

1M-4

ユーザとの対話機能をもつメール InterCam

奥村成吾<sup>\*1</sup>, 布川 博士<sup>\*2</sup>, 宮崎 正俊<sup>\*1</sup>, 野口 正一<sup>\*3</sup>

\*1 東北大学教養部 \*2 東北大学電気通信研究所 \*3 東北大学応用情報学研究センター

1 はじめに

現在、コンピュータネットワークを用いた人間の間のコミュニケーションの手段として、電子メールの利用は重要な役割を持っている。電子メールは蓄積形のデータ交換であるため時間の制約を解決できる(非同期性)、多くの相手に同じに送信できる(同報性)などの利点がある。

しかしながら、現在のメールシステムは、(1)テキストベースとし、(2)読み書きを各ユーザが持っている単純なメール読み書きプログラムを利用する、が現状である。

そのためユーザは、(1)ごく単純なメールを返信する場合でも、テキストエディタの使い方など複雑な操作を知らなければならない。(2)メールの内容に応じたインタラクションができない(対話性の欠如)等の欠点がある。そのため、メールによって伝えたいこと、メールによって相手から得たいことなどの本質的なコミュニケーションが、非本質的な操作によりフィルタリングされてしまう、といった問題がある。

本稿では、これらの問題を解決するために、電子メールとしての利点を持ち、かつ、コミュニケーションの内容にあわせた対話機能を持つカード型のメール、InterCam (InteractiveCard Mail) について述べる。

2 InterCam

InterCam はメール自身がユーザとの対話機能を持っているカード型のメールである。InterCam メールは通常のメール転送システムによって転送されるため、通常のメールが持つ利点はすべて有している。それらに加え、コミュニケーションに応じた対話性さらに、特定の機種に依存せず、複数のネットワークでも利用可能なオープン性も有している。

InterCam システムにおける一つのメールは一枚のカードによって表現される。このカードにはユーザとのインタラクションを行なうためのボタンや、文字の入出力を行うためのフィールド、ビットマップなどが含まれている。

図1にInterCamメールの例を示す。ユーザはメールに対して、参加可能な日時を答え、送信ボタンを押すことにより返信される(この場合はリストの形で("新年会" "1993年1月" ("17" "18")))が返信される)。このようにInterCamメールでは、それぞれのコミュニケーションの内容に応じた対話機能が、メール自身の中に含まれている。

3 InterCam 記述言語

InterCamのメールはInterCamメール記述言語によって記述されている。その記述をInterCamインタプリタにより実行(インタプリタによる評価)することにより、メール自身がユーザとインタラクションを行なう(図2)。

図3に図1のメールに対する記述を示す。図3の例からわかるようにInterCam記述言語は極めて高いレベルの記述のみを規定している。そのため、インタプリタを異機種上に実装することが容易に可能である。InterCamインタプリタ実現されていれどどのような機種でも同様のインタラクションが可能となり、これによりInterCamのオープン性を確保している。

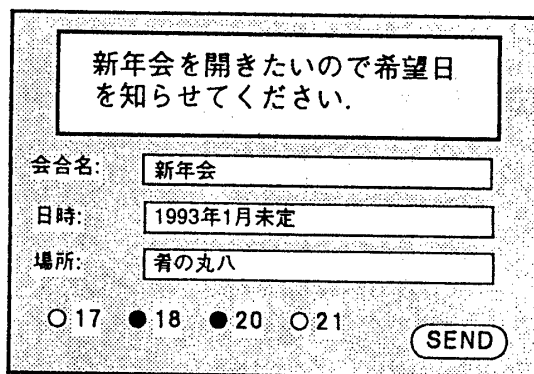
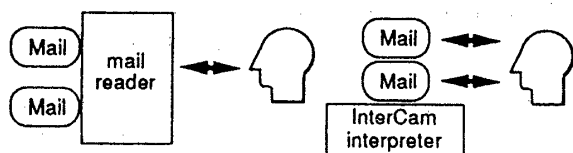


図1 InterCamのメール例



(a) 従来のメール (b) InterCamのメール

図2 従来のメールとの比較

Interactively Communicating Electrical Card Mail : InterCam  
 Okumura Seigo<sup>1</sup>, Nunokawa Hiroshi<sup>1</sup>, Miyazaki Masatoshi<sup>1</sup>, Noguchi Shouichi<sup>1</sup>  
<sup>1</sup> College of General Education, Tohoku University  
<sup>2</sup> Research Institute of Electrical Communication, Tohoku University  
<sup>3</sup> Research Center for Applied Information Sciences, Tohoku University

#### 4 InterCamの実装

InterCamシステムの実装をするには各機種用のInterCamインタプリタを実装する。その方法には、(1)InterCamをそのまま実装する方法、(2)分散記述言語DeLisを実現し、その上に(DeLisのアプリケーションとして)InterCamインタプリタを実装する、の2つがある。

- (1)の利点は、InterCamインタプリタ自身が、InterCamを利用するための(ソフトウェア、ハードウェアの)資源が少ないということである。そのため、通常のパソコン通信に使用する程度のコンピュータ、ネットワーク上でもInterCamは利用可能である。反面(2)で述べる利点はなくなる。
- (2)の利点は、InterCam自身の機能の向上にともなうInterCamインタプリタの記述関数(増加や削除を含む)変更を、InterCamメール中にInterCamシステム記述言語であるDeLis関数を含ませることにより、InterCamシステム内で行なえることである。現在のシステムではこの機能を用いてテンプレートを實現している。

(1)(2)いずれの方式を用いてもInterCamとしては同様の対話機能を持つ。(2)の方式で實現したシステムを図4に示す。図中DeLisの上に機種間の違いを考慮してInterCamインタプリタを記述する関数群を用意した。その上でInterCamインタプリタを記述し他機種での実装性を高めている。

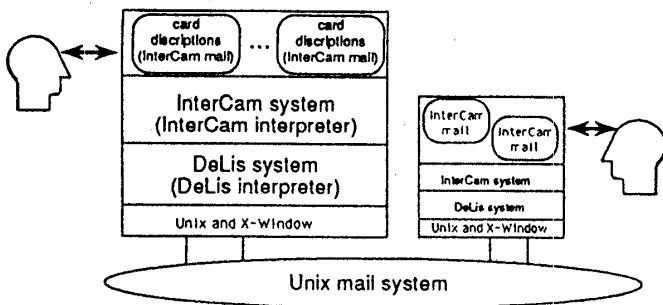


図4 DeLisによるInterCamの実装

#### 5 テンプレート機能

InterCamのカードの作成は、図3を見てもまだまだ労力が必要である。電子メールは決まった形式が多いので、必要なカード形式をテンプレートとして定義し、利用すればメールの作成は容易になる。

テンプレートは、カードが引数をもった形である。すなわち、テンプレートは、カードがテンプレートとして登録されているものである。図3の記述においてコメント、会合名、場所などを引数としたカードを定義し、それをテンプレートとして登録すればよい(図5に示す)。

しかし、この機能を用いた場合、送り主が相手の持っていないテンプレートを使用したメールを送った時、受けとり主が

メールを読めないことが起こり得る。それに対して、受けとった側はテンプレートがない旨を送り主に知らせてテンプレートを送るという作業を行なっている。これは、4の(2)で述べた方式を用いて實現されている。

#### 6 まとめ

今回、我々が作成したシステムにより、非同期かつインタラクティブなメールシステムを實現でき、単なるメールを越えたカードという情報が飛び交う世界の一端を見ることができた。今後の課題として

- メールを記述する環境の整備
- 単に相手からでなくテンプレートのサーバ
- InterCamのマルチメディア化

などがあげられる。また現在、4で述べた(1)の方式によるPC版InterCamの作成を試みている。

#### 参考文献

- [1] 布川, 三石, 宮崎, 野口: 分散環境のための言語系, 第45回情報処理学会全国大会講演論文集 pp 5.77-5.78

```
(defineCard 'kaigou ; カード名 kaigou でカードを定義
  (400 300) ; カードのサイズ

; メッセージを定義
(setMessage 'message "新年会を開きたいので希望日を知らせて下さい")
(setMessage 'name "新年会")
(setMessage 'date "1993年1月未定")
(setMessage 'place "有の丸八")
(setq hopedaylist '(17 18 20 21))

; メッセージフィールドを作成
(createField 'msg_1 (10 10 380 80) message Font1)
; 会合名のフィールドを作成
(createLabel 'name_1 (10 100 100 40) "会合名" Font2)
(createField 'name_1 (120 100 170 40) name Font1)
; 日時のフィールドを作成
(createLabel 'date_1 (10 160 100 40) "日時" Font2)
(createField 'date_1 (120 160 170 40) date Font1)
; 場所のフィールドを作成
(createLabel 'place_1 (10 220 100 40) "場所" Font2)
(createField 'place_1 (120 220 170 40) place Font1)
; チェックボタンを作成
(createGroupCheckButton 'chk_bt (10 250 300 40)
  hopedaylist Font1)
; メール返送のボタンを作成
(createStrButton 'send (320 250 70 40) "SEND" Font1
  (setq ReturnMail (list name date (getCheckButton chk_bt
    'checked))))
  (returnMail ReturnMail)
  (exit))
); カードの定義の終了

(createCard kaigou); カードの實現
```

図3 図Aのメールに対する記述

```
; カードの定義
(defineCard 'kaigou
  (message name date place hopedaylist); 引数
  (400 300) 図Cと同様)
; テンプレートの定義
(defineTemplate 'kaigou
; カードの實現(引数を与えている)
(createCard
  "新年会を開きたいので希望日を知らせて下さい"
  "新年会" "有の丸八" "1993年1月未定"
  '(17 15 18 20))
```

図5 テンプレートによる記述