

PilotMail: グループ会話支援メールシステム*

4 T-7

市村 哲 塚田 晃司 伊藤 陽子 松下 温†
慶應義塾大学‡

1 はじめに

オブジェクト指向モデルをグループ内通信に適用し、チームのメンバー間で行うメールのやり取りをオブジェクト間のメッセージパッシングと捉えることにより、グループの多様な会話モデルに動的に対応できるメールシステム PilotMail を考案した。このメールシステムの主要目的は、人と人がコミュニケーションする際にしばしば発生する「誤解」を抑え、相互コミュニケーションを親密にし協同作業を促進させることにある。

2 チームウェアとメールシステム

実際の社会組織ではグループによる協調作業が活動の基礎となっており、その中では人と人あるいは集団と集団間のコミュニケーションが無数に行われる。我々の提唱する「チームウェア」では、グループ内コミュニケーションのうち特に蓄積型分散環境でなされるものに対して支援をしている。我々は「チーム」を、「各自が達成目標を確実に把握しつつ協調して実作業を進めるエキスパートの集団」と定義しているが、グループワークでは、当初予定していたスケジュール通りには作業が進行せずそのための連絡を互いにとりあったり、個人が作業をしている際に発生した問題点を打ち合わせによって解決しなければならぬ状況に遭遇する。すなわち、他の人と進行を合わせたり、作業を依頼したり、必要な情報を交換しながらチームの作業は進むのである。

グループウェアを従来のシステムと区別する重要な要件の一つとして、Winograd は人と人との協調構造に基づいて設計されたシステムであることを挙げているが、これはグループウェアが人間のインタラクション構造に基づいて設計されるべきであることを指している [1]。すなわち、人間の行う会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ことを示している。この必要性ゆえ、最近、会話モデルを基盤としたグループウェアが注目されるべき研究対象と成りつつある。

ところが、電子メールは今や重要な通信手段に成長したにもかかわらず、人間のインタラクション構造に基づいて作成されておらず、グループの会話モデルを考慮していない。このために、コミュニケーションのためのツールでありながら、パーソナルユースのためだけに設計されていると言わざるを得ない。

3 PilotMail システム

コミュニケーションの際に発生する「誤解」を抑えるために、送信者の意図をできるだけ正確に受信者に伝えるために、我々は以下の2つのアプローチを行った。

受信者への情報処理手続きの転送: これにより送信者の意図にしたがって受信者が忠実に振る舞うことが保証される。すなわち、送信者が会話の行方を予測でき、信頼して処理を受信者に任せられるようになる。しかしながら、他人から要求を鵜呑みにして履行することは煩わしいだけでなく危険をはらんでおり、他人から依頼された要求であっても自分なりに解釈して実行できる機構が必要となる。そこで、メッセージに対する最低限の共通の認識が存在しさえすれば、オブジェクト内部の具体的な処理を詳細に知る必要がないというオブジェクト知識表現をメールシステムに採用した。この様にオブジェクト指向知識表現の計算モデルはチームの作業の進め方に非常に近いものを持っており、複数の制御権を持ったもの同士の関係が自然な形で表現できるという特長を持っている。

メールの構造化: Lens システムのようにメールを構造化し、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管理した。このようにメールが情報の内容に適合した構造(フィールド群)を持っているため、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り、また、チームで共通の構造を定義しておけば共通の情報形式でメールが伝わるため、その後の情報の整理および返答がしやすくなる。しかしながら、メールを仕事の種類に応じて構造化してしまうと、例外的な仕事や処理を他人に依頼する場合、結局プレーンなテキストとし

*PilotMail: Group Conversation based Mail System

†S. Ichimura K. Tsukada Y. Itoh Y. Matsushita

‡Keio University

てその依頼を記述しなくてはならない。そこで我々は、任意のメールに任意のフィールドを追加して、そのフィールドにデータ処理手続きを記述できるようにし、非定型業務および複雑な処理の依頼への対応性を高めた。

つぎに、簡単な具体的使用例を挙げて我々のメールシステムを概説する。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（斜体字は新たに入力した文字）。

TO:	AgentSet input.
SUBJECT:	String input.
DATE:	Date input.
TIME:	Time input.
PLACE:	Place input.
PARTICIPANTS:	AgentSet input.
TOPIC:	String input.

図 1: デフォルトテンプレート

MeetingAnnouncement 構造を選択すると図 1 のデフォルトテンプレートが得られる。図でわかるように手続きの記述言語は smalltalk-80 を基盤としているが、図中の AgentSet は Agent のみが入る Set である。また、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。

DATE:	Date today addDays: 3.
TIME:	Time input.
PLACE:	Place input.
PARTICIPANTS:	AgentSet input.
TOPIC:	String input.
TO:	PARTICIPANTS.
SUBJECT:	TOPIC.
NOTIFY:	PARTICIPANTS doit: [Schedule notify: SUBJECT date: DATE time: (TIME subtractHours: 1)].

図 2: 訂正後のテンプレート

もしデフォルトのテンプレートを変更したい場合は、図 2 のように自由に変更できる。図では、ミーティングの開催日を 3 日後に設定し、更に、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ーのノーティフィケーション機能が起動されるように設定している。図のように Agent および AgentSet クラスの doit メソッドが受信者へのデータ処理手続きの転送をするための命令であるが、引数のブロック内のみが受信者へ転送される。

手続きの変更が終了した後手続きの実行を行うが、上に書いてあるフィールドから順に（ただし doit メソッドのブロック引数内は評価されない）実行される（図 3）。そして、実行結果を見て、それが正しければメールを送信する。

DATE:	March 15, 1991
TIME:	13:00:00
PLACE:	Matsushita lab
PARTICIPANTS:	Matsuura, Tsukada, Hiraiwa
TOPIC:	TCW meeting on new mail
TO:	%(Matsuura, Tsukada, Hiraiwa)
SUBJECT:	'TCW meeting on new mail'
NOTIFY:	%(Matsuura, Tsukada, Hiraiwa) doit: [Schedule notify: SUBJECT date: DATE time: (TIME subtractHours: 1)].

図 3: データ入力後のテンプレート

メールで転送されるデータは下図のようなプレーンなテキストであり、既存の電子メールを利用できる。

To: Matsuura, Tsukada, Hiraiwa
Subject: TCW meeting on new mail
CLASS:(String)MeetingAnnouncement

NOTIFY:(Proc)[Schedule notify: SUBJECT date: DATE \ time: (TIME subtractHours: 1)].

FROM:	%Ichimura
SUBJECT:	'TCW meeting on new mail'
DATE:	March 15, 1991
TIME:	13:00:00
PLACE:	Matsushita lab
PARTICIPANTS:	%(Matsuura, Tsukada, Hiraiwa)
TOPIC:	'TCW meeting on new mail'
NOTIFY:	[Schedule notify: 'TCW meeting on new mail' date: March 15, 1991 time: (13:00:00 subtractHours: 1)].

図 4: 受信者の受け取るテンプレート

ミーティングの参加者の手にするメールは図 4 である。受信者が依頼された処理を理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ーにノーティフィケーションの指示が伝わる。スケジュールマネージャーの振る舞いは各自が独自に規定するものであり、チームでの共通の認識を守れる範囲で notify メソッドを変更できる。

4 おわりに

送信者の意図を正確に受信者に伝えるために、構造化と情報処理手続き転送に基づいた PilotMail システムを提案した。現在、このメールシステムを拡張して、掲示板機能、協同執筆支援への応用を検討しており、我々の開発したチームウェアデータベースと融合させることにより更に利用価値の高いシステムへと発展させたいと考えている。

参考文献

- [1] Terry Winograd, Fernando Flores, "Understanding Computers and Cognition", Addison-Wesley Publishing Company Inc., 1986.