

電子メールシステムにおけるメールボックス設計の一手法

落水浩一郎 田中勝 平田明良 今泉恵美子
(静岡大学 工学部 情報工学科)

1. はじめに

本報告では、電子メールシステムの設計事例をもとに、パーナス・モジュールの認識法と、そのシステム機能全体に果たす役割についての考察結果を述べる。

論文(1)~(7)の内容によると、パーナス設計法によるシステム構造は以下のような特徴をもつ。

図1において、A部は入出力データ、B部はシステム内部に一時的、または恒久的に保持されるデータ、C部は入力から出力への変換論理と実現する部分である。

パーナスの情報いんぺい原理は、A部、B部に適用され Secret と呼ばれるパーナスモジュール群が実現される。

A部の Secret は、システムに対する入出力データの様式(format)の変化を吸収するためのメカニズムを実現する。すなわち、表1に示すように、システムが対象とする入出力データについて、あらかじめ予測できる(anticipated change)様式をすべて列挙し、その情報内容を認識した上で、abstract interfaceとしてサブプログラムを与える。

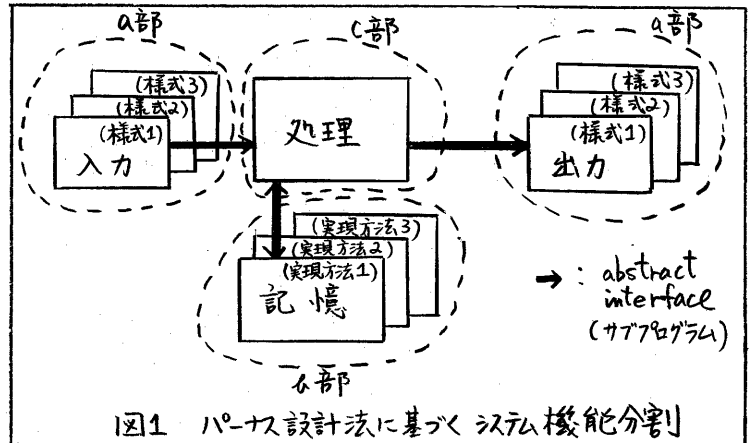


図1 パーナス設計法に基づくシステム機能分割

様式1 February 10, 1941	} → 年, 月, 日 →	YEAR MONTH DAY
様式2 10 February 1941		
様式3 2/10/1941		
様式4 41 February 10		
(actual interface)	(様式をとり除いた 情報内容)	(abstract interface)

表1 図1 A部に対する abstract interface の例 (6)(7)

B部には、システム内部に一時的または恒久的に保持される情報に関して、その実現の詳細(データ構造)と、実現方法に依存する処理(アクセス法)の詳細をいんぺいする。すなわち、システムの中核的な機能と達成する処理部(C部)において、その動作論理と構築するのに必要最小限の情報(抽象データ)のみをサブプログラムにより提供する。(2)(3)

C部の論理は、A部、B部から提供される abstract interface を用いて構築される。このとき、A部、B部はそれぞれ、実様式、実現の詳細を抽象した役機械として実現されるが、図1に示すように、システムは単一ユーザ専用ではなくA部、B部の具象例を組み合わせた構造になっている(program family)(4)。

システムのある特定の版はA部、B部内に定義された機能の最小単位(minimal subset)と uses relations によって組みあがられる(6)(7)。

以上が D.L. Parnas による一連の研究成果の概要であるが、不明な点がある。

(1) 各部のパーナス・モジュールの認識法が明解でない。

(2) C部と各部の境界、または、各部がシステム機能全体に対して果たす役割が明解でない。の2点である。

2. システム機能の分割とパーナス設計法

要求仕様をもとに、システムの初期設計をおこなう際、開発管理の問題もあわせて考慮する必要があり、代表的なものの一つにシステム機能の分割がある。

分割の必然性は以下の点にある。納期は契約によって定まる。一人の人間が作りあげうるステップ数/人・日には上限がある。定められた期間内に完済するためには、複数人で仕事をするオーバヘッドを十分承知の上で、一人の人間が作りあげべきソースステップ数を減少させるために、作業を並列化せざるを得ない。そのためには、分割単位(一人でプログラミングをおこなう範囲)を定め、かつその間の効率的作業手順(つくる効率)を定める必要がある。作業単位間の独立性が保証されるためには、分割単位内のインタフェースデータを完全に定める必要があるが、その決定の良し悪しは、少し先に進んでみるければわからないという現実がある⁽⁸⁾。その意味において、分割と変更の問題は同時に議論されるべきである。むしろ、変更の問題をより積極的にとりあつかうことによる利点の方が大きい。何故ならば、開発と保守の背後にある基本的な構造は「決定の構造」である。開発とは、「新しく決めていくこと」であり、保守とは「決定を変更すること(再決定)」である。開発時にも決定の変更はなされるので、開発と保守の違いは、既に決められているものと新しく決めるものの量的な差にすぎない⁽⁸⁾。決定の内容、表現法の問題は構造化プログラミング以来、議論されており⁽⁹⁾、パーナス法においても決定されていく情報群の分配法が中心の課題である⁽¹⁾。ところで、複数人の人間が並列に作業を進めうるための最低限の保証は分離コンパイルであるが、このような分割は、一人でプログラミングをおこなう範囲内での分割方法論である。システム機能の分割は、要求仕様中で定義された情報とその操作の世界をシステム機能に変換する過程において認識されるべき性格のものである。図1のA部は、入出力仕様であり、要求仕様中に明示されるものであるので認識が容易である。しかし、B部は、要求仕様中の情報とその操作の世界から、データ型を発見していく過程で見出されるものであるので、要求仕様との関係を論じなければ焦点が定まらなくなる。この結果、1の(1),(2)の問題が生じているものと思われる。3節以降に、この問題に対する一つの検討結果を示す。

3. T-NET メールシステムの概要

T-NETメールシステムとは、公衆電話回線を利用したネットワークの応用レイヤの一つであり、画像情報を含む文書情報をメールとしてネットワークサイト間のユーザ間で交換する機能をもつ。

転送サブシステムTは、メールプロトコルを実現し、メールを送受信する。エンドユーザサービスサブシステムEは、送信メールの編集作成、受信メールを出力する。両システムのインターフェースはメールボックスで実現される。

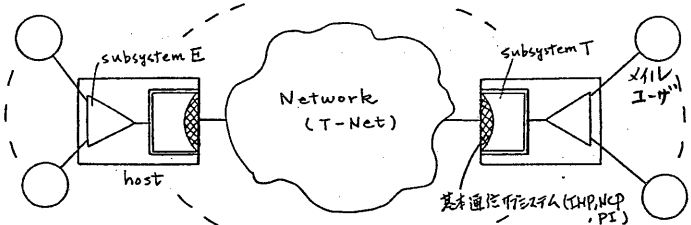


図2 T-NET メールシステムのシステム構成

本システムの詳細については文献(10)を参照して頂きたい。ここでは、4節以降の説明の準備として、メールシステムがとりあつかう情報と、ネットワークコマンドレベルでの情報操作を説明する。

3.1 メールの構成情報

①画情報はFAXから入力し、マーク位置に従い、切出しがなされる。この時、縦/横の長さ(mm)も計算される(図3)。②コード情報はエディタで入力する。③画情報は必要なら、SPC法でサイズ変換をおこない、縦/横ドット数を計算した上で二次元MH法で圧縮し、図4の形式で送信用領域に登録する。圧縮された画情報と縦/横ドット数(情報量)の組を画型実領域と呼ぶ。④コード情報は、制御コードをとり、一行の最大文字数にあわせて空白の詰めものをし、送信用領域に登録する。整形されたコード情報と縦/横文字数の組をコード型実領域と呼ぶ。⑤仮想ファクスと互換機の走査線密度のみを標準化して定義し、仮想ファクスへ出力するためのdisk上のバッファを仮想ページと呼ぶ。これにより、図5に示す座標空間が定義される。実ファクスへの出力はSPC法により線密度変換を施しておこなう。⑥メールはページの集まりである(図6)。ページと仮想ページは実質は同じである。⑦ページは、領域からなる(図7)。⑧領域には画型領域とコード型領域の2つの型がある。⑨画型領域は画型実領域を仮想ページ上におくときの左上隅の仮想ページ上のアドレスを付加することで定義される。これを編集条件という。⑩コード型領域は、コード型実領域に、アドレスと1文字当りの縦/横のドット数を付加することで定義される。これも編集条件と呼ぶ。⑪仮想ページ上に配置される領域内に領域変算子を定義する。"十"の変算は透明に、"."の変算は不透明にしてオ2オペランドとオ1オペランドにはりあわせる。はりあわせの構造を表現する式をページ構成式と呼ぶ。⑫メール内の各ページは領域を共有できる。⑬1ページ内に存在する領域と、領域をつくるために利用される実領域との間には多対1の対応を許す。⑭メールの出力はすべてFAXを利用しておこなう。

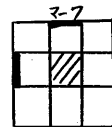


図3 画情報

repeat	data count	圧縮データ
--------	------------	-------

以下n本の走査線は同じ情報
→ 圧縮データのビット長

図4 走査線の表現

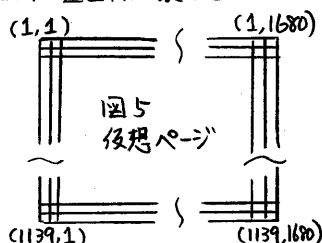


図6 メール

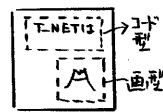


図7 ページ

領域名	属性	情報量
	コード型 画型	

図8 送信領域管理テーブル

⑮仮想ページ上に配置される領域内に領域変算子を定義する。"十"の変算は透明に、"."の変算は不透明にしてオ2オペランドとオ1オペランドにはりあわせる。はりあわせの構造を表現する式をページ構成式と呼ぶ。⑯メール内の各ページは領域を共有できる。⑰1ページ内に存在する領域と、領域をつくるために利用される実領域との間には多対1の対応を許す。⑱メールの出力はすべてFAXを利用しておこなう。

3.2 メールの管理情報

①サイトのメールユーザ毎にメールボックスが1つ存在する。②メールユーザは、メールディレクトリ木の1つのpathに対応して、ユーザ名とキーが与えられ、木の葉にメールボックスが定義される。③メールボックス内には3種の情報がある。送信メール、受信メール、送信用領域の3種であり、それぞれ、送信メール管理テーブル、受信メール管理テーブル(図9)、送信領域管理テーブル(図8)で管理する。送信メール管理テーブルは、

mail-id	時刻	発信名	ページ1構成式	領域1	属性	情報量	編集条件
				領域n	"	"	"
			ページm構成式	領域1	属性	情報量	編集条件
				領域l	"	"	"

図9 受信メール管理テーブル

発信人名の欄が存在しないことを除いて、図9と同じである。

3.3 ネットワークコマンドとシステム機能

システムの機能を定めているコマンドと、3.2で述べた情報群との関係を表2にまとめる。ユーザが存在する側のシステムTを T_u 、対向ホストのシステムTを T_s と記す。

T	(1) $\text{@USER} \langle \text{user-id} \rangle, \langle \text{key} \rangle$	T_u では、メールアドレスの探索と、おこない、アクセス権の獲得とメールボックスのオープンとする。 T_s は、 user-id を保持することにより、メールの発信人を知る。
	(2) @NLIST	T_s のメールアドレスを走査して、登録されているすべてのユーザ名を知る。
	(3) $\text{@DESCRIBE} \langle \text{mail-id} \rangle$	T_u は、実領域の内容 3.1③を除く、メールの記述情報を送出する。 T_s は、それを受取る。 T_u, T_s とも @MAIL コマンドとの対応をつけるため mail-id を記憶する。
	(4) $\text{@MAIL} \langle \text{mail-id} \rangle$ $TO \langle \text{address} \rangle$	T_u は、以前に @DES コマンドで記述したメールに対応する実領域をおくり、 T_s は、それを受取る。 T_s は address をもとに、メールアドレスを探索して、配信人名(複数可)を知り、それぞれのメールボックスに(3)の情報とあわせて受信メールとして登録する。
E	(1) $\text{ACCESS} \langle \text{user-id} \rangle, \langle \text{key} \rangle$	メールアドレスの探索による、アクセス権の獲得とメールボックスのオープン
	(2) $\text{ENTRYC} \langle \text{text-file-name} \rangle$	指定された入力ファイルをもとにコード型実領域をつくり、送信領域管理テーブルに登録する。
	(3) $\text{INPUT} \langle \text{file-name} \rangle$	3.1の①
	(4) $\text{SIZE} \langle \text{size} \rangle \langle \text{file-name} \rangle, \langle \text{file-name} \rangle$	3.1の③のうちサイズ変換部分
	(5) $\text{ENTRYG} \langle \text{file-name} \rangle$	画型実領域を作成登録する 3.1③
	(6) CATALOGS	送信領域管理テーブル中に登録済の実領域に関する情報のリスト
	(7) CATALOGR	受信メール管理テーブル中に存在する、すべての受信メールに関する情報のリスト
	(8) CATALOGD	送信メール管理テーブル中に存在する、すべての送信メールに関する情報のリスト
	(9) $\text{ERASES} \langle \text{area-name} \rangle$	送信領域管理テーブル中の実領域の削除。ただし、削除しようとしている実領域
	(10) $\text{ERASER} \langle \text{mail-id} \rangle$	送信メールの削除
	(11) $\text{ERASED} \langle \text{mail-id} \rangle$	受信メールの削除
	(12) $\text{OUTPUTR} \langle \text{mail-id} \rangle$	受信メールをFAXに出力する。
	(13) $\text{OUTPUTS} \langle \text{mail-id} \rangle, R$	送信用領域中の領域について、領域そのもの、あるいは複数の領域の演算結果を確認のためFAXに出力する。 R E指定すると、作成エリアとして、送信メールとして登録する。

表2 コマンドと情報操作

4. システム内部のデータ設計とパーナスモジュールの認識

パーナス法は、(入出力を含む)情報の定義場所(図1のa部、b部)と、その利用場所を分離する方法論である。両者の間には $\text{abstract interface}$ で結合される。

$\text{abstract interface}$ を実現するサブプログラムは定義部にまとめられる。この結果、「情報定義」を実現するソースコード部と、「情報を利用する」を実現するソースコード部も分離される。パーナス法の効果は、前者の部分がソースコード全体に散在することを防止し、情報を利用する側には、 $\text{abstract interface}$ 名(サブプログラム名)のみが出現することであり、この結果、変更管理が容易になる。

このことから、その効果をより大きくするためには、利用(参照)の箇所が多い情報をパーナス・モジュール中にとじこめる必要がある。本システムでは、(@Mail , $\text{outputs} \langle R \rangle$, outputR , @DESCRIBE)が達成しようとしているシステム機能間に、3節で述べたレベルでの情報とその操作に関して、多くの重複が予想され、パーナスモジュール認識のきっかけがある。

このような方針のもとに、要求仕様中の情報とその操作の世界を単文単位に分解し、一目で見わたせるような図を作成した。図10にその概要を示す。

このとき、単純にパーナスモジュールを決め、 $\text{abstract interface}$ と定義することと

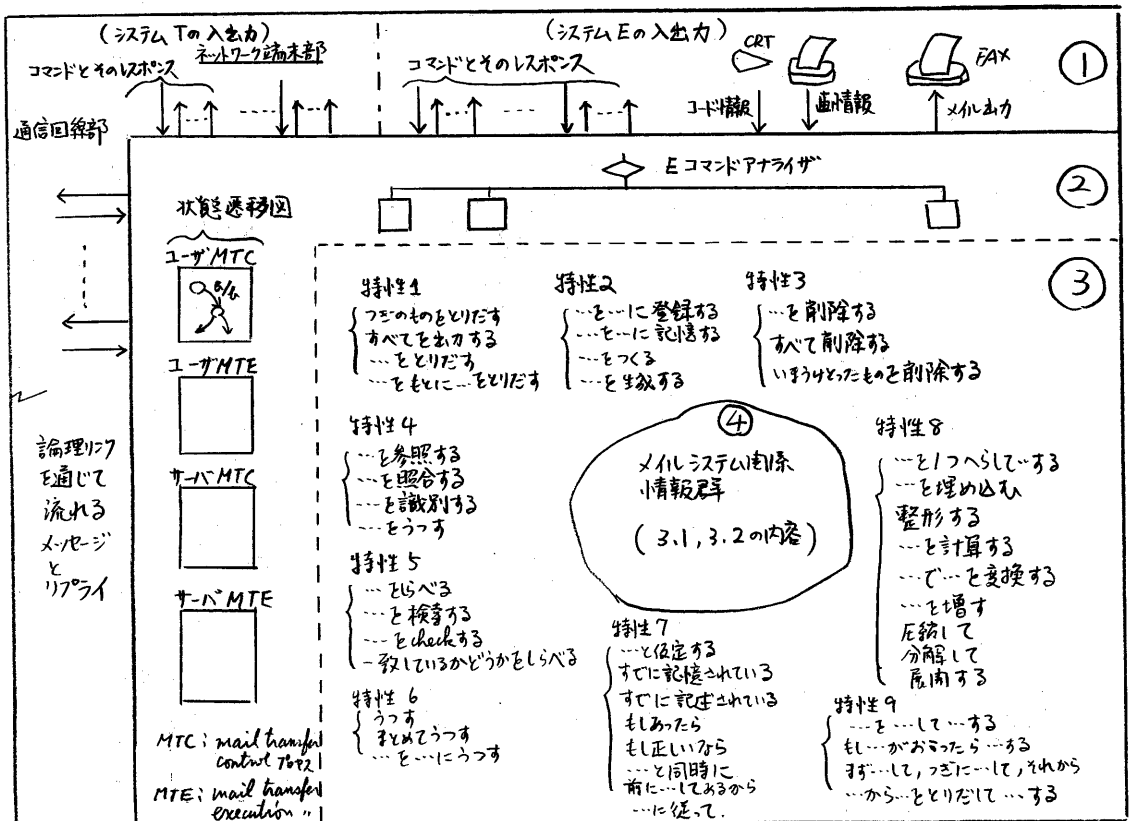


図10 データ型発見に利用したinformalな図

避けるため、文献(11)に従った方法に見き、また、システム機能を十分に理解し、機能に対応する重複したソースコードが散在しないことを目標として、まず、データ型を認識することを目指にした*。図10において、①はシステムの入出力部であり、要求仕様で定められている。②は、システムの動作状況を理解するための手掛りであり、システムEはコマンドアナライザ、システムTはプロトコルプロセスに対する状態遷移図を利用した(コマンドの投入順序に強い制約があるため)。④は、3.1, 3.2で述べた情報群の詳細な記述である。③は、④の情報を利用して、②の処理ロジックを文章表現するときに出現した表現である。特性1~6をデータ型発見の素材に、特性7~9は処理ロジック中の手続きまたは大域データ定義の素材に利用した。

以上の作業は、図1.C部から、作業をはじめるのが、ごく自然であることを意味する。

次に、(1)同じ情報を同じように処理することから生じる、コマンド間の機能の重複をなくす。(2)作成、編集、出力、送受信にともなう、1回の処理単位にあたる情報はまとめる。という方針のもとに、特性1~6の表現を整理した結果、図11に示すデータ型が認識された。結果としてみると、(1)あるセッション開設時に必要とされる主記憶バッファの構造(処理対象メール記述情報)。(2)送信または受信のために保存しておくメールの情報(メール記述情報保存ファイル)および、それらの管理情報(3)コマンド間にあたる情報(例えば処理対象mail-1表)。の3つの主要なデータの分類が認識され、それぞれに、セッション開設時の処理単位に依存したデータ型が認識されたことになる。いくつかのユ

*この過程を形式化する試みはまだ成功していないので、ここでは、図10のようなinformalな形式に従った。

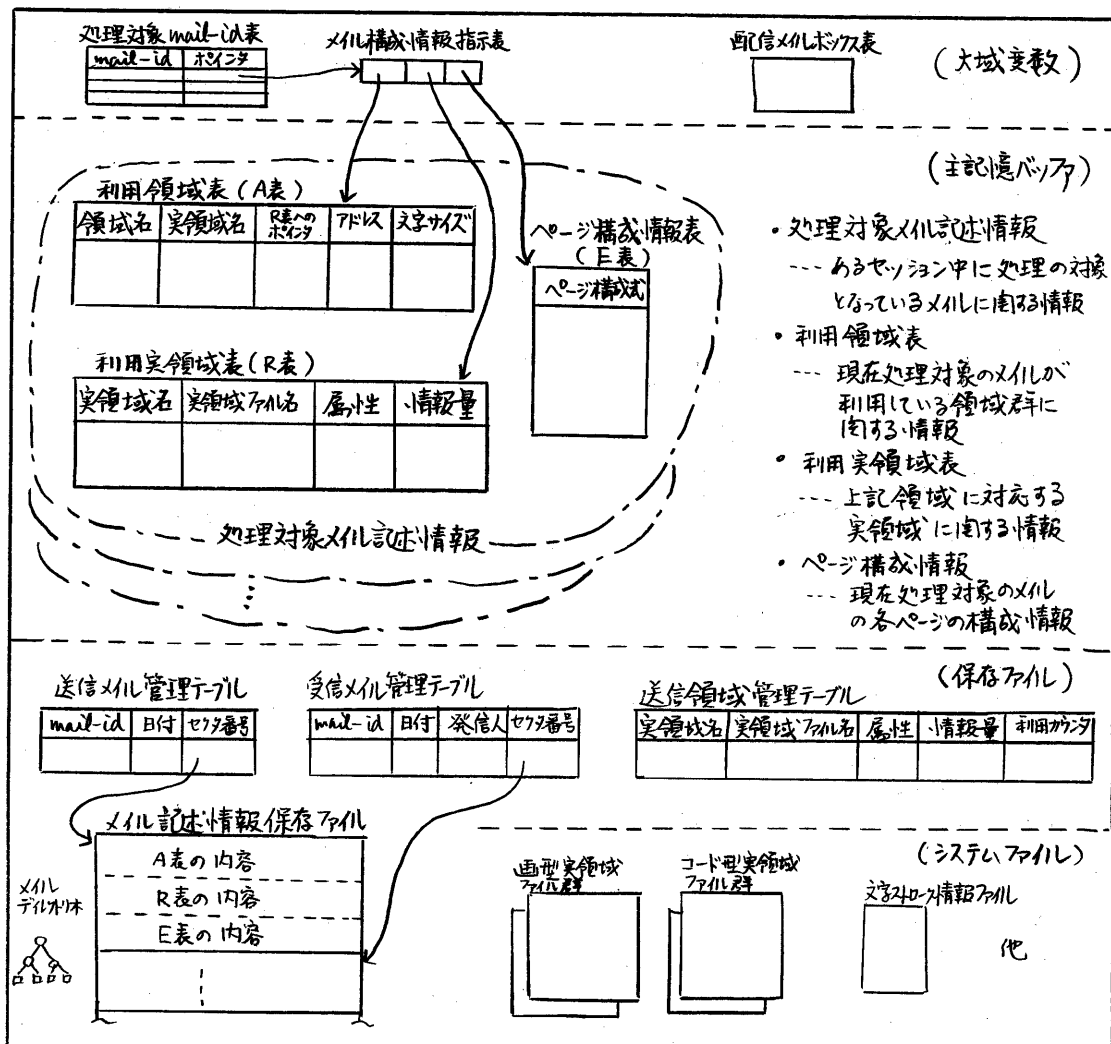


図11 主要データ型一覧

ンドを例にとり説明する。Tuの@DESではA,R,E表をうめ、A,E表を送出する。Tuの@MailではR表と対応する実領域を送出する。Tsの@DESでは、A,E表をうめ、A表をもとにR表の実領域名をうめ。Tsの@Mailでは、R表の残りとうめ、実領域をうめとったのち、処理対象メール記述情報をメール記述情報保存ファイルに登録し、受信メール管理テーブルに情報を追加する。outputsはA,E表をうめ、送信領域管理テーブルを探索してR表をうめたのち出力する。outputs,Rはさらに、処理対象メール記述情報をメール記述情報保存ファイルに登録し、送信メール管理テーブルに情報を追加する。このようにして発見されたデータ型こそ、図1内部でパースモジュール内にいんべいされるべきものである。このときabstract interfaceは図10③の特性1~6を決定したデータ型からなめることで定まる。interface関数のパラメータは次のようにして定めた(図12)。

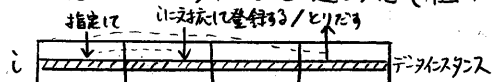


図12 パラメータ設計

すなわち、データインスタンスをキーを利用して指定し、その意を識別子を返してもらうつぎに、その識別子を利用して対応する位置に登録したり、取り出したりする。

最後に、図11の各項目毎にパーナス仕様を作成したが紙面の都合で省略する。

5. 処理部のアルゴリズム設計

ここで、「はじめに」で述べた不明点(2)に対する我々の見解を述べる。パーナスモジュールは、システム機能全体からながめたとき「システム機能に対して最適化した拡張されたファイル」として位置づけるのが最も自然である。abstract interface (サブプログラム) はアクセス法として位置づける。すなわち、図11も部はC部に対する「拡張されたファイルシステム」の役割を果たす。図13で, outputs, outputで利用するメール出力アルゴリズムを例にとり説明する。

メール出力アルゴリズムの設計方針

は以下の通り。

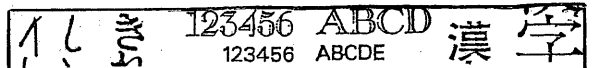


図14 FAX出力の一部

(1) FAX出力のためのFAXバッファ(仮想ページファイル)は、主走査線コード(1680ビット)の順ファイルとして実現し、1コードはディスクの1セクタに保存される。(2)領域は図14の出力紙の部分空間ととめる。(3)ページ構成式とオペラト(領域)を評価おこなうと、領域演算をおこなうとDiskアクセスの効率が非常によくなる。(4)ページ構成式と仮想ページの主走査線毎に1139回評価する。図13において、下線部はabstract interface (サブプログラム名)であり、拡張されたファイルシステムへのアクセスと見なすことができる。

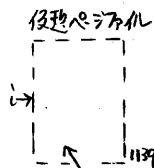
メール出力(Mailout)

for j:=1 to Page do INTPT(j)

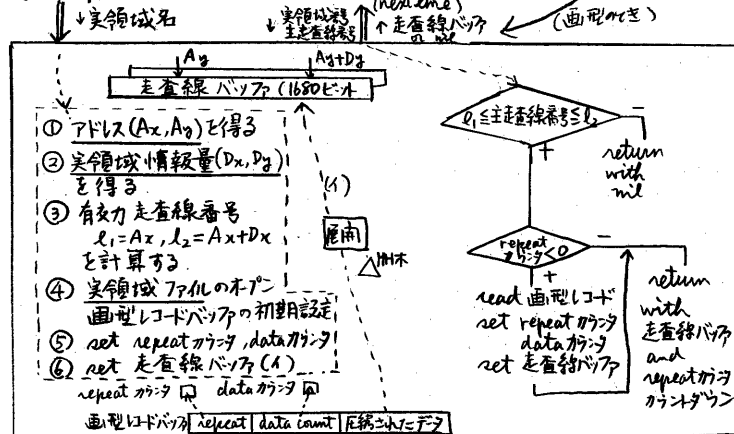
図13 メール出力アルゴリズム(一部)

ページの出力(INTPT(j))

- ① j番目のページ構成式を得る
- ② ページ構成式と逆ポリゴニに変換し、領域名と実領域名、実領域の属性を得、記号表をつくる。
- ③ 記号表をもとに、対応する実領域ファイルをオープンし初期設定(SETBC, SETBG)をする
- ④ for i:=1 to 1139 do (逆ポリゴニ)の評価
(オペラトスタックで結合(a_ib_i*)がおこなわれるとき nextline(i), nextline(i)をおひだし、バッファ上で*の演算をおこなう)
- ⑤ 仮想ページファイルへ1コード追加する



(SETBG)



- ・ SETBG
指定された画型実領域に因に①~④の処理で初期設定する
- ・ NEXLINE
指定された領域の主走査線に対応する評価値を返す
- ・ INTPT(j)
メール中のページjの評価をする
ページ構成式を主走査線毎に1139回評価する

(SETBC)

(nextline)

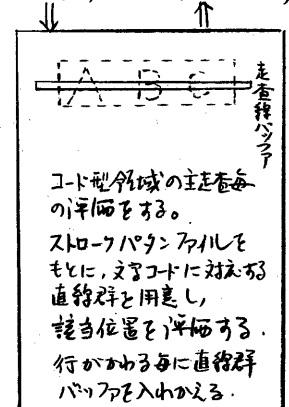


図1にメールシステムのアーキテクチャの概念図を示す。図2におけるシステムDとシステムE間のデータインタフェースであるメールボックス群には、拡張されたファイルシステムとして、パーナスモジュールによって実現されたことになる。また、



においては、データパイプと呼ぶことにする)を状態遷移図の入出力部やデータフロー図のデータ参照の部分に装着することで実現され、別の版は、とりかえ用のデータパイプでおきかえることで実現される。本研究の結果は、現状では一つの事例にもとづくもので、どの程度一般性を持たせうるかは今後の課題である。

文献 (1) D.L. Parnas, "Information Distribution Aspects of Design Methodology", Proc. IFIP Cong. 1971. (2) D.L. Parnas, "A Technique for Software Module Specification with Examples", CACM, vol. 15 No. 5 (May 1972) P.330. (3) D.L. Parnas, "On the Criteria To be Used in Decomposing Systems into Modules", CACM, Vol. 14 No. 12 (Dec. 1972) P. 1053. (4) D.L. Parnas, "On the Design and Development of Program Families", IE³, Trans on SE, Vol. SE-2 No. 1 (March 1976) P. 1. (5) D.L. Parnas, G. Handzel and H. Würges, "Design and Specification of the Minimal Subset of an Operating System Family", IE³ Trans. on SE, Vol SE-2, No. 4 (DEC 1976) P. 301. (6) D.L. Parnas, "Use of Abstract Interfaces in the Development of Software for Embedded Computer Systems", Naval Res. Lab., Washington, DC, NRL Rep. 8047, June 1977. (7) D.L. Parnas, "Designing Software for Ease of Extension and Contraction", IE³, Vol. SE-5 No. 2 (March 1979) P. 128. (8) 菊水 浩一郎, "ソフトウェア工学の技術課題", 昭和58年電気内務学会東海支部連合大会 シンポジウム, P.5-83. (9) E.W. Dijkstra, "Notes On structured Programming", 「Structured Programming」, ACADEMIC PRESS (1972). (10) 高橋 薫, 白鳥 則郎, 野口 正一, "計算機網における画像-文字統合型メールシステムの設計", 電子通信学会電子計算機研究会資料, EC 82-45, 1982年10月. (11) 今泉 恵美子, 菊水 浩一郎, "ソフトウェア開発過程の記録法の一事例", 情報処理学会ソフトウェア工学研究会資料, ソフトウェア工学 32-1, 1983年9月.