶應義塾大学理工学部計測工学科教授。工学博士。マルチメディア通信および処理に関するコンピュータネットワーク、分散処理、グループウェア、ヒューマンインタフェースなどの研究に従事。「コンピュータ・ネットワーク」(培風館)、「コンピュータネットワーク入門」(オーム社)、「インテリジェント LAN 入門」(オーム社)、「人工知能の実験」(近代科学社)など著書多数。電子情報通信学会、人工知能学会、IEEE、ACM、フェジィ学会各会員。