

この資料は日本エム・テクノロジー学会員専用です。
この資料を学会員以外がコピーしたり、学会員以外に配布することを禁じます。

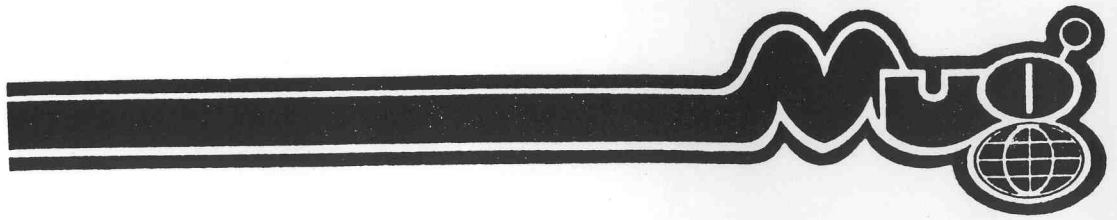
Copy right : M Technology Association - Japan

日本エム・テクノロジー学会事務局
〒259-1193 神奈川県伊勢原市望星台
東海大学医学部・基礎医学系
大櫛陽一

Tel: 0463-93-1121 ext. 2140

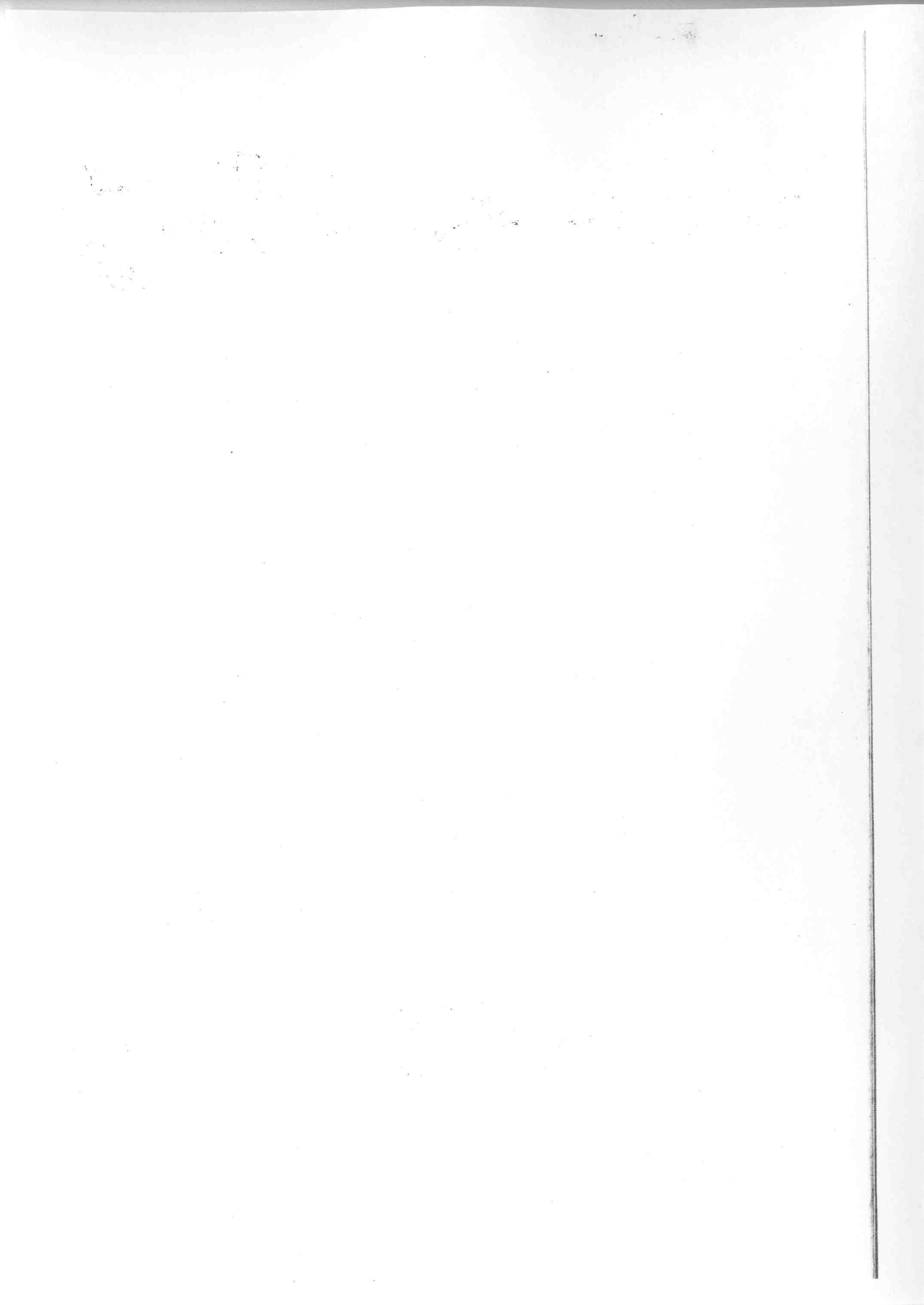
Fax: 0463-96-4301

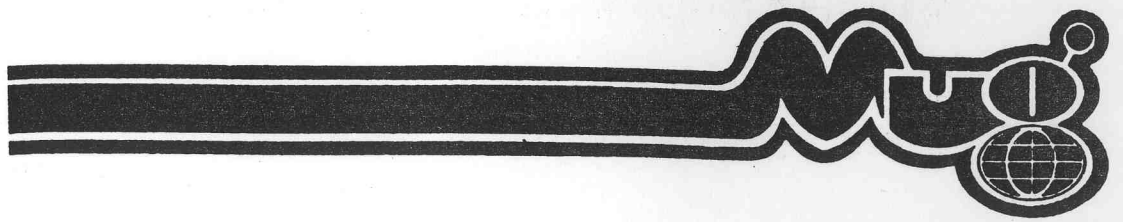
Email: youichi@keyaki.cc.u-tokai.ac.jp



第11回日本MUG学術大会
論文集

1984
名古屋





第11回日本MUG学術大会

プログラム

大会長

大谷元彦

1984年11月24(土)・25(日)・26(月)

名古屋

第11回日本MUG学術大会プログラム日程表

11月24日(土)		11月25日(日)		11月26日(月)	
9:00	マンブス講習会 於：マンブスシステム 研究所	9:00	学術大会 於：愛知県産業貿易館	9:00	学術大会 於：愛知県産業貿易館
	講師：蝶理情報 システム株式会社		開 会		「システムデザイン 2」 座長 河村 徹郎
10:30			「Implementation」 座長 木村 一元	10:30	
10:45				10:45	
	講師：蝶理情報 システム株式会社	11:00			特別講演 大 榎 陽 一 若 井 一 朗 司 会 馬 場 謙 介
12:30	昼 食	11:15	特別講演 R. G. Davis, Ph. D. 司 会 大 谷 元 彦	12:15	昼 食
13:30		12:15	昼 食 運営委員会・評議員会 里 村 洋 一	13:00	総 会
15:00	講師： 住友電気工業株式会社	13:30	「システムデザイン 1」 座長 平川 顕 名	13:30	「ネットワーキング」 座長 里 村 洋 一
15:15		15:00		15:00	
15:00	講師： 住友電気工業株式会社	15:15	特別講演 J. M. Lewkowicz W. R. Houser 司 会 若 井 一 朗	15:15	「アプリケーション」 座長 八日市谷 隆
17:00		17:15		17:00	閉 会
		18:00	懇 親 会 (西館2階 モンブラン)		

会場案内図

マンプスシステム研究所

☎ <052> 936-5660

■交通案内

地下鉄新栄町北 6分

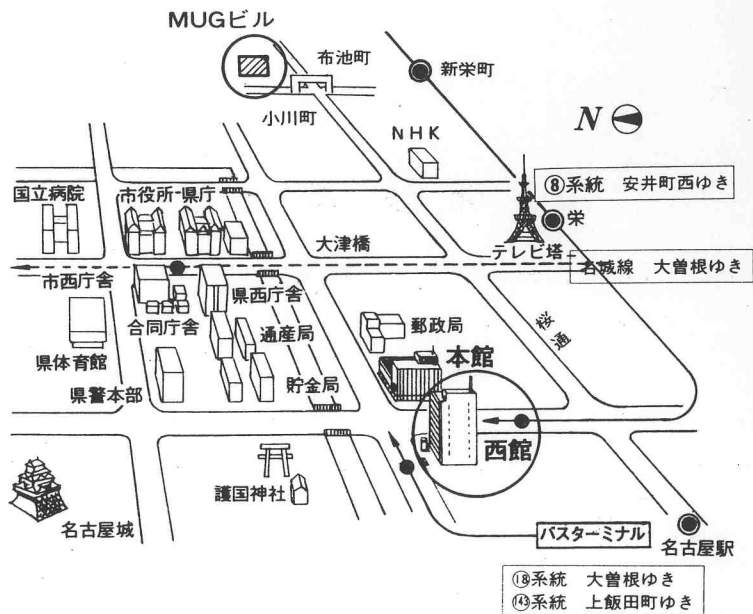
愛知県産業貿易館(西館)

☎ <052> 231-6351

■交通案内

市バス——⑧ ⑬ ⑭ 系統 外堀町通本町下車

地下鉄-----名城線 市役所下車 徒歩10分



大会長より

本年は、メーカー、システムハウス、ベンダー、メディカル／ノンメディカルユーザー、研究者の方々と一緒に、マイクロMUMPSを含むハード／ソフト環境、アプリケーション、ソフト交換、開発と導入、日本語MUMPS、知識ベース、ネットワーキング、海外からの新知識、将来展望など、ゆったりとした会場で充分話し合い、知識の交換と身になる勉強が出来るように計画しています。多数の御参加を希望します。

会 期：11月24日(土)、25日(日)、26日(月)

会 場：マンブスシステム研究所
愛知県産業貿易館 (左頁 案内図参照)

参加者各位

(11月15日以降の登録者は、当日参加費をいただきます。)

	会 員	一 般
1. 「MUMPS講習会」	5,000円	8,000円
——マイコンによる日本語MUMPS 初級より応用まで——	(当日参加費)(7,000円)	(10,000円)
2. 大会参加費	3,000円	5,000円
	(当日参加費)(4,000円)	(7,000円)
3. 論文集代	約 2,000円	約 4,000円
4. 懇 親 会	3,500円	3,500円

発表者各位

- 発表時間は一題につき発表10分間、討論5分間です。但し、セッションの最初には、セッション基調を置き、時間を延長します。
各セッションに与えられた時間を参考にして下さい。
- スライドを使用される場合は該当セッション開始30分前までにスライド受付にご提出下さい。OHPを使用される場合は、そのスライド受付にご連絡下さい。また発表者到着確認のため必ず会場受付にお立寄り下さい。

連絡先

第11回日本MUG学術大会事務局

〒470-11 愛知県豊明市沓掛町田楽ケ窪1-98
藤田学園保健衛生大学医学部衛生学教室
TEL 0562-93-2456

第11回日本MUG学術大会プログラム

第1日目 11月24日(土)

於：マンブスシステム研究所

***講習会 (MUMPS講習会) AM9:00~PM5:00**

——マイコンによる日本語MUMPS 初級より応用まで——

第2日目 11月25日(日)

於：愛知県産業貿易館西館10階大会議室

受付開始 8:30~

***開 会 AM9:00~**

***Implementation AM9:00~11:00**

座長 木村 一元 (独協医大)

- | | |
|--|-----------------------------|
| 1 MUMPS環境の現在の進歩.....John J. Althouse (イギリスMUG技術ディレクター) | |
| (9:00~9:45) | |
| 2 コンパイラ型MUMPSによるパフォーマンスの向上 | 1 |
| | 石原 哲ら (三井造船システム) |
| 3 ALEF MUMPSの特徴 | 鈴木利明 (アレフコンピュータ) 3 |
| 4 U-MUMPS | 久江 正ら (住友電工・ME) 7 |
| 5 USTATIONのMUMPSの開発 | 小林 勝ら (住友電工・ME) 11 |
| 6 KANJI MUMPS — RESOLVED AND UNRESOLVED ISSUES — | 15 |
| | D. A. Smith ら (マンブスシステム研究所) |

***特別講演 AM11:15~12:15**

座長 大谷元彦 (藤田学園医学部)

V. A. ファイルマネージャーの利用法

R. G. Davis, Ph. D. (レキシントンV. A. 医療センター)

(昼休憩 PM12:15~13:30)

運営委員会・評議員会 里村洋一

第11回日本MUG学術大会プログラム

*システムデザイン 1 PM13:30~15:00

座長 平川 顕 名 (京大病院)

- | | |
|--|----|
| 1 MUMPSによる言語と知識ベースの設計 (13:30~14:00) | 19 |
| 里村 洋一 (千葉大・医) | |
| 2 MUMPSによる集計ルーチンとファイルエディター | 23 |
| 佐藤 甫夫ら (千葉大・医) | |
| 3 剖検輯報データベースを例として——データチェックにおける日本語処理の効用—— | 29 |
| 福永 泰明ら (産業医大) | |
| 4 研究者用患者データ管理システム(CUIPDS)における時系列データ処理 | 39 |
| 本多 正幸ら (千葉大・医) | |
| 5 MUMPSよりのVMSおよびDSM資源の利用 | 43 |
| 田久 浩志ら (産業医大) | |

*特別講演 PM15:15~17:15

座長 若井 一朗 (マンブスシステム研究所)

DEVELOPMENT OF KNOWLEDGE BASE AND KNOWLEDGE NETWORKS IN MUMPS (15:15~16:00)

John M. Lewkowicz
(ニューヨーク州 コーネル大学)

なぜアングラMUMPSが連邦機関の中核となったか (16:00~17:15)

W. R. Houser
(合衆国一般公務管理局情報資源企画部顧客調整官)

*懇親会 PM18:00~

於：愛知県産業貿易館西館2階 モンブラン

第11回日本MUG学術大会プログラム

第3日目 11月26日(月)

於：愛知県産業貿易館西館10階大会議室

*システムデザイン 2 AM9:00~10:30

座長 河村 徹郎 (大阪成人病センター)

- | | | |
|--|--------------------------|----|
| 1 SMALL CONCEPT (9:00~9:30) | 馬場 謙介ら (産業医大) | 45 |
| 2 大量データのバッチ処理 | 田久 浩志ら (産業医大) | 51 |
| 3 VAX-DSMへ漢字システムとグラフィックシステムの統合化 | 馬場 勝美ら (日本専売公社) | 59 |
| 4 MUMPS Color Graphics and its Enhancement by Zcall for Mouse
Functionality | D. A. Smithら (マンブスシステム研) | 63 |
| 5 Micro MUMPSの使用経験 | 木村 一元ら (独協医大) | 71 |

*特別講演 AM10:45~12:15

司会 馬場 謙介 (産業医大)

日本語MUMPS

大 榎 陽 一 (大阪府立羽曳野病院)

Micro MUMPS

若 井 一 朗 (マンブスシステム研)

(昼 食 PM12:00~13:00)

*総 会 PM13:00~13:30

会長 里村 洋一 (千葉大・医)

第11回日本MUG学術大会プログラム

*ネットワーク PM13:30~15:00

座長 里村 洋一 (千葉大・医)

- | | | |
|-------------------------------------|--|----|
| 1 異種MUMPSシステム間のネットワーク (13:30~14:15) | _____ | 75 |
| | R. F. Walters, Ph. D. (カリフォルニア大学工学部教授) | |
| 2 藤田学園保健衛生大学病院におけるMUMPS使用経験 | _____ | 81 |
| | 山内 寛ら (藤田学園コンピュータ管理センター) | |
| 3 病院情報システムによるLANの利用 | 田川 順通ら (三井造船・システム) | 85 |
| 4 LANを用いた病院情報システム | 河村 徹郎ら (大阪成人病センター) | |

(休 憩 PM15:00~15:15)

*アプリケーション PM15:15~17:00

座長 八日市谷 隆 (東北大・医)

- | | | |
|---------------------------------------|---------------------------|-----|
| 1 複合設計開発における品質管理 (15:15~15:45) | 今泉 幸雄 (日本アップジョン) | 91 |
| 2 健康予測システム | 柿崎 賢一ら (高岳製作所) | 95 |
| 3 藤田学園保健衛生大学病院における検査システムの特徴 | _____ | 101 |
| | 落合 潤一ら (藤田学園コンピュータ管理センター) | |
| 4 藤田学園保健衛生大学病院におけるMUMPS薬剤システムの特徴 | _____ | 105 |
| | 竹林 和行ら (藤田学園コンピュータ管理センター) | |
| 5 藤田学園保健衛生大学病院における病棟看護業務とMUMPSシステムの現状 | _____ | 109 |
| | 蓑原美奈恵ら (藤田学園コンピュータ管理センター) | |
| 6 MUMPS POETRY | D. A. Smithら (マンブスシステム研) | 113 |

*閉 会 PM17:00

- 75

- 81

「Implementation」

- 85

座長 木村一元 (獨協医大)

- 91

- 95

-101

ター)

-105

ター)

-109

ター)

-113

1984年度 第11回 日本マンブスユーザーズグループ学術大会 「Implementation」 ②

コンパイラ型MUMPSによるパフォーマンスの向上

滝沢正隆, 吉田欣弥, ○石原 哲

三井造船株式会社 システム事業本部

システムエンジニアリング事業部 電子機器技術部

現在用いられているMUMPS言語は、インタプリティブでありマンマシン言語としては非常に使い易くプログラム開発上の効果は大きい。しかし、インタプリティブであるが故に、命令の実行時間の点で考えると、コンパイラ型言語に比べると、どうしても遅くバッチ処理的な業務には不利であつた。

米因インターシステムズ社のVAX/VMSのもとで稼動するコンパイラ型MUMPS(商標名:M/VX)のベンチマークを行つたところ、アプリケーションの処理時間において従来比で10倍以上の高速性が実証された。現在ではVAXにつづいてPDPでも利用でき、M/11+としてリリースされている。

1. コンパイラ型MUMPS

M/VX(あるいはM/11+)は従来のインタープリンター型の利点を生かしたまま、ソースプログラムの入力のとおり一行入力することに自動コンパイルが行われる。個々のコマンドまたはコマンドと組み合わされた関数の実行時間は2~10倍向上している。このコンパイル方式では、プログラマはあくまでもインタープリタと何ら変わらないコーディングが行える。又コンパイルの待ち時間も無い。さらにソースコードを消去して残つたオブジェクトコードに対してソースコードと同じ要領でエディットすることもできる。

• M/VX

VAX/VMSオペレーティングシステムのもとで動き、MUMPS言語プロセッサ、データベース管理、ユーティリティから成っている。このM/VXは32ビットパーチャルメモリを搭載したVAXコンピュータの能力を十分に引き出すように設計されており、どのVAXへもインストールできる。その特徴は、

- ① 従来のインタープリンター型に比べて2倍以上のスループットの向上
- ② 同様に2倍以上のオンラインユーザ数が実現されていること
- ③ 1983年の拡張機能を含むANSI標準MUMPSであること
- ④ VAX/VMSのもとで他言語と同時使用が可能であること
- ⑤ RMSファイル、メールボックスを通じ他言語系とのデータ転送が可能であること
- ⑥ ASCII, EBCDICの各種の磁気テープフォーマットが使用可能であること
- ⑦ VAX, PDPのいずれともMUMPS間の直接ネットワーク(DDP)が可能であること
- ⑧ インストレーションは30分以内で可能であること
- ⑨ VAX/VMSがサポートするすべてのハードウェア構成で使用可能であること
- ⑩ RMSを使わない独自のデータベース管理がされていること

● M/11+

M/11+はPDP-11シリーズのための専用オペレーティングシステム，MUMPS言語プロセッサ，データベース管理，ユーティリティから成っている。

その特徴は、

- ① 従来のインタープリタ型に比べて2倍以上のスループットの向上
- ② 3MBのメインメモリで63ユーザ，2MBで56ユーザまでのオンライン処理が可能なこと
- ③ 512KB～4MBのメモリをサポートしていること
- ④ 最大100個の共通ルーチンバッファの動的割付によりオーバーヘッドが減少すること
- ⑤ 最大1MBのディスクキャッシュが備わっていること
- ⑥ 2KBのディスクブロックサイズが備わっていること
- ⑦ 1983年の拡張機能を含むこと
- ⑧ ASCII，EBCDICの各種磁気テープフォーマットを持つこと
- ⑨ VAX，PDPのいずれとも分散型データベース処理が可能であること
- ⑩ インストールが容易であること

2. パフォーマンスの改善

VAX-11/750を用い当社のコンテナ自動化ターミナルのアプリケーション (Road Receiving Reception) をもとに処理時間のシミュレーションを行った結果をFig 1.に示す。10トランザクションあたりのレスポンスタイムを計測した。実際のオペレーションにおいて待ち時間の差異が10倍位生じると考えることができる。

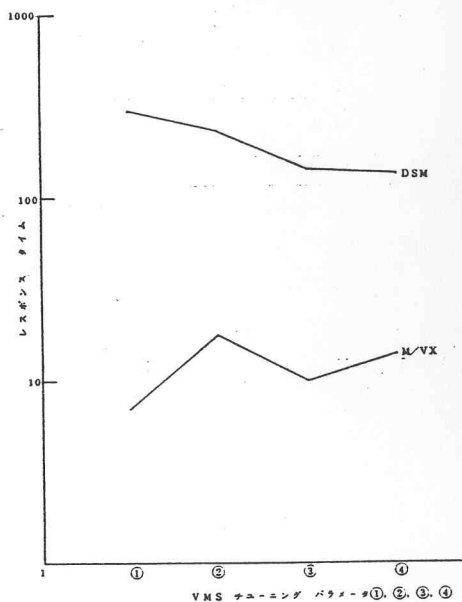
3. 将来の方向

現在はカナまでしかハンドリングされていないが、近く漢字も使用可能となるであろう。

参考資料：新型MUMPSのご紹介 (三井造船)

Fig 1. コンテナ自動化ターミナル Application Program (Road Receiving Reception-) を使用し同時に8台の端末で処理した場合の比較

VMSC ユーニング パラメータ	①	②	③	④	備 考
PRIORITY	4	4	12	12	ユーザーの優先度
WSQUOTA	250	750	750	750	ワーキングセット拡張サイズ
WSEXTENT	300	900	900	900	ローン領域サイズ
QUANTUM	20	20	20	4	プロセス時間



1984年度 第11回 日本マンブスユーザーズグループ学術大会
「Implementation」 ③

ALEF MUMPS の 特徴

○ 鈴木 利明

有限会社 アレフ コンピュータ
440 豊橋市 花田三番町94

ALEF MUMPS バージョンA-0.9とJ-0.9の開発

A. 当社では、バージョンA-0.9とJ-0.9のMUMPSを開発しました。
A-0.9はASCII仕様で、J-0.9は日本語仕様である。
このふたつは共に、ANSI'83標準MUMPSをマルチJOB以外のすべてを満足し、ディスクの識別はボリュームネームとファイルネームを用いている。
変数名（ローカル、グローバル）の全長は、255バイトまで使用可能です。

B. 仕様概要は次の様である。

- 1 機種 現在、動いている機種はNEC 9800, 富士通FM-11, シャープMZ-5500です。
NEC 9800, シャープ MZ-5500のプロセッサはインテル8086, 富士通 FM-11は、モトローラ6809である。
- 2 メモリー メモリーの配置は、ローカル変数エリア8KB, スタックエリア4KB, ルーチンエリア8KBである。その他のフリーエリアはすべてディスクキャッシュエリアとして使用している。
- 3 ディスク 5インチフロッピー, 8インチフロッピーディスクが使用できる。
ウインチェスターディスク仕様は開発中です。
- 4 OS OSは、U-systemを使用している。
- 5 JOB A-0.9, J-0.9は共にシングルユーザ仕様である。
\$JOB特殊変数は、常に0を返す。
JOBコマンドは実行されない。
LOCKコマンドは実行されない。

- 6 文字 バージョンA-0.9はASCIIのみである。バージョンJ-0.9はASCIIとJISC6226（日本語）の混在ができる。
- 7 ルーチン名 ルーチン名に使用出来る文字は、ASCIIとC6226である。
ルーチン名の識別は、32バイトまでである。
- 8 ラベル ラベルに使用出来る文字は、ASCIIとC6226である。
ラベルは、記述してあるすべての文字を識別する。
- 9 ローカル変数 ローカル変数名に使用出来る文字はASCIIとC6226である。
識別は63バイトまでである。添字はASCIIとC6226が使用出来る。
識別は63バイトまでである。数値表現可能数値すべてが添字として使用できる。
全長制限（ローカル変数名、全添字、各種制御コードを合わせたもの）は255バイト以内である。
- 10 グローバル変数 グローバル変数名に使用出来る文字はASCIIとC6226である。
識別は32バイトまでである。添字はASCIIとC6226が使用出来る。
識別は63バイトまでである。数値表現可能数値すべてが添字として使用出来る。
全長制限（グローバル変数名を除いた全添字、各種制御コードを合わせたもの）は255バイト以内である。
- 11 数値 MUMPS内部において数値は、10進浮動小数点表現を使用し、仮数の有効桁数は中間演算終了時に22桁、変数代入時に18桁に丸める。指数は、10E-99から10E99まで表現できる。
- 12 追加システム変数
- \$ZROUTINE ルーチンボリュームネームとルーチンファイルネームを値として持つ。読み書き可能である。
- \$ZGLOBAL グローバルボリュームネームとグローバルファイルネームを値として持つ。読み書き可能である。
- 13 追加コマンド
- ZDELETE 指定したルーチンを消去する。
- ZEDIT 指定した範囲の行を修正する。
- ZINSERT 指定した行の前から行を挿入する。
- ZLOAD 指定したルーチンをルーチンエリアに読み込む。
- ZMOVE 行ポインタの移動。
- ZPRINT 指定した範囲の行を表示する。
- ZREMOVE 指定した範囲の行を消去する。
- 14 追加演算子
- ** 演算子 # の定義に合わせた整数除算演算子である。
 $N=N*(N**M)+(N\#M)$ が演算精度内において常に成立する。
- <> <> と '=' は 同値演算子である。
- >= >= と '<' は 同値演算子である。
- <= <= と '>' は 同値演算子である。

C. ボリュームネーム、ファイルネームの識別

これまでのシステムでは、フロッピーディスクの識別はドライブ名で行われてきた。例えば、"ドライブA"、"ドライブ1"というふうに使ってきた。そのためディスクを差し間違えて内容を壊すことがある。

これは指定したドライブに入っているディスクを確認しないで処理をするからである。当システムでは、ディスクの確認をボリュームネームとファイルネームの識別で行なうことによりディスクの破壊を防いでいる。

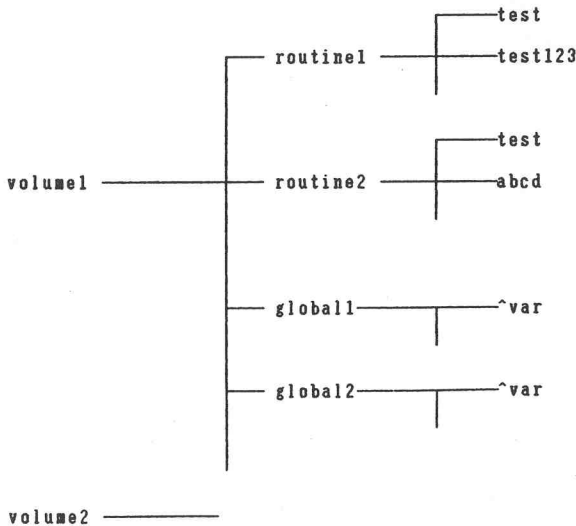
このために当システムでは、ルーチンにはシステム変数\$ZROUTINE、グローバルには、システム変数\$ZGLOBALを新設した。ファイルのアクセスは、\$ZROUTINE、\$ZGLOBALで指定したボリュームネームのディスクにあるファイルネームのファイルに対して行なわれる。

この方法ではドライブの指定は必要なく、指定した求めるボリュームネームのディスクがどのドライブにあるかは、システムが自動的におこないます。指定したボリューム（ディスク）がない時には、システムはユーザに対しボリュームネームを表示して、ディスクがセットされるまで待ちます。これは、これまでのシステムは出来なかったことです。

例

```
SET $ZROUTINE="routine1^volume1",$ZGLOBAL="global1^volume1"
DO test WRITE ^var
SET $ZROUTINE="routine2^volume1",$ZGLOBAL="global2^volume1"
DO test WRITE ^var
```

ボリュームネーム | ファイルネーム | ルーチンネーム、グローバルネーム



D. プログラムの開発、デバッグ

1 VIEWコマンド

VIEW 0, VIEW 1, WRITE 引数なしは、ローカル変数名 (添字付) と値を表示する。

VIEW 2は、定義してあるグローバル変数名をリスト表示する。

VIEW 3は、定義してあるルーチン名をリスト表示する。

VIEW 引数なしは、MAC (機械語プログラム) をコールする。

MACのサブコマンド

M: メモリー内容の16進表示。

D: メモリーの逆アセンブラ表示。

F: ディスクの物理的入出力。

など。

2 BREAKコマンド

システムは、BREAKコマンドを実行するとMUMPSプログラムの実行を中断して直接モードに入る。再開は、直接モードからZGOコマンドで行なう。

BREAKコマンドのネスティングは10重まで使用出来る。

3 エラー表示

MUMPSプログラム実行中にエラーを起こすと、システムはスタックエリアの内容を表示し、次にエラーメッセージを表示する。

スタックエリアは、FOR, DO, EXECUTE, 間接ネーム、間接引数を使用した時にスタックされたいろいろなパラメータを保持している。

表示されたスタックの内容により、ユーザーはエラーを起こした正確な位置とそれまでの経過を知ることが出来る。

4 エディタ

ユーザはMUMPS用のエディタをコールすることができる。

エディタのサブコマンドには、サーチ、チェンジなどがある。

E. ディスクフォーマットの統一

互換性のあるディスクメディアでは、そのフォーマットは統一してある。これにより異なる機種間でも、差し替えが可能である。

U-MUMPS

小林 勝、久江 正、上戸 隆

住友電気工業株式会社 ME開発室

〒554 大阪市 此花区 島屋1丁目 1番 3号

著者らは、住友電工が開発した、68000マイクロプロセッサをベースとするワークステーションUSTATION上に、MUMPS（商品名：U-MUMPS）を装備した。U-MUMPSは、(1)ANSI標準およびtype Aに準拠し、かつ日本語処理もサポートしている。(2)UNIXOSというマルチ言語環境下において稼動する。(3)ブリコンパイル方式等の採用により、処理速度の向上を図っている。(4)プログラム開発保守・システム運用管理に必要なユーティリティライブラリを持つ。

1. はじめに

本セッションでは、著者らがワークステーションUSTATION¹⁾上に装備を行ったU-MUMPSの特徴および性能について述べ、いくつかのデモプログラムを用いてU-MUMPSを紹介する。

2. U-MUMPSの特徴

U-MUMPSの特徴を下記に示す。

(1)ANSI標準(XI I. 1)およびその拡張仕様type A²⁾に準拠するとともに表1に示すコマンド・関数を拡張している。

(2)UNIXオペレーティングシステム的环境下で、シングルユーザーにおいてもマルチユーザーにおいても実行が可能であり、UNIXがもつ豊富なユーティリティが活用できる。また、ホストファイルサーバ(HFS)機能を用いることにより、他の言語プログラムとの同期通信が可能である。

表1 U-MUMPS拡張コマンド・関数一覧

項目	内容
ZALLOCATE	ローカルまたはグローバル変数アクセスを占有する
ZDEALLOCATE	ZALLOCATEで占有された変数を解放する
ZGO	BREAK後のプログラム継続実行
ZINSERT	ユーザプログラム修正の際のラインの追加挿入
ZJOB	指定プログラムの別パーティションでの起動
ZLOAD	プログラムをユーザパーティションへロードする
ZPRINT	ユーザパーティションのプログラムダンプ
ZREMOVE	プログラム一部又は全部の抹消
ZSAVE	プログラムのディスクへの格納(ブリコンパイルの実行)
ZTRACE	プログラム内またはプログラム間のプログラム実行トレース
ZWRITE	ユーザパーティションのローカルデータダンプ
\$ZBOOLEAN	任意データのビットマスク計算
\$ZCRC	任意データのチェックサム計算
\$ZDATE	内部日付表現を外部日付フォーマットに変換
\$ZHEX	16進数表現と8進数表現の変換
\$ZNEXT	ローカルまたはグローバル変数の指定された次の添付のデータ参照
\$ZVERIFY	ディスクの指定ブロックの診断
\$ZWIDTH	漢字を含むデータの出力時の印字幅計算

- (3) システムの稼動状況、データベースの整合性のチェック、エディタ等、MUMPSの稼動およびプログラムの開発を支援する約150本のルーチンから構成されるユーティリティライブラリを装備している。表2-1、表2-2に主要ユーティティ一覧を示す。
- (4) 言語処理系は、言語コンパイラと疑似コードインタプリタより構成され、プリコンパイル方式の採用により、実行時の性能を向上させている。
- (5) パーティションのサイズは、20KBを標準に、メモリの制限範囲内で、実行時にダイナミックに変更できるダイナミックパーティションである。また、同一ルーチンを複数のJOBが使用する場合、各ジョブでルーチンイメージを持たず、共有する(シェアードルーチン方式)ため、プログラムロードのためのオーバヘッド時間を短縮できるとともに、メモリ領域を有効に活用できる。
- (6) データ、コメント、ラベル、変数、プログラム名を、カナ、漢字にまで拡張し、カナ漢字変換等のユーティリティも装備しており、充実した日本語プログラミングが可能である。
- (7) ジャーナリング機能、JOB間通信、UNIXファイル間通信、バックグラウンドジョブの起動・停止、ファイル保護、システムのバックアップ等、システムの運用に不可欠な機能を有している。

表2-1 U-MUMPS主要ユーティティ一覧(II)

本特徴で、述べたようにU-MUMPSは、プログラミング環境、データベースの構築・システムの運用面において、実用的なシステムを提供している。

3. U-MUMPSの処理速度

表3に、社内において行った、シングルユーザにおけるベンチマークテストの結果を示す。U STAT IONには、CPUに68000(クロック10MHz)を採用したU STAT ION(E10)、内部クロックを12.5MHzにし、キャッシュメモリの付加、ディスクアクセスの高速化を行ったU STAT ION(E15)の2機種が、現在

分類	プログラム名	内 容
プログラム開発・保守	%	ルーチンエディタ
	% FL	現UCIのルーチン一覧(各ルーチンの1行目出力)
	% INDEX	ルーチン間のクロス・リファレンスリスト出力
	% INDSTR	ルーチンの構造化リストイング
	% NEWD	日付を指定し、その間で更新したルーチン一覧を出力
	% RCHANG	シンタックス更新部分を指定し、該当シンタックスを含む複数ルーチンの一斉変換
	% RCOPY	UCI間でのルーチンコピー
	% RD	現UCIのルーチン一覧
	% RDEL	現UCIの複数ルーチンの抹消
	% RPRT	ルーチンリスト出力
	% RR	外部デバイスからのルーチンのリストア
	% RS	外部デバイスへのルーチンセーブ
	% RSE	シンタックスを指定し、該当シンタックスを含むルーチンサーチ
	ZEDITD	エディタ使用法の出力
グローバルデータ処理	% GCH	データ保護モード等グローバル変数の属性変更
	% GCHANG	データ変更部分を指定し、該当データを含むグローバルデータを一斉変更
	% GD	現UCIのグローバル変数名一覧
	% GDEL	現UCIの複数グローバル変数の抹消
	% GEDIT	グローバルデータエディタ
	% GL	グローバルデータリストイング
	% GR	外部デバイスからのグローバルデータのリストア
	% GS	外部デバイスへのグローバルデータのセーブ
	% GSEL	データストリングを指定し該当データを含むグローバルデータをサーチ
	GLBPLACE	ディスク上でのグローバル変数の生成領域の指定
システム管理	% ACTJOB	稼動中の全てのジョブ№を返す
	% DEVUSE	稼動中の全てのジョブと所有しているデバイス№の一覧出力
	% ECHO	ターミナルに対して、エコー/ノンエコーのコントロールを行なう
	% GUCI	現在のUCI名を得る
	% LOGON	UCIの変更
	% PARTSIZ	稼動中のパーティションサイズをダイナミックに変更する
% MDMP	指定ロケーションのメモリダンプおよびVIEWバッファダンプ	

日本マンプスユーザーズグループ

ある。

MUMPSの代表的なコマンドに対して、実施した結果、社内DEC社PDP11-34 (DSM Version 2.0) に比して、メモリ内オペレーションに対して、USTATION (E10) では、4~6倍、USTATION (E15) では、7~11倍、ディスクオペレーションに対しては、USTATION (E10) では、1~1.5倍、USTATION (E15) では、2~3倍の性能を示している。

4 おわりに

著者らは、汎用OSであるUNIXのもとで稼動するUSTATION上に、従来のMUMPSに比して、飛躍的に実行速度が向上したU-MUMPSを装備した。

また、本論文で述べたように、運用面に関しても、実用的な機能を装備しており経済的なシステム構成を提供し得る。

また、USTATIONのシリーズ化と共に、E10、E15の上位機種であるE20、下位機種であるE5へのU-MUMPSの装備を順次行う予定である。

また、U-MUMPSのアプリケーションとしては従来の医事会計、臨床検査等のメデイカル分野のみならず、LANを用いた電子メール、ワープロ、データベース検索システム等のオフィスオートメーション分野への拡大も今後期待される。

表 2-2 U-MUMPS主要ユーティリティー一覧(2)

種類	プログラム名	内 容
システム管理	% SBP	シーケンシャルディスクプロセッサに対する現在のステータス、ブロック№、バッファオフセットを表示
	% SDEV	指定されたデバイスに対する、オープンパラメータを指定する
	% SP	ディスクの空きエリアの表示
	% SS	ジョブ稼動状況の表示
	BLKDMP	指定のディスクブロックダンプ
	DBFIX	ディスクデータ回復のため指定ディスクブロックの書き換え
	DISKMAP	マップブロックのダンプ (各マップ毎空きスペースを表示)
	JRNAL	ジャーナルエリアのアロケーション・イニシャライズ、ジャーナルのスタート・ストップ指示、ジャーナル稼動状況の表示およびジャーナルからのデータ復旧を行なう
	KILLJOB	指定ジョブの強制終了
	LOCKTAB	ロックテーブルの表示
	MAPBP	ディスクバッファの表示
	MAPDDB	デバイスディスクリプトブロックの表示
	MSMDR	MUMPSデータベースのバックアップコピー
	SBP	シーケンシャルディスクスペースのアロケーションおよびデアロケーション
	SSD	MUMPSシステムの停止
	STU	MUMPSシステム立ち上げ (MUMPS起動により自動的に呼ばれる)
その他	SYSGEN	MUMPSシステムジェネレーション
	UCIMGR	UCIの生成・変更および一覧出力
	VALIDATE	ディスクデータの診断
	VERIFY	マップブロックの修復
	% D	システム日付の表示
	% DH, % HD	10進表現と16進表現の変換
	% DI, % DO	日付内部表現と外部表現の変換
	% HELP	ユーティリティプログラムについての説明出力
	% SQRT	ルート計算
	% T	システム時刻の表示
% TI, % TO	時刻内部表現と外部表現の変換	

表3 ベンチマークテスト結果(社内比較)

分類	テ ス ト 項 目	DSM-11(V20)		VMS (V20)	U-MUMPS	
		PDP 11/34	PDP 11/44	VAX780	E10	E15
ローカルデータ処理	F I=1:1:50000	19.1	8.5	8.5	2.1	1
	F I=1:1:5000 S A=I	11.9	6.0	-	2.0	1
	F I=1:1:5000 S A="1234567890"	12.7	5.1	-	1.7	1
	F I=1:1:100 F J=1:1:50 S A(J)=J	5.1	2.4	-	2.2	1
	(F I=1:1:50 S A(I)=I) F I=1:1:100 F J=1:1:50 S A=#N(A(J))	20.5	9.2	-	2.0	1
	(F I=1:1:50 S A(I)=I) F I=1:1:100 F J=1:1:50 S A=#D(A(J))	4.4	2.0	-	2.0	1
演算子処理	F I=1:1:100 F J=1:1:50 S A(J)="1234567890"	4.9	2.3	-	2.0	1
	F I=1:1:5000 S A=I+I-1	12.4	4.8	1.6	2.0	1
	F I=1:1:5000 S A=I*I/I	7.5	3.6	1.3	2.0	1
	F I=1:1:5000 S A=I#100#10	12.5	5.8	-	2.1	1
	F I=1:1:5000 S A="1234567890"_"67890"	6.8	2.7	0.9	1.8	1
	F I=1:1:50 F J=1:1:100 S A=I>J	11.6	4.6	-	1.7	1
関数処理	(S X=1234567890, Y=X) F I=1:1:5000 S A=X=Y	11.1	4.9	-	1.9	1
	(S X=1234567890, Y="A") F I=1:1:5000 S A=X(Y)	11.4	3.8	-	1.9	1
	(S X=0123456789, Y="0123456789A") F I=1:1:5000 S A=X(Y)	12.6	3.8	-	2.2	1
	(S X="853.03.05") F I=1:1:5000 S A=X?1A2N1", "2N1", "2N	12.5	4.2	-	1.9	1
	F I=1:1:5000 S A=#A("A")	10.4	4.8	1.6	1.6	1
	F I=1:1:5000 S A=#C("65")	9.2	3.9	2.6	2.0	1
グローバルデータ処理	F I=1:1:5000 S A=#E("12345", 2, 3)	11.9	4.4	1.9	1.9	1
	F I=1:1:5000 S A=#F("1234567890", "6")	13.0	3.9	1.9	1.9	1
	F I=1:1:5000 S A=#J("12345.7890", 10, 1)	2.4	1.1	1.6	1.7	1
	F I=1:1:5000 S A=#L("12345")	11.3	4.8	1.9	1.6	1
	F I=1:1:5000 S A=#P("1/2/3/4", "/", 2, 3)	10.9	4.3	-	1.9	1
	F I=1:1:5000 S A=#T(A)	4.2	1.6	-	1.8	1
グローバルデータ処理	(S X="1234567890", L1=1000, Y=5) F I=1:1:Y F J=1:1:L1 S ^A(J)=X	2.1	1.0	4.3	2.1	1
	(S X="1234567890", L1=1000, Y=5) F I=1:1:Y F J=1:1:L1 S A=#A(J)	2.0	1.0	4.0	2.0	1
	(S X="1234567890", L1=1000, Y=5) F I=1:1:Y F J=1:1:L1 S A=#N(^A(J))	1.8	0.8	3.7	2.0	1
	(S X="1234567890", L1=1000, Y=5) F I=1:1:Y F J=1:1:L1 S A=#D(^A(J))	1.9	0.9	5.8	2.1	1
	(S X="1234567890", L1=100, L2=10, Y=1) F I=1:1:Y F J=1:1:L1 F K=1:1:L2 S ^A(J, K)=X	2.2	0.8	0.8	1.9	1
	(S X="1234567890", L1=100, L2=10, Y=1) F I=1:1:Y F J=1:1:L1 F K=1:1:L2 S A=#A(J, K)	2.1	1.2	3.9	2.1	1
	(S X="1234567890", L1=100, L2=10, Y=1) F I=1:1:Y F J=1:1:L1 F K=1:1:L2 S A=#N(^A(J, K))	2.2	1.1	5.1	2.0	1
	(S X="1234567890", L1=100, L2=10, Y=1) F I=1:1:Y F J=1:1:L1 F K=1:1:L2 S A=#D(^A(J, K))	2.1	1.2	5.3	2.1	1

E15による処理時間を1とする相対処理時間で表示

〔参考文献〕

- 1) 荒牧 達、千葉一彰、川村憲司、山田具男、山口正史、下村正洋： USTAT I ON、住友電気123 1983年 9月
- 2) ANSI: MUMPS LANGUAGE STANDARD、ANSI DR AFT 1982

UNIXはベル研究所の登録商標である。

USTATION上のMUMPSの開発

○ 小林 勝、 久江 正、 上戸 隆

住友電気工業株式会社 ME開発室

〒554 大阪市 此花区 島屋1丁目 1番 3号

U-MUMPSは、68000マイクロプロセッサベースの高性能のワークステーション(USTATION)上で稼動するMUMPSであり、擬似コード(P-code)コンパイラを始め処理速度向上のための種々の技術により、コストパフォーマンスに優れたシステムを提供する。またU-MUMPSはUSTATIONの基本OSであるUNIXのもとに稼動し、UNIXの持つ豊富なソフトウェアツールを利用できる他、日本語サポート等機能面の拡充もなされている。

1. はじめに

USTATIONは、住友電工が開発した68000マイクロプロセッサベースの高性能ワークステーションである。このほど筆者らは、このUSTATIONの基本OSであるUNIXのもとに、MUMPS(商品名:U-MUMPS)を装備したので、これを報告する。

2. U-MUMPSのハードウェア構成

U-MUMPSのハードウェアであるUSTATIONについて述べる。写真1に外観を、図1にハードウェア構成、表1にハードウェア・ソフトウェアの主な仕様を示す。

CPUは32ビットアーキテクチャを持つ、16ビットマイクロプロセッサ68000を使用し、内部バスはVERSA BUSを採用している。メモリは標準1MBで最大8MBまで実装可能である。ディスクは、標準80MB、最大320MBの容量をウィンチェスタタイプのドライブでサポートする。フロッピディスクは両面倍密1MBのドライブを標準1台サポートする。

また標準で、RS232C仕様のシリアルインターフェイス2回線、およびセントロニクスパラレルインターフェイス1回線を装備する。オプションとしてMT等の接続が可能であり、シリアルインタ



写真1 USTATIONの外観

ーフェイスボードは1枚追加することで4回線毎の拡張が可能である。

OSは、UNIX System IIIに4.1BSDの機能を拡張したUniplus⁺を採用し、CPUボードのRAM上に絡納される。また標準でFORTRAN77およびC言語をサポートしている。

なお機種については内部クロック10MHz仕様のCPUを用いたUSTATION(E10)と、内部クロックを12.5MHzにし、キャッシュメモリの付加、ディスクアクセスの高速化を行なったUSTATION(E15)がある。E15はUNIX system Vで稼動する。

3. U-MUMPSの機能

U-MUMPSは、図2に示す5つの基本構成からなり、UNIXの環境下で、MUMPSのマルチユーザ環境を作り出す。以下この構成要素に従って、U-MUMPSの機能を説明する。

表2にU-MUMPSの主な仕様を示す。

(1) 言語コンパイラ

U-MUMPSの言語処理系は、言語コンパイラと、擬似コード(P-code)インタプリタからなり、言語コンパイラはMUMPSコードのシンタックスチェックをした後、P-codeインタプリタの実行形式に翻訳する。これにより、実行速度の高速化を図るとともに、インタプリタの利点である、デバッグの容易性はそのまま生かされている。

U-MUMPSの言語仕様は、ANSI標準(X11.1およびその拡張仕様type A)に準拠するとともに、後述する日本語処理等の拡張を行なっている。

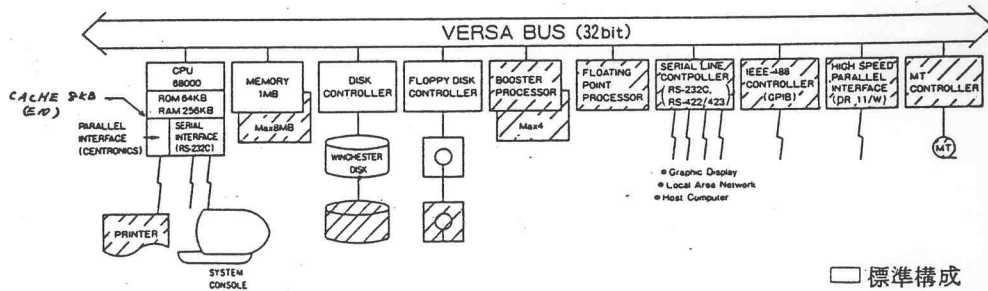


図1 USTATIONのハードウェア構成

表1 USTATION (E10, E15)仕様

<仕様>

ハードウェア		
C P U	MPU	68000マイクロプロセッサ(12.5MHz)
	MMU機能	スタンフォードアーキテクチャ
	ROM	最大64KB
	RAM	256KB
	キャッシュ	8KBアクセスタイム45ns (E15)
U	インタフェース	RS-232Cシリアルインタフェース2回線 8ビットパラレルインタフェース(セントロニクス仕様)1回線
	メモリ	1MB 最大8MB(35Onsec)
ディスク	80MBまたは160MB(アンフォーマット) フロッピーディスク x1 最大2ドライブ	
フロッピーディスク	両面高密度(1MB)および片面単密度(256KB) サポートフロッピーディスク x1 最大2ドライブ	
ディスプレイ	キャラクタディスプレイ 画面サイズ:12インチ 画面構成:80文字(132文字)×24行	
外形寸法	700(高さ)×400(幅)×700(奥行)mm	
重量	最小構成	約90kg
	電源条件	電圧 AC100V±10 % 周波数 50/60Hz
デスク	700(高さ)×1,200(幅)×700(奥行)mm	
チェア		
オプション ハードウェア	補助プロセッサ、インダストリアルMT フローティングポイントプロセッサ グラフィックディスプレイ カラー/モノクロビットマップディスプレイ プリンタ、ストリーミングテープ	
オプション インタフェース	シリアルインタフェース(RS-232C, RS-422, RS-423のうちいずれかを選択)4回線 IEEE-488(GPIB)インタフェース ハイスピードパラレルインタフェース(DR-11/W)	
ソフトウェア		
OS	UniPlus ⁺ (UNIX System V)	
言語	C, Fortran, Assembler	
オプション	Pascal/Lisp, C Prolog, C Basic グラフィックパッケージ CYCORE 日本語レーザ印刷データベース UNIFY, MUMPS 漢字ワープロ、漢字エディタ ウィンドマネジャー	

図2 U-MUMPSのソフトウェア構成

なお、変数名・コマンド等の表現は英大文字同様、英小文字も使用可能である。

(2) データベース管理

U-MUMPSのデータベースはUNIXファイルの一つのファイルとして作成される。ファイル構造はMultiway Btree構造をとるとともにキー圧縮を行なっている。またマルチユーザでの共有データベース利用を可能とするため、排他制御や機密保護等の機能を有する。

グローバルデータのデータ長は、添字・変数とも最大255バイトであり、添字には負の数値をとることもできる。

(3) オペレーティングシステム

モニタ

オペレーティングシステムモニタは、UNIXオペレーティングシステムとU-MUMPSのインターフェイスを行ない、U-MUMPSのマルチユーザ環境を作成する。MUMPSコードで書かれたプログラムの起動、パーティションの割付、動作状態の監視、I/Oスーパーバイザを通してUNIXオペレーティングシステムに対し、周辺装置、端末のサービス要求を行なう。

パーティションサイズは20KBを標準に、実行時にダイナミックに変更できる、ダイナミックパーティション方式をとる。またプログラムの実行形式であるP-codeは、各パーティションに持たず、共有するシェアードルーチン方式であり、プログラムロードのオーバーヘッドを短縮する。

(4) I/Oスーパーバイザ

I/Oスーパーバイザは、システム周辺装置・端末の入出力およびエラー検出等を行なう。入出力それぞれ256バイトのバッファを持ち、ディスクについてはキャッシュバッファが実メモリの大きさに応じてとられ、スループットを向上させている。

またUNIXシステムファイルとのインターフェイスのために、HFS(ホストファイルサーバ)と呼ばれる仮想デバイスが用意されており、これによりUSTATIONがサポートするFORTRAN, C, LISP, PROLOG等の言語による処理系と同期通信が可能である。

(5) ユーティリティライブラリ

日本マンプスユーザーズグループ

表2 U-MUMPSの仕様

メインメモリ	1MB-8MB
ディスク容量 (他の記憶媒体)	80MB-320MB (フロッピーディスク、MT)
最大ジョブ数 (パーティション数)	255ジョブ
ログイン可能な ユーザ数	16ユーザー
パーティションサイズ	ダイナミックパーティション(標準20KB)
拡張コマンド および関数	ZALLOCATE, ZDEALLOCATE, ZGO, ZINSERT, ZJOB, ZLOAD, ZPRINT, ZREMOVE, ZSAVE, ZTRACE, ZWRITE, \$ZBOOLEAN, \$ZCRC, \$ZDATE, \$ZHEX, \$ZVERIFY, \$ZWIDTH
ファイル構造	Multiway B-tree (キーコンプレッション)
文字列型添字	可(英数・記号・漢字・カナ) 最大:255/バイト
パフォーマンス エンハンスメント	アリコンパイル、ダイナミックディスク キャッシング、シェアードルーチン
通信方式	分散型データベース、ネットワークサポート (開発中)
その他	日本語プログラミング、フルサポート upper/lower case character フルサポート ローカル変数は2044/バイトまで可 グローバル添字は255/バイトまで可 負のグローバル添字フルサポート 包括的なユーティリティパッケージ セキュリティ Log-on エラートラップ

U-MUMPSはプログラムの開発支援(エディタ、ドキュメンテーションツール等)およびシステム運用支援(システムジェネレーション、データベース整合性チェック、ジャーナリング等)のための約150本の包括的なユーティリティを持つ。

4. U-MUMPSの日本語処理

U-MUMPSは漢字を含む日本語処理を以下の様にサポートする。

(1) コード体系

外部コードにはJISコード体系(JIS-C6226)をとる。またこの時漢字シフトイン/アウトコードは端末毎に設定でき、JISコード体系をとる任意の端末が接続可能である。ファイルフォーマットには、MS-DOS等で採用され、パソコン等の日本語処理用内部コードの標準になりつつある、シフトJISコードを採用し、他機種との互換性を持たせている。

(2) 漢字入力方式

漢字入力方式はカナ漢字変換方式をとり、漢字辞書・熟語辞書を装備する。

(3) 日本語の使用範囲

グローバル・ローカル変数のデータのみならず、プログラムの名称・ラベル名・テキスト文およびコメント、グローバル・ローカル変数の変数名・添字に漢字を含む日本語を使用できる。

(4) 言語拡張

全ての関数・演算子を日本語処理が可能な様拡張した。また新たに漢字を含むデータの印字幅を計算する関数(\$ZWIDTH)および、カナ・漢字を識別するパターンマッチングコード("カ"および全角"T")を追加した。

```
$ZH($A("漢"))→8ABF   $C(#8ABF)→"漢"  
$ZH($A("A"))→41      $C(#41)→"A"  
$ZW("漢A",4)→3       $ZW("漢A",3)→2.5
```

5. おわりに

U-MUMPSは、処理速度向上を旨とした種々の技術により、本大会デモンストレーションセッション(U-MUMPS)に詳述する様に、従来機種に比べ、E10で平均3~5倍、E15で平均6~10倍の性能を得ることができた。またU-MUMPSは本論文に述べた様に実用的な機能を装備しており経済的なシステム構成を提供し得る。今後さらに、LANとの結合による分散データベースサポートの開発等、より拡張性・発展性に豊んだシステムとしていく計画である。

*UNIXはベル研究所の登録商標である。

日本マンブスユーザーズグループ

1984年度 第11回 日本マンブスユーザーズグループ学術大会
「Implementation」 ⑥

Kanji MUMPS -- Page 1

Kanji MUMPS -- Resolved and Unresolved Issues

○ Donald A. Smith and Ichiro Wakai, M.D.

MUMPS System Laboratory, Nagoya

The MUMPS computer language cannot survive as a viable alternative in the Japanese computer market unless it supports Kanji data processing in a standardized way. In this paper we shall review an approach that, we think, is powerful but simple. We shall then, more importantly, point out those areas where work remains to be done.

As a preliminary remark, let us consider that some people might argue that even if MUMPS standardizes support for two-byte JIS characters, still Arabic, Korean Hangul and other alphabets would be left unsupported. These perfectionists might even go so far as to suggest a three byte character code to enable encoding of any possible earthly alphabet -- but then what about alphabets for beings from other galaxies? In this paper we shall assume rather nationalistically that the JIS codes are adequate for use in Japan. The MUMPS community cannot by itself revolutionize the conventions of the computer industry, and nor can the computer industry please everyone.

We have already explained our ideas for Kanji MUMPS in a proposal (presently a Type B MUMPS release) before the MDC last June in Philadelphia. Please see that proposal for more details. Clearly, MUMPS should support Kanji and other two-byte characters as data values of local and global variables. (From here on, we use the term "Kanji" to include all two-byte characters, unless the context implies otherwise.) This much said, the question arises "What is the maximum length of a Kanji string?" The current MUMPS standard specifies 255 as the answer for ASCII strings, but 255 Kanji characters would require at least 510 physical bytes, more if shift-in and shift-out are necessary. We think that most people would agree that "255" should mean 255 physical bytes for the following reasons: (1) The number 255 was chosen because of the almost exclusive use of binary computers in which an 8 bit byte can express values from 0 to 255; hence strings can easily be represented by 1 length byte and up to 255 physical bytes. To require 255 logical Kanji characters and thus over 255 physical bytes per string would unduly complicate the implementors' task. (2) While we have know of no quantitative measurements here, it is clear that the information carrying capacity of the average Kanji string is higher than that of ASCII strings; for example, single Kanji or jukugo would require many syllables in ASCII. Hence, the limitation to 255 physical bytes (or 127 Kanji characters) is not overly

restrictive.

Kanji should also clearly be allowed in local and global variable names and subscripts, as well as in routine labels. This requirement is easily met by implementors, too. As for Kanji routine names, they too should be supported within the constraints of the operating system environment. (CP/M and MS/DOS, for example, do not allow Kanji routine names.)

It is also clear that the standard MUMPS functions \$EXTRACT, etc. should treat Kanji strings in such a way that single Kanji are counted as single characters despite their physical representation as two bytes. The individual bytes comprising a Kanji are not "seen" by \$FIND or \$PIECE, for example. \$EXTRACT and \$LENGTH count characters LOGICALLY not PHYSICALLY. Please see the above mentioned Type B proposal for examples. Similarly, the READ command option "READ X#10" should read 10 logical characters by default.

\$CHAR and \$ASCII need a little more special treatment. Specifically, they should be based on the JIS codes expressed, of course, in decimal. Hex-to-decimal and decimal-to-hex conversion functions (like \$ZHTD, and \$ZDTH) are useful. Pattern match is a simple matter of deciding upon pattern codes to categorize subsets of two-byte characters. These pattern codes themselves should be drawn from the set of two-byte characters. There are no difficult philosophical issues here, just the need for a committee's decision.

One more point is that there may be times when one wishes to disable the default Kanji processing for functions. At those times, one wants to examine the physical representation of strings. For example, one may be handling binary data, or one may wish to know the physical length of a string because of storage or file transfer considerations. For those occasions, a special variable can switch the MUMPS interpreter between modes: logical counting mode vs. physical counting mode.

Let us turn now to the more interesting and divisive unresolved issues pertaining to Kanji MUMPS.

1. JIS, shift-JIS and other representational issues. (Affects \$NEXT/\$ORDER sorting sequence, READ *X.) Microcomputers have adopted the shift-JIS two-byte encoding method as a de facto standard. But the MUMPS language is still most used on minicomputers, which employ the unshifted JIS encoding method (with shift-in and shift-out bytes sandwiching Kanji strings). Can the MDC decide on one or the other of these methods for internal coding, or can this point be left to the discretion of the implementors? In the latter case, the MDC could still require that \$CHAR, \$ASCII and transfer between MUMPS machines conform to one standard.

Furthermore, some implementors may want to represent all characters as two bytes internally, in which case ASCII characters would have perhaps a 0 as the first byte. Though inefficient in terms of memory, this method does simplify some of the internal processing.

More realistically, some implementors might want to do away with the one byte katakana so that a byte with the sign bit on would combine with the following byte to form a single two-byte character.

The question of internal representation has implications for the sorting sequence of \$NEXT and \$ORDER (and the follows operator "]"). Depending on the internal code chosen, various sorting sequences are easier or harder to implement. For example, in the shift-JIS method of encoding, it is easy to order the characters according to the JIS codes, since shift-JIS and JIS preserve a one-to-one ordering. However, the annoying fact is that with shift-JIS, the one byte katakana sort in between one group of two byte characters and another group of two-byte characters. Is this situation acceptable?

Lastly, we ask in passing what READ *X should return when a Kanji character is inputted? Should it read one physical character dependent on the first byte of the physical representation, or should it wait for the second (or third and fourth) bytes and return the JIS code the way \$ASCII would? If we chose to allow varying internal encoding methods, then the former option (where READ *X depends on the first byte of the internal representation) would yield different results on different implementations.

The MDC cannot make decisions in a vacuum: it must consider ease of implementation in its standards. Otherwise, the language would become too difficult to implement, as well as inefficient and slow.

2. Screen output formatting: "How long is long?" As computer output devices become more sophisticated, problems of managing the output format increase. Years ago, when MUMPS was developed, the standard input/output device was a keyboard with attached printer. Nowadays bit-mapped displays with varying font sizes and orientations are becoming increasingly prevalent. Output characters are no longer all of a single size. Now there is han-kaku, zen-kaku, quarter size, double size. Indeed, on machines like the Apple McIntosh, character generation is done on the fly by the software. There is no telling what width, height and orientation the character will be!

Clearly, \$X and \$Y as currently defined are impotent to handle such an environment. Even now, cursor positioning via escape sequence makes \$X and \$Y often invalid as representing the current cursor position on most MUMPS implementations.

Consider too that MUMPS strings will have at least three different lengths associated with them: the physical length as a sequence of bytes in memory; the logical length as the number of characters in some natural language alphabet; and the output length as a measure of how wide the character appears on the output device. Probably the third form of length will need a new function such as \$WIDTH, while the first two can be handled with a mode flag as mentioned above.

Notice that these problems are not limited to Japanese language MUMPS. American and European MUMPS users will have the same problems.

3. Are 1 and 2 byte spaces equal? The JIS code (2121) represents the two byte space. The code 160 represents the one byte space, while the code 32 represents the standard ASCII space. Are these three equal? The MUMPS standard could provide that on input they be translated to one common code. For example:

```
>S X=$C(160) W $A(X)
32
S X=$C($ZHTD(2121)) F I=1:1:$L(X) W $A(X,I)," "
32 32
```

Alternatively, the two byte space could be converted to one ASCII space rather than two ASCII spaces.

4. Kanji input methods. It is here that the most problems arise in standardization. Input by JIS code, by romanji-to-kana and then to KANJI conversion, by word processing-like jukugo converters, by tablet, even by voice -- all these and more are feasible. The most common method uses a (romanji-to-) kana-to-Kanji converter called by pressing certain control keys on the keyboard. Typically, this method allows entry of one Kanji at a time, in which additional keystrokes select from among the choices provided. Compared to the memorization (or time) needed for entry via tablet or code, this method is a big improvement. But for those accustomed to "intelligent" Japanese word processors which permit input of jukugo or longer phrases via simple syntax analysis, even this method is cumbersome.

Of course, a MUMPS global is an ideal place for storing kanji readings (indexed by sound) and jukugo. Indeed, even on a microcomputer, we have been able to store and process a 15000 entry jukugo file with acceptable response time. The main problem compared to a word processor is the constant necessity to press control keys to switch between 1 and two byte mode.

The suggestion has even been made that the READ command could have parameters specifying what type of read should be performed: regular ASCII, kanji-henkan, jukugo look-up, etc. Of course, normally, the user would want to decide how to input. Only if MUMPS itself handled Kanji interface would this method be useful. In that case the MUMPS program would have to switch modes based on user choice.

To us it seems premature to burden MUMPS with the task of providing these features. Far better to let the operating system handle the numerous problems involved. The MUMPS Standard would otherwise be several hundred pages long. Further, it is more reasonable to have a division of labor in this regard, so that the problems of input can be handled efficiently by a separate module of the operating environment. Too, any decisions in this regard are premature in face of the rapid rate of technological change. Were MUMPS to settle upon one method now, it would fossilize itself into Iron Age technology favored by one tribe of implementor.



「システム デザイン 1」

座長 平川 顕名 (京大病院)

MUMPSによる言葉と知識ベースの設計

里村洋一

千葉大学医学部附属病院医療情報部

千葉市千葉市亥鼻1-8-1

「要約」 次世代のコンピュータにおけるマンマシーンインターフェースは、自然語の理解を基本としなければならない、本稿では、言語の意味構造の解析に基づいて、自然語をいかにコンピュータに記憶させるか、またそれを、どのように検索するか、あるいは、知識を言語とどう結びあわせるかについて考察する。知識カプラーはMUMPS内で実現できるツールとして、この未来システムへの近道であり、まず医学知識について、人工知能の準備を整えるのに好適と思われる。

1. はじめに

マンマシーンインターフェースの問題解決のため、様々な方面でのソフトウェア開発ハードウェア開発が進んでいる。音声認識、文字認識、マウスやタブレットによる入力など実用化の進んでいるものも多い。しかし、真の意味でコンピュータが人間と応答できるためには、コンピュータが入力情報に関して、それを理解する能力を備えていなければならない。今日まで、そうした理解力は、プログラムによって記述され、そのプログラムの許す範囲で動くものとされてきた。しかしながら、患者記録のように、複雑な要因のからむデータを受け取る際には、どんな詳細なプログラムの記述も、その処理には、まず不十分であると思える。なぜならば、コンピュータは、人間の使用する言葉に対してほとんど無知であり、どんなに推論機能が充実していても、思考の材料となるデータに関する言語的解析と知識がないかぎり、その機能は空まわりするだけだからである。

次世代のコンピュータは、人工知能を実現するものとされているが、単にすぐれた推論マシンを作りあげるだけでは十分でない。人間の知識が、どのような構造をし、どのように連がり合っているかを知り、それを記憶しなければ、生れたての赤んぼうのようなものであろう。

今日まで、医療情報処理の分野でも、自然語入力や、その解釈、それによる医学知識の利用などが試みられてきてはいる。L. L. Weed たちのPROMISプロジェクトの中で使用している知識カプラーもその一つであり、同様な考えは、COSTARにも採用されている。しかし、これらは医学知識や自然語を特定の目的の入力に較べて定義された形で作成されており、予想外の入力に対して対応する力は持っていない。PROMISやCOSTARが優れた能力を持つシステムである事は否定しようもないが、未来のコンピュータを考える時、さらに応用力の広い知識やデータの蓄積方法とその利用論理を考えざる

を得ない。

演者は、自然語の構造解析を基にして、医学用語とそれによって記述される医学知識及び患者データをいかにコンピュータ内に記録するかを考察し、Weedらの知識カプラーの応用拡大を試みている。

ここでは、その概念設計について論じる。

2. 自然語の構造

言語の構造には、音韻構造、文法構造、意味構造、などがあるが、音声確認は別な問題とし、全てのデータは文字化されていると仮定すると、文法と意味論を考えればよいことになる。文法構造についての考察は別の機会とし、ここでは、意味構造について考察してみる。

言語は、人間が感覚や体験に基づいて認識する小数の基礎的な言葉と、これら結び合わせることによって、合成したり、新しい概念を作り出したりする事によって作られる言葉とによって成り立つと考えられる。例えば、自分 \leftrightarrow 他人、快い \leftrightarrow 痛い、から味方 \leftrightarrow 敵のような言葉が連結によって作られよう。日常のほとんどの言葉は、このような他の言葉や概念の連結によって人間の中で定義されているようである。そこで、このような連結方法、つまり言葉の関係を解析する事によってその意味構造を解析する事ができ、これを基にして、コンピュータ内で言葉を定義する事が可能であると思われる。このような言葉の意味構造をSemantic Structureといい、言語学の重要な分野Semanticsの対象となっているが、本稿では、コンピュータ内で言葉を定義する事に目的を絞って、やや独断的ではあるが、言葉の関係を抽出してみる。

3. 言葉と言葉の意味関係

言語学者達は、言葉の意味関係について詳しく分析している。我々が、利用し得るであろうと思われる関係の分類について述べてみよう。

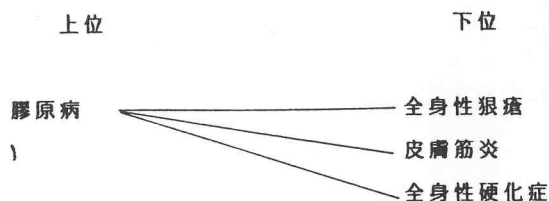
a. 同義関係 (Synonymity)

ある語と別の語が同じ意味を持つ関係である。「肺癌」と「肺の癌腫」はこの関係にあると考える。また「Lung Cancer」も同様であろう。しかし、言葉が異なれば必ずからニュアンスの違いがあるのが当然であり、また、しゃべる人によって意味の相異のある事もある。しかし、一般的にまず共通概念を表わすものはこの関係に入れられる。この関係は特に、自動翻訳などで重要であり、また、自動コーディングの基本的なデータ構造でもある。著者は、主としてこの関係だけを記述したカプラーによって、自動コーディングのシステムを試作している。より一般的には、話者間に限られた同義語もある。「種」(たね)といえは、一般には植物の種子のことであるが、不妊について話している医者の間では「精子」と同義であり、新聞記者の間では記事の材料と同義である。このような、

状況に応じた同義語の定義は、コンピュータ処理上から最も困難な課題のように思える。

b, 上位語、下位語、関係 (hypernym、hyponym)

hierarchicalな分類をする時に成立する関係で、1の同義語と同様、自動コーディングにも使用される。下位の語は上位の語の表わす概念の中の一つとして定義される。



下位語同志の間に両立性がなければ、これらの間に共下位語という関係が成立する。この関係は、分類の識別において利用価値が高いように思える。

c, 非両立、(incompatibility)

これは、2語間ではなく3語以上の間の関係を表わし、ある1つの語の意味する所は、残り全ての否定を意味する。分類においては、下位の同レベル間で前記のような共下位語として incompatibilityが成立しているのが正しい。

d, 反義 (Antonymy)

多い ↔ 少ない、男 ↔ 女のように対極的關係であり、事象を峻別する。

e, 反対 (Coversness)

一つの事象が相互に成立する関係、親 ↔ 子、取る ↔ 与える などがその例である。

f, 部分全体関係

手 ↔ 指、胃 ↔ 幽門などのように、ある語が他の語の一部を意味する関係である。これは、一見、上位下位関係と同じに思われるが、指は手の一種ではないところから、この関係を区別した方がよいと考えられている。

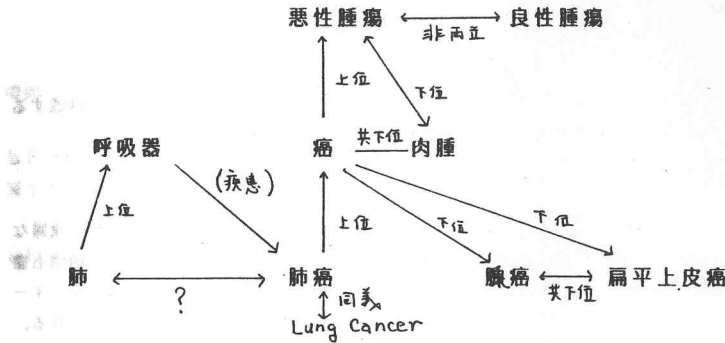
g, その他の関係

これ他に、順序関係、内外関係、水平関係、縦関係、空間動移動関係などがある。これらは、動詞や修飾語において成立する事が多い

4. 言葉に関するデータベースの作成

このような、言葉の意味構造をベースにして、自然語理解を実現するための、コンピュータ内の辞書、即ち、言葉のデータベースを作成する事を考えてみよう。

ここでは、知識カプラーの応用が最も単純でかつ明解である。各々の語の間に、上記の関係を記述する関係子を用いて結び合わせの作業を順次行ってゆく、これは、丁度、子供が新しい言葉を覚えてゆく過程と同じであろう。今、「肺癌」という言葉が新しく作成されたとしよう。そこでは、下図のような関係が考えられる。



この図において、肺と肺癌の関係を考えてみよう。これは、先きのべたような関係とは異ったものである。肺癌は肺の一種でもなければ肺の一部でもない。肺は肺癌という言葉形成している一部分である。そこで、言葉を分解して、その全体と部分の関係を追加しておくのが良いと思われる。

5. 言葉と知識の結合

言葉と言葉の意味構造による関係結合は、それ自身知識の一形態であるが、このような基本的関係だけでは、知識を迅速に検索する事は困難である。そこで、知識関係という特別な結合を考えて、これを利用する事にする。これは、L. L. Weed らの知識カプラーの形に他ならない。この面での関係では、原因と結果、疾患と治療法、症状と疾患、などのように、その利用方法に基づき任意に設定してゆくべきであろう。

6. おわりに

未来のコンピュータシステムは、言葉を理解できなければならない。知識は言葉の上に成り立つものである。とすれば、言葉のデータベースは、知識ベースに先立つものと考えられよう。

実際に人間に匹敵する語彙を持つコンピュータの実現はそれほど夢物語ではない。既にワープロに於て、かなりの実績がある。我々はこれらの語彙に知識を結びつける努力をする時期に来ていると考える。

「システムデザイン1」 ②

MUMPUSによる集計ルーチンとファイルエディター

○佐藤 甫夫, 佐藤 真理, 古閑啓二郎, 岩佐 博人
山内 直人

千葉大学医学部 精神医学教室

千葉市亥鼻1-8-1

〔要約〕

データファイルの集計用プログラムを作成する汎用ルーチン (GLC) とファイル編集プログラムを作成する汎用ルーチン (GLD) を作成した。上記汎用ルーチンの概略を述べプログラムリストを付記した。

〔序〕

データファイルの集計プログラムをMUMPUSで書くと、簡単なプログラムは1-2行で足りるが、複雑なものになると、10行-20行に及ぶ事も屢ある。且つ、多角的な集計になると、同様なプログラムを何回も書く、必要が生じてくる。このような場合、汎用ルーチンによって、プログラムの手順を一様化して置くと、エラーミスを防止するだけでなく、プログラム点検の手間が省けるので実用上のメリットが大きいと考えられる。

我々は、実際に集計を行うに当たって、上記の汎用ルーチン、GLC、GLDを作成使用したので、以下にそのプログラムについて述べる。

〔一般的事項〕

GLC、GLDともコマンドモードにて実行 (D GLD, D GLC) する。

実行に先立ってプリンター番号 (%DEVN)、一行の字数 (%LENGTH) が定義されているものとする。

〔GLC〕

データファイルとして ^PSY (A, B, C, D) =データ1^データ2
の形を予想している。

A, B, C, D等は、データのカテゴリーを示すノードデータで例えば、年度、病名、性別、I
D番号などである。データとノード (A, B, C, D) は1対1に対応しているものとする。

D GLC を実行すると、パラメーター入力後、ローカル変数STEP (1), . . . STEP (n) に
以下のようなプログラムが作成される。

*この例では n=4

```
STEP(1)="X LR(1) F I1=0:0 X LS(1) D CPL1:%VV="" Q:%VV="" X LP
(%DS),STEP(%DS+1)"
STEP(2)="X LR(2) F I2=0:0 X LS(2) D CPL1:%VV="" Q:%VV="" X LP
(%DS),STEP(%DS+1)"
STEP(3)="X LR(3) F I3=0:0 X LS(3) D CPL1:%VV="" Q:%VV="" X LP
(%DS),STEP(%DS+1)"
STEP(4)="X LR(4) F I4=0:0 X LS(4) D CPL1:%VV="" Q:%VV="" X LP
(%DS) D DSUM"
```

作成されるプログラムは、パラメーターの値により多少異なる。例えば

SQSUM. . . . データ1の二乗和をとる場合=1, NO=0

AVE. データ1, データ2の平均をとる場合1, NO=0

これらの値に応じてデータ集計後の処理プログラム、CPLnを選択して、STEP (n) に組み込む。

プログラム実行がSTEP (n) に来たとき、A, B, C, Dに対応するノード変数の内容は、それぞれ、
N1, . . . Nn に定義され、%VARにはグローバル変数 ^PSY (N1, . . . , Nn) が定義され

ている。また、%VV=Nn となっているので、%VV=" " を上記無限FORループのQUIT条件として用いている。

最終レベル(データの入っているレベル) n は DSUM と云うルーチンで集計を行う。集計の前に LP (n) が定義されていて、[通常, LP (1), . . . , LP (n-1) = " "], LP (n) の中で %S, %SSに、データ1, データ2が定義されていなければならない。

通常, LP (n) = というプロンプトが出た時

S %S=@%VAR, %SS=\$P (%S, " ^", 2), %S=\$P (%S, " ^", 1)

```
LP(1)=" "
LP(2)=" "
LP(3)=" "
LP(4)="S %S=@%VAR,%SS=0"
```

の如く、データ1, データ2を %S, %SSに移動するプログラムを入力する。

LP (n) の中では、このほか、N1, . . . Nn, %VAR を使用して、のちに利用する変数 N11, N12, N21等を定義する事も出来る。

LR (n) は、nレベルのカウンターをリセットするプログラムで、GLCが自動的に作成する。

```
LR(1)="S %ZDS=1,N1="" X CRS(1)"
LR(2)="S %ZDS=2,N2="" X CRS(2)"
LR(3)="S %ZDS=3,N3="" X CRS(3)"
LR(4)="S %ZDS=4,N4="" X CRS(4)"
```

```
CRS(1)="S %ZCS=1 S @("C1"_"5=500000") F %ZCRS=1:1:4 S @("C1"_%ZCR
S_"=0")"
CRS(2)="S %ZCS=2 S @("C2"_"5=500000") F %ZCRS=1:1:4 S @("C2"_%ZCR
S_"=0")"
CRS(3)="S %ZCS=3 S @("C3"_"5=500000") F %ZCRS=1:1:4 S @("C3"_%ZCR
S_"=0")"
CRS(4)="S %ZCS=4 S @("C4"_"5=500000") F %ZCRS=1:1:4 S @("C4"_%ZCR
S_"=0")"
```

LS (n) は、下位ノードの部分集計結果を処理するプログラムで、同様にGLCが作成する。

```
LS(1)="S %ZDS=1 D CUPB:%ZDS<%ZCS,ASN,CUP:%VV="" "
LS(2)="S %ZDS=2 D CUPB:%ZDS<%ZCS,ASN,CUP:%VV="" "
LS(3)="S %ZDS=3 D CUPB:%ZDS<%ZCS,ASN,CUP:%VV="" "
LS(4)="S %ZDS=4 D CUPB:%ZDS<%ZCS,ASN,CUP:%VV="" "
```

集計結果は、通常 \wedge PSY (A, B) = result1デリミター1 \wedge result2デリミター2 の如く入るが、ノードBを例えばBsと特定の値に指定した場合は、変数名を自動的に変更して \wedge PSY2NC (Bs, A, C) = result1デリミター1 \wedge result2デリミター2 の如く自動的にセットされる。

総計は \wedge PSY ([ALL])に入る。尚、上記の結果、つまり result1デリミター1 \wedge result2デリミター2 の後ろに データレベル レファランスの回数、N=. . . %Sの最大値が MAX=. . . %Sの最小値が MIN=. . .

と自動的に追加されて入る。

上記のデリミターは、GLC のなかで、DEL1 ? = 等のプロンプトが出た時

(AGE-AV) 等 と入力する。

```
DEL1="(DEMO)"
DEL2="(DEMO2)"
```

ASN, AS 等は、GLC, GLDで汎用されるサブルーチンで、ファイル中に実在するノードデータやグローバル名を、N1, . . . Nn, %VARにセットしprogramableな変数とするルーチンである。

[GLCの変数名入力]

GLCのなかで、 variable name ? というプロンプトが出た時
^PSY ("*", "*", "*", "*") と入力する。
ノードを特定する場合 ^PSY ("*", 295.0, "*", "*") と入力。

この入力により、GLCはファイル名とデータノードのレベルを知り、日付、特定ノードの有無を知って必要なふろの作成を行う。

以上が GLCの概要である。(詳細はリスト参照)

```
GLC="X STEP(1) R !."GLC END",!, "PROGVAR PRINTOUT ? <Y>=1.<NO>
=0 1/0 ? =", %S Q:%S=0 0 %DEVN U %ZDEVN:%LENGTH D PROGVAR C %D
EVN U 0 Q"
```

[GLD]

GLDは、GLC と同様に STEP (1), . . . STEP (n) を定義するルーチンである。

変数名とデータレベルの指定は

variable name ? ^PSYADM

node level ? 4

の如く下線部を入力して行く。すると、GLDは STEP (1), . . . STEP (4) を作成する。

```
STEP(1)="S %ZDS=1,N1=N1S F I1=0:1 S %ZDS=1 D ASNS S @("QC="_N1Q)
Q:%ZVV="!"QC X STEP(2)"
STEP(2)="S N2="" F I2=0:0 S %ZDS=2 D ASN Q:%ZVV="" X STEP(3)"
STEP(3)="S N3="" F I3=0:0 S %ZDS=3 D ASN Q:%ZVV="" X STEP(4)"
STEP(4)="S N4="" F I4=0:0 S %ZDS=4 D ASN Q:%ZVV="" X LP"
```

GLDでは、データのボトムレベルでどのようなプログラム処理を行うか入力求め LPにセット

LP-global set ? のプロンプトの時 1 を入力すれば

global name ? ^PSYLP1 の入力で、予め^PSYLP1にセットされているプログラムが LPに入る。

global set ? 0 の時は LP=のプロンプトに対し、処理プログラムを直接入力(255文字、1行以内のプログラム)する。

```
LP="S ^PSYADDD(N1,N3,N4)=@%VAR W !,N1"
```

N1の初期値とN1に関するquitコンディションを指定出来るようになっていたので、ファイル作成を何台かの端末を同時に使用して、短時間でを行うことが可能である。また、遠方で動いている端末の動作を監視する事も出来る。

通常、GLCの前処理として、GLDによりファイルを作成して置くと便利である。

(まとめ)

以上、汎用集計ルーチンGLCとファイルエディターGLDについて述べた。このルーチンはいずれも単純化、モジュール化されており、機能の拡張、修正が容易である。今後さらに機能の充実を図って行きたいと考えている。

(付記)

以下に、プログラムリストの一部 (GLC, GLDを動かすために必要なサブルーチンを全て含む) を添付する。

また、GLC, GLD 動作時のローカル変数の値も一部添付する。

(GLC 動作時)

```

XC3F="0"
XCC="1"
XCPL="3"
XCPLL="2"
XCRS="3"
XCS="3"
XD="C4"
XDD="1"
XDEVN="105"
XDS="3"
XLENGTH="52"
XPC="1"
XPC(1)="1"
XPRINTF="0"
XS="80"
XSS="0"
XV="291.8","M","1495733",""
XVAR="^PSYADM4("291.8","M","1495733","")"
XVARCPL="^PSYADM4("291.8","M","1495733")"
XVV=" "
XVVV="291.8","M","1495733"
AVE="1"
C11="2342"
C12="0"
C13="25"
C14="323 (HPD/A)"
C15="8 (HPD/A)"
C21="0"
C22="0"
C23="0"
C24="0"
C25="500000"
C31="80"
C32="0"
C33="1"
C34="80 (HPD/A)"
C35="80 (HPD/A)"
C41="80"
C42="0"
C43="1"
C44="80 (HPD/A)"
C45="80 (HPD/A)"

```

(GLD動作時)

```

%PRINTF="0"
%S="Y"
%V="290.4","F","1411528","1"
%VAR="^PSYADM4("290.4","F","1411528","1")"
%VV="1"

N1="290.4"
N1R="0"
N1S=" "
N2="F"
N3="1411528"
N4="1"
QC="0"
SLEVEL="4"

GSET="0"
I="4"
I1="6"
I2="0"
I3="0"
I4="0"

```



```

AS      S %V=N1,%VV=N1 I %DS>1 F %AS=2:1:%DS S @("%VV=N"_%AS),
%V=%V_"", "%V_"_VV
      S %VAR=VARN_"(""%_V_"")" Q
ASNS    I N1=""&'I1 D AS Q:%D(@%VAR) S @("N1=%ZS("_%VAR)")
D AS Q
ASN      D AS S @("N"_%DS_"=%ZS("_%VAR)") D AS Q

```

```

CRSD    S CRS(I)=""S %CS="_I_" S @("C"_I_"""""5=500000""") F %
CRS=1:1:4 S @("C"_I_"""_%CRS_"="0""") Q
LSD      S LS(I)=""S %DS="_I_" D CUPB:%ZDS<%CS,ASN,CUP:%VV=""
Q
LRD      S LR(I)=""S %DS="_I_",N"_I_"=""X CRS("I_"") Q
STEP1A  S STEP1=""X LR("I_"") F I"_I_"=0:0 X LS("I_"")_%S('SQS
UM%AVE:" D CPL1","'SQSUM%'AVE:" D CPL5",1:" D CPL")_%VV=""
Q:%VV="" Q
STEP1B  S STEP1=""X CRS("I_"") S %DS="_I_" S N"_I_"=ARF(%DS) D
AS I %D(@%VAR)_%S(SLEVEL>I:">9",1:"=1")_" " Q
STEPD    D @(%S(ARF(I)=""*"":STEP1A",1:"STEP1B")) S STEP(I)=STEP
1_"X LP(%DS)_%S(I=SLEVEL:" D DSUM",1:"STEP(%DS+1)") Q
STEPDD  F I=1:1:SLEVEL D STEP1D,CRSD,LSO,LRD
Q

```

```

CUP      Q:%CS=1&%DS=1) S @("%CC=C"_%(CS-1)_"4<C"_%CS_"4") S:
%CC @("C"_%(CS-1)_"4=C"_%CS_"4") S @("%CC=C"_%(CS-1)_"5>C"_%CS
_"5") S:%CC @("C"_%(CS-1)_"5=C"_%CS_"5") F %CRS=1:1:3 S @("C"
_(CS-1)_%CRS_"=C"_%(CS-1)_%CRS_"+"_%CS_%CRS)
S %CS=%CS-1 Q
CUPB     F %BS=0:0 Q:%ZDS=%CS D CUP
Q
DSUM     S %D="C"_%SLEVEL,@(%D_"1="_%D_"1+%S"),@(%D_"2="_%D_"2+%
SS"),@(%D_"3="_%D_"3+1"),@("%DD="_%D_"4<%S") S:%DD @(%D_"4=%S"
) S @("%DD="_%D_"5>%S") S:%DD @(%D_"5=%S") W !,N1 Q

```

```

CPL      S @("%C3F=C"_%ZDS_"3=0") Q:%C3F D CPL0 S %SQ=%ZSS-(%S+%
S),@("%SQE=C"_%ZDS_"3>1,%SQ=%S(%SQE:(%SQ+C"_%ZDS_"3)/(C"_%ZDS_"3-
1),1:"""-""")) D SQ,CPL3,CPL2 Q
CPL0     S @("%S=C"_%ZDS_"1/C"_%ZDS_"3,%SS=C"_%ZDS_"2/C"_%ZDS_"3")
D DROUND Q
CPL1     S @("%C3F=C"_%ZDS_"3=0") Q:%C3F D CPL0,CPL3,CPL4 Q
CPL2     S @%VARCPL=%S_DEL1_"^SIGMA="_%SQ_"^N="NNN"_"^MAX="MAX
_"^MIN="MIN Q
CPL3     S @("NNN=C"_%ZDS_"3"),@("MAX=C"_%ZDS_"4"),@("MIN=C"_%ZDS_
"5") S:%ZDS=1 %VVV=%S('ARF:"[ALL]",1:ARF(ARF)) S:%ZDS>1 %VVV=%S(
'ARF:"",1:ARF(ARF)) G CPL3+2:%ZDS=1&'ARF,CPL3+1:'ARF F %CPL=ARF
+1:1:SLEVEL S:ARF(%CPL)'="*" %VVV=%VVV_"", "%V_"_ARF(%CPL)
S %CPLL=1 S:%VVV="" %VVV=N1,%CPLL=2 F %CPL=%CPLL:1:%ZDS
-I S:ARF(%CPL)=""* @("%VVV=%VVV_"", "%V_"_N"_%CPL)
S %VARCPL=VARN_%S('ARF:"",1:ARF_"NC")_"(""%_VVV_"")"
Q
CPL4     S @%VARCPL=%S_DEL1_"^_%SS_DEL2_"^N="NNN"_"^MAX="MAX"
_"^MIN="MIN Q
CPL5     S @("%C3F=C"_%ZDS_"3=0") Q:%C3F S @("%S=C"_%ZDS_"1,%SS=
C"_%ZDS_"2") D CPL3,CPL4 Q

```

```

SROUND  S %S=%S*1000+0.5,%S=%S#1,%S=%S/1000 Q
DROUND  D SROUND S %SS=%SS*1000+0.5,%SS=%SS#1,%SS=%SS/1000 Q
SQ       Q:%SQ="" S %S1=%SQ/3,%S2=0 G:%S1=0 SQ+1 F SQ=1:1:15
S %S2=(%SQ/%S1)+%S1)/2,%S1=%S2
S %SQ=%S2,%SQ=%SQ*10000+0.5,%SQ=%SQ#1,%SQ=%SQ/10000 K
%S1,%S2 Q
SQA      S %SQA=%S1-%S2 S:%SQA<0 %SQA=-%SQA Q
OPEN     Q:%PRINTF %DEVN U:%PRINTF %DEVN:%LENGTH Q
CLOSE    C %DEVN U 0 Q

```

```

GLC      R !,"GLOBAL SUMMATION PROGRAM",!,"DEFINE GLOBAL NAME A
ND NODES SUCH AS ^PSYICDW("","*",,"N79","*",,""),!,"*"*="NODE
UNSPECIFIED",!,"GLOBAL NAME ? =",%S S %S=$P(%S,""),1)
      S VARN=$P(%S,"(",1),%VV=$P(%S,"(",2),ARF=0 S:'$D(%
PRINTF) %PRINTF=0 F %CRS=1:1 S ARF(%CRS)=$P(%VV,"",,"",%CRS)
Q:ARF(%CRS)=" S:ARF(%CRS)'="*%&'ARF ARF=%CRS
      S SLEVEL=%CRS-1 R !,"IF YOU WANT TO CALCULATE SIGMA OF
%S INPUT <1>,IF NO <0> SQSUM ? =",SQSUM,!,"AVERAGE ? <YES>=1
,<NO>=0 ",AVE
      W !,"YOU HAVE TO DEFINE LP(I) FOR EACH NODE LEVEL",!,"
LP(I) MEANS LEVEL PROPER JOB. %DS=THE LEVEL OF THE RELEVANT N
ODE.",!," DEFINE THE EXECUTABLE JOB LIKE:""S N21=$E(N1,1,3)""
OR JUST <CR>"
      W !,"YOU CAN USE %DS,%VAR,N1,N2,N3 ETC TO DEFINE OTHER
VARIABLES",!,"YOU HAVE TO DEFINE %S AND %SS AT THE BOTTOM LEV
EL OF NODE,SUCH AS ""S %S=%XVAR,%S=$P(%S,"")",1),%SS=$P(%S,""
("",2),%S=$P(%S,"("",1)""
      W !,"THESE %S AND %SS ARE AUTOMATICALLY SUMMED UP INTO
C11,C12,C21,C22 ETC. C13,C23,C33 ARE TIMES OF REFERENCE TO TH
E BOTTOM LEVEL NODE"
      F I=1:1:SLEVEL W !,"LP(",I,")=" R LP(I)
      S GLC="X STEP(1) R !,"GLC END",!,"PROGVAR PRINTOUT
? <Y>=1,<NO>=0 1/0 ? ="",%S Q:%S=0 0 %DEVN U %DEVN:%LENGTH
D PROGVAR C %DEVN U 0 Q W !,"IF YOU WANT TO EXECUTE THIS PROG
RAM ""X GLC""<CR>",! Q
      PROGVAR W !,"VARN=",VARN," STATIST=",STATIST D STEPW,LSW Q
      STEPW F I=1:1:SLEVEL W !,"STEP(",I,")=",STEP(I)
      Q
      LSW F I=1:1:SLEVEL W !,"LS(",I,")=",LS(I)
      Q
      Q
      GLD      R !,"GLOBAL NAME ? (E.G. ""^PSYHOSP"" ) & TO WHAT NOD
E LEVEL ?",!,"GLOBAL NAME ? ",VARN," TO WHAT NODE LEVEL ? "
      ,SLEVEL R !,"STARTING VALUE OF N1 ? <CR> OR ANY VALUE. N1 ? ="
      ,N1S
      R !,"ANY SPECIAL QUIT CONDITION FOR N1(E.G N1>500000 E
TC <NO CONDITION>=0",!,"N1 QUIT CONDITION ? =",N1Q
      R !,"LP SET FROM GLOBAL ? <Y>=1,<NO>=0 ",GSET I GSET
      R !,"LP-GLOBAL NAME ? ",%S S LP=%S W !,"LP=",LP," IS SET",!
      G GLD+4
      R !,"LP ? =",LP,!,"ARE YOU SURE <Y> ? ",%S G:%S'="Y" G
      LD+3
      S:'$D(%PRINTF) %PRINTF=0 S:'$D(%LENGTH) %LENGTH=80 R:'
      $D(%DEVN) !,"PRINTER NO. ? ",%DEVN R:'$D(STATIST) !,"STATIST(E
      .G 831231 ETC) ? =",STATIST F I=1:1:SLEVEL D LPD
      W !,"VARN=",VARN," SLEVEL=",SLEVEL,!,"LP=",LP D STEPW
      R !,"ARE YOU SURE ? <Y> OR NO ",%S G:%S'="Y" GLD X STEP(1) W
      !,"VARN," GLD END",! Q
      Q
      LPD      S' STEP(I)=$S(I=1:"S %IS=1,N1=N1S ",1:"S N"_I_"=""" )
      _"F I"_I_"$S(I=1:"=0:1",1:"=0:0")_" S %DS="_I_"$S(I=1:" D ASN S
      Q("QC="N1Q) Q:%VV="""!QC ",1:" D ASN Q:%VV=""" )_"$S(I
      =SLEVEL:"X LP",1:"X STEP("_(I+1)_"") Q

```

剖検輯報データ・ベースを例として

— データチェックにおける日本語処理の効用

○福永泰明^{*}，馬場謙介^{**}，田久浩志^{***}

*産業医科大学情報管理課，**同第2病理学教室，***同振動室
北九州市八幡西区医生ヶ丘1の1

日本病理学会編輯の剖検輯報の巻末の疾病別頻度表の作成は，1974年次の剖検症例からコンピュータ化された。1982年次剖検症例分の頻度表作成作業に日本語を導入した。最終目的である頻度表をまず日本語化した。日本語化はこれに止まらず，入力データの修正過程にも適用した。このことによりデータ修正工数を17日減すことが出来た。

§ 1. はじめに

日本病理学会編輯の剖検輯報は昭和33年(1958年)の剖検症例をまとめて，第1輯として発行されて以来，毎年1年間の全国の剖検例を集録して発行していることはよく知られている¹⁾。昭和49年(1974年)の剖検症例から，剖検輯報掲載症例の頻度表をコンピュータで形成していることはすでに報告した²⁾³⁾。頻度表を作成するに当たってグローバルに収納されたデータは検索用のデータ・ベースとしていろいろな研究に應用されている。

近年，日本語処理が導入され，昭和57年(1982年)の剖検症例より疾病別頻度表作成に日本語を取り入れた。剖検輯報の疾病別頻度表作成のコンピュータ処理は入力データの正確さの検査段階に非常に工数がかかるという性質がある。そこで，日本語表示による帳票の見やすさの向上だけでなく，入力データエラーの修正段階にも取り入れ，データ処理における人とコンピュータの仲介作業段階の作業工数低減を図ることが出来た。これらの点に焦点を絞って報告する。

§ 2. 剖検輯報の概要

剖検輯報は大別して，病理剖検所見を記述した「記述部分」とその頻度を掲載した諸集計表からなる。記述部分は剖検施設，剖検番号，住所，年令，職業，性，臨床診断，出所，68字までの所見，備考からなる(図1)。

病理所見は主病変と副病変とに分けられる。巻末には年間の全国の剖検例の疾病別年令階級別性別頻度表が記載されている(図2)。1973年次剖検例までは主病変，副病変の中から重要な疾病を手作業で表の形にまとめていたが，1974年次分からは主病変をコンピュータを使って疾病別頻度表の形で集計している。データ量は年々増加の傾向にある(図3)⁴⁾。頻度表作成のコンピュータ化が始まった1974年次分では，剖検症例数で23,163症例，病名コードの種類で443種類，記載施設数で286施設であったのが，

A82-1 小倉南区	42才 無	5 職	成人T細胞白血病 (松井病院)	重複癌 1)T細胞白血病+2甲状腺癌小癌(乳頭腺癌)転:1)骨髄,肝,脾,肺,腎,皮膚,膵臓,胆膵 2)なし ①:クリプトコッカス症 2. 腔水症	皮膚
A82-2 八幡東区	40才 運転	2 手	多発性骨髄腫 (1内)	多発性骨髄腫(IgA型)(脂肪腫)転:脊椎,小竹盤腔内後腹膜,左壁側胸膜,肋間軟部組織 1. 圧迫性尿管閉塞 2. 水腎症 3. 肺水腫	制癌
A82-3 若松区	75才 無	5 職	結核,左無気肺 (2内)	左無気肺(100f,形成不全(組織奇形?)) 1. 右器質性肺炎+気管支肺炎(1080f) 2. 心両室肥大(440f,慢性肺性心)	抗生
A82-4 八幡西区	66才 58才	3 師	肝臓,胃癌 (3内)	肝細胞癌(多発結節性,索状型)転:肝,胆膵,門脈-組管内腫瘍転 1. 肺腸破裂による腹腔内出血(2300ml) 2. 胃癌 3. 十二指腸潰瘍	制癌・輸血
A82-5 若松区	58才 無	3 職	急性骨髄性白血病 (1内)	急性骨髄性白血病(くすり型からの類在化,骨髄細胞腫) ①: 右肺動脈血栓 2. ムコール菌症(肺動脈血栓,気管支) 3. 心肥大	放・制癌
A82-6 八幡西区	69才 僧	3 僧	肝臓 (2外)	肺癌(右上葉,扁平上皮癌)転:気管 1. 両側器質性肺炎 2. 両肺尖癒着 3. 左胸水 4. 食道潰瘍 5. 胃癌癒着 6. 右腎囊腫	放・制癌
A82-7 八幡西区	4生日	5 職	心不全(心奇形) -野不全(小児)	新生児+未熟児(在胎38週,2352f) 1. 大動脈狭窄症(管前型)+心室中隔欠損,動脈管閉存 2. 実質臓器の乏酸素性変化	放・制癌
A82-8 小倉南区	67才 主	5 職	肺癌 (1外)	肺癌(肺頭部,低分化腺癌)転:肝,肺,膵臓,膵膵膵症 リ)膀胱,後腹膜,肝門部 1. 腹水(3500ml) 2. 肝内胆汁停滞 3. 胸水	手・放
A82-9 小倉北区	58才 消防	5 士	多形性腺芽腫 (脳外)	多形性腺芽腫(右前頭-頭頂葉) 1. 急性化膿性髄膜炎 2. 器質性気管支肺炎 3. 肺膿瘍 4. 感染脾 5. 全身黄斑 6. 脂肪肝	手・放
A82-10 中野市	56才 元市議	5 長	肺癌 (1外)	肺癌(頭部,囊胞腺癌)転:胃,小腸,大腸,肺,腎,副腎,リ)脾,胃 1. 気管支肺炎 2. 肺うっ血水腫 3. 胃びらん 4. 腹水500cc	制癌
A82-11 若松区	66才 自営業	3 業	硬膜下出血+慢性腎不全(2内)	幕縮腎(細胞脈硬化性+左腎動脈起始部狭窄,50:80f) ①:硬膜下血腫+脳出血+ヘルニア 2. 解離性大動脈瘤 3. 心肥大 4. 動脈硬化	放・制癌・輸血
A82-12 遠賀郡	17才 高校	3 生	リンパ上皮腫 (1内)	低分化扁平上皮癌(Regaud型リンパ上皮腫)放射線照射,化学療法後の状態 転:両肺,肝,胸骨,脊椎骨,脾 リ)肝門,気管 1. 胸腹水	放・制癌・輸血
A82-13 遠賀郡	73才 無	3 職	前立腺癌+骨転移 (転移)	前立腺癌(低分化腺癌)転:肺,肝,腎リ)膀胱,膀胱,膀胱 1. 胃+十二指腸ビラン 2. 消化管内容物貯留,3. 気管支肺炎	放・手
A82-14 八幡東区	70才 無	3 職	肺癌 (2外)	肺癌(右中下葉,中分化扁平上皮癌)再発 転:前立腺癌 ①: 間質性肺炎 2. 右肺動脈狭窄	放・手
A82-15 八幡西区	25才 運転	3 手	成人T細胞白血病 高Ca血症(1内)	T細胞白血病	

主表: 悪性腫瘍(原発部位別)

(その1)

病 変	年 令		消 亡 年 令																				年 令 不 詳			
	消 亡 合 計	死 産 児	新 生 児	1	4	9	14	19	24	29	34	39	44	49	54	59	64	69	74	79	84	89		94		
口 咽	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
舌	95	0	0	0	0	0	0	0	0	1	0	2	4	6	7	14	13	16	11	10	9	2	0	0	0	
大 唾 腺	27	0	0	0	0	0	0	0	0	0	1	2	0	1	0	0	1	0	4	0	0	3	0	0	0	0
歯 肉	13	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	2	1	2	3	1	1	0	0	0
他 の 口 腔	63	1	0	0	0	0	0	0	0	0	0	0	0	1	6	8	9	10	12	8	6	2	0	0	0	0
咽 頭 NOS	41	0	0	0	0	1	1	0	0	1	0	1	1	6	4	5	4	6	5	3	2	1	0	0	0	0
上 咽 頭	41	0	0	0	0	0	0	2	1	0	1	0	4	3	7	8	8	1	1	3	2	0	0	0	0	0
中 咽 頭	13	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0	2	4	3	0	0	0	0	0	0
下 咽 頭	48	0	0	0	0	0	0	0	0	0	0	0	3	3	11	10	8	4	6	2	1	0	0	0	0	0
結 腸 直 腸	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	4	7	2	2	0	0	0	0	0	0	0
食 道	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0

1984年次分では剖検症例数で39,050症例，病名コードの種類で840種類，記載施設数で471施設になっている（図4）。

§ 3. 剖検輯報の粗データの作成

剖検輯報の頻度表作成過程をまず説明する（図5）。

全国の各施設から前年に剖検した症例の所見を記述した原稿が毎年10月末にまとめて病理学会に送付される。それを病理学会で複写し，複写原稿は産業医科大と印刷屋に送付される。印刷屋ではただちに剖検輯報の病理所見を記述した「記述部分」の植字にとりかかる。複写原稿は3つに分けられ，東大，慈恵医大，産医大とでその複写原稿からのコーディングをする。コーディングされた入力用紙を磁気テープにパンチして産医大のコンピュータ（VAX 11/750）に投入した。このデータを粗データと呼ぶ。

§ 4. データチェックの概要

まず最初に行うデータチェックは，磁気テープで送られて来たデータの施設別の剖検番号と剖検症例数がすでに記録している施設別の剖検番号と剖検症例数に合致するかどうかのチェックである。このチェックによりデータのダブリや洩れを見つける。データの内容のチェックについては次に示す5段階に分けて行う（図6）。

(1) すでに報告³⁾した25種類の項目単位の論理チェック

1つの症例の項目毎に決められた条件のチェックで，途方もない年令や定義された以外の性などである。

このチェックはコンピュータでどの項目がどう言う理由でエラーになっているかまで判定出来，その項目だけ原稿と照合すれば訂正出来る。

(2) 1つの症例中の項目間の関連チェック³⁾

これは性別・病名の異常（男性の卵巣の悪性腫瘍等），年令・病名の異常（0～1才の動脈硬化症等），原発部位，組織型別の異常（胃の扁平上皮癌等）である。

このチェックはコンピュータでどの項目とどの項目とがエラー条件に合致しているかを判定する。この場合，エラーと判定された項目だけでなく，同一のグループ（複数の悪性腫瘍がある場合，原稿に記載されている原発部位，組織型の全て）の全ての項目を原稿と照合して，判断する必要がある。

(3) 不確定病名のチェック

コード設定時にいくつかのはっきりした概念以外の〔その他〕の概念に相当するコード（他の肺疾患，分類困難な疾患等）である。

(4) 誤りやすい症例のチェック

病名が同じでも意味の違う病名（骨と乳腺のページェット病の違い等）があり（14種類），それらの誤りやすい病名コードのチェックである。

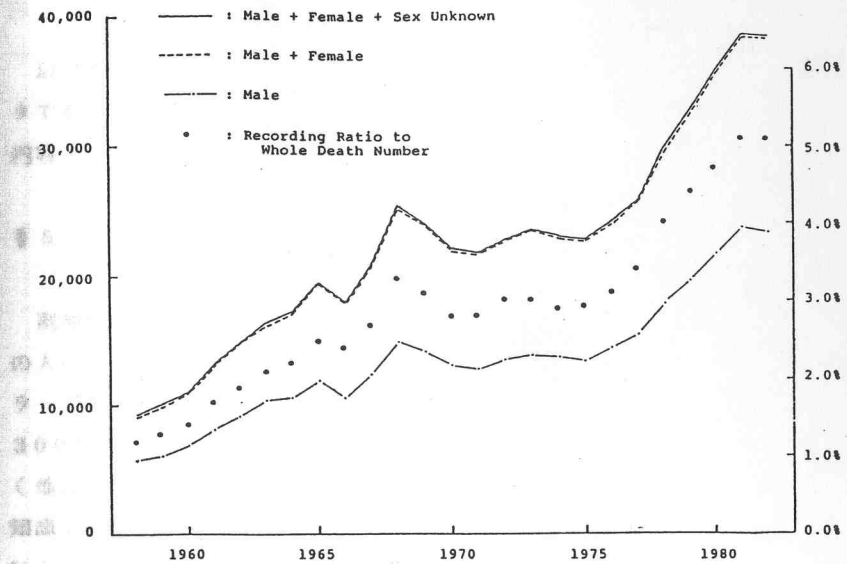


図3 剖検症例数年別推移

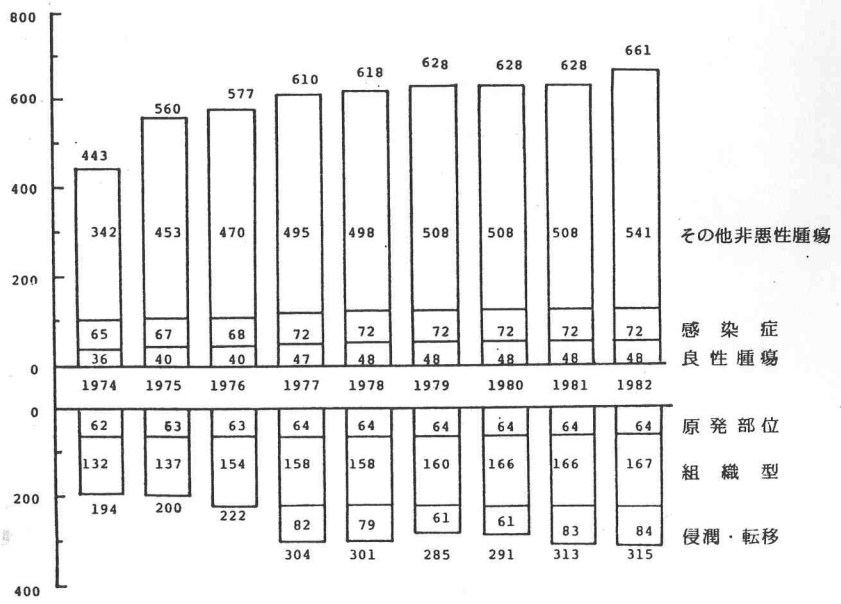


図4 剖検報記載の病変コードの種類別の年別推移

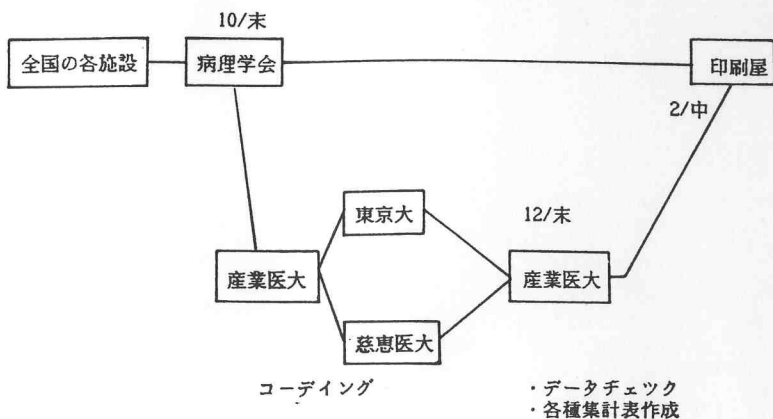


図5 剖検輯報の各種頻度表作成過程

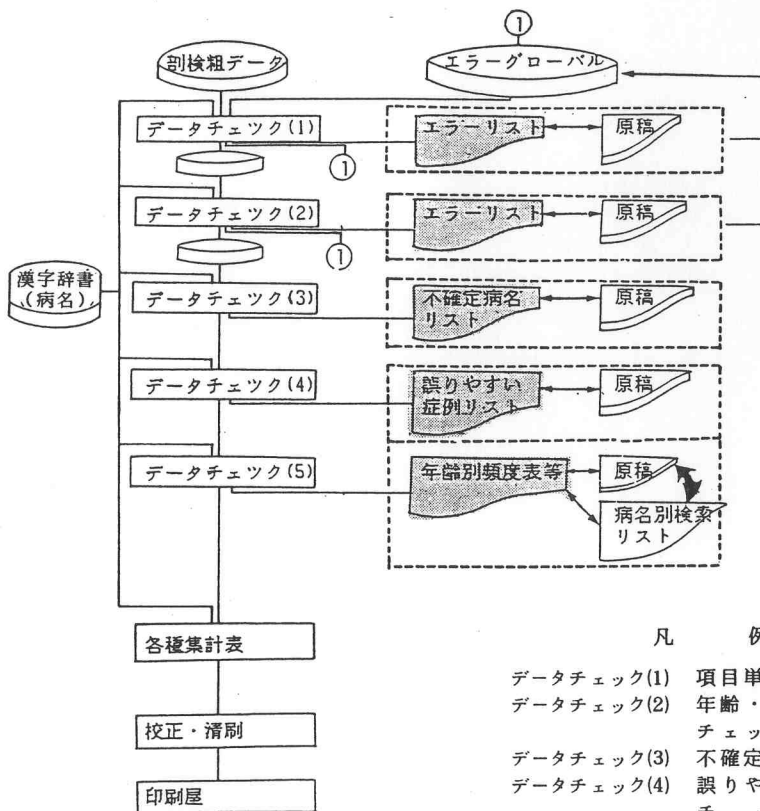


図6 頻度表作成コンピュータ処理過程

(5) 仮集計表によるチェック

年齢・病名別および原発部位・組織型別の症例数のチェックでデータ全体の集計結果から異常を見つけるためのチェックである。

以上の5段階のチェックを行っている。上記の(3)~(4)のチェックはコンピュータで該当病名を含む症例を全データの中から抽出し、病理学の専門家により全項目の内容チェックを行っている。

§ 5. データチェックにおける日本語の出力

剖検輯報の頻度表作成にゆりされた時間は、わずか1ヶ月程度で、しかも剖検輯報の入力データのうちチェックすべきデータ量の多いこと(項目単位の論理チェックエラーで約2800症例、項目間の関連チェックエラーで約400症例、不確定病名で約300症例、誤りやすい症例で約3000症例程)とエラーの解明とより正確なデータ(他の肺疾患や分類困難な疾患等)にするのに全て原稿との照合が必要であり、専門知識も必要とするため、非常に工数がかかり、データチェックに頻度表作成期間の70%を割りあてざるを得ない。

どうして、剖検輯報のコンピュータ処理では、一般のバッチ処理に比べてデータチェックに手間がかかるかの要因を人間、段取り、原稿、コンピュータの要素に分けて、特性要因図を書くとおよそ図7の如くである。

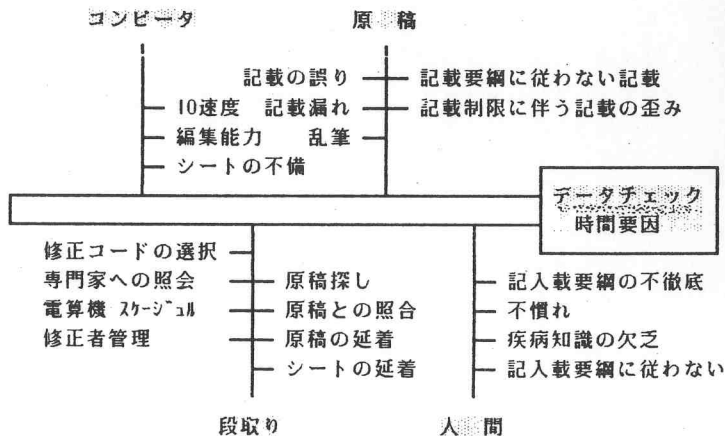


図7 データチェックの所要時間を左右する諸要因

データチェックのための段取り要素を作業順に並べ(図8)、その各作業別の所要時間比を見ると出力読み(コンピュータで出力されたチェックリスト上の病名コード、住所コード等の日本語への翻訳)に要する作業が段取り作業の約60%を占めていた(図9)。

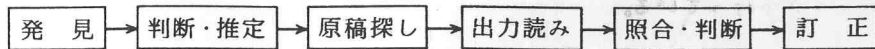


図8 チェック段取り過程

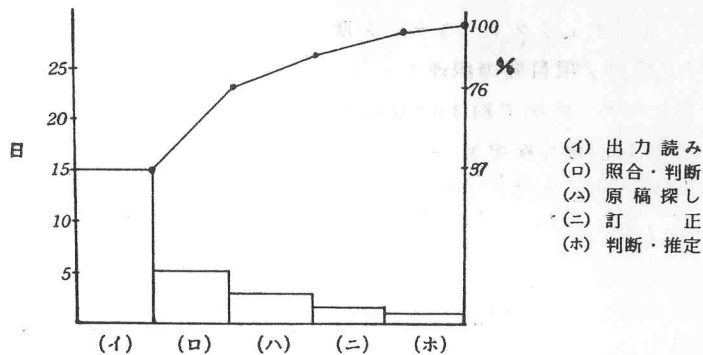


図9 チェック段取り別所要時間比

§ 6. 病名等の辞書・データの格納ならびにデータの修正方法

前述より、データチェックに当って病名コード等の日本語への翻訳がかなりの負担になっていることがわかった。そこでコンピュータでデータチェックリストを出力する時にエラーの病名コード等を日本語で出力し(図10)、14種類の集計表(表2)の全てを日本語出力した。

これらの日本語出力は病名辞書および住所辞書、施設名辞書をグローバルデータとして作り、それらの参照により処理した(図11)。

誤りやすい症例等については病名コードからただちに該当症例が検索出来る様に病名コードをノードにしたインバーテッドグローバルを作成した(図12a, 12b)。

データの更新はCRTから対話型で行うことにし、オペレータが修正したい項目内容を入力すると、その個所をサーチしてその位置をマークし、次に入力された修正内容がその部分と置き換る様にした(図13)。これにより数字と記号だけの並びのデータの中から修正すべき項目を見つけ、そこにカーソルを動かす手間が省けた。

前述の粗データをコンピュータでチェックする時はエラーデータとエラーなしのデータとを分離し、データ修正後の再チェック時のコンピュータ処理の入出力時間の減少を図った。

- 日本語化前 -

< BOKEN DATA CHECK LIST >

```

73/140060/73222/# /064    ///1///40/GG/////RR//9/M4857530845605319M/
9:4857530845605319:HI AKUSEI BYOHENSU ERROR
73/140060/73559/# /      03 /2//2///99/GG/////RR///M77887997M/
:HI AKUSEI BYOHENSU ERROR
73/140060/73566/# /059    ///1///1/40/G1519G/8143/////R8168R/01//MM/
:RINPASETSU TENI 1 JYU KAZU ERROR
73/140060/73599/# //1/45///40/GG/////RR//2/M77977790M/
SEX ERROR
73/140060/73623/# /      00 /2/////40/GG/////RR//1/M74777788M/
:SEX ERROR
1:HI AKUSEI BYOHENSU ERROR
73/140060/73796/# /071    ///1///1/40/G1878G/8903/889/01///R66R/01/1/M2509M/
GENPATSU NO.1 ERROR
  
```

- 日本語化後 -

< 剖検データ エラーリスト >

```

73/140060/73222/# /064    ///1///40/GG/////RR//9/M4857530845605319M/
140060:産業医科大学病院病院病理部 40:福岡
非悪性病変数エラー
4857:癌下性肺炎 5308:食道潰瘍 4560:食道静脈瘤NOS 5319:胃潰瘍NOS(マロリ・ワイスを含む)
73/140060/73559/# /      03 /2//2///99/GG/////RR///M77887997M/
140060:産業医科大学病院病院病理部 99:不 明
非悪性病変数エラー
7788:新生児 7997:このシートでは分類困難な症候群・疾病
73/140060/73566/# /059    ///1///1/40/G1519G/8143/////R8168R/01//MM/
140060:産業医科大学病院病院病理部 40:福岡
リンパ節転移エラー
81:胃周四 68:その他
1519:胃 8143:腺癌(低分癌)
73/140060/73599/# //1/45///40/GG/////RR//2/M77977790M/
140060:産業医科大学病院病院病理部 40:福岡
:性別エラー
7797:胎児死亡NOS 7790:浸軟児
73/140060/73623/# /      00 /2/////40/GG/////RR//1/M74777788M/
140060:産業医科大学病院病院病理部 40:福岡
:性別エラー
非悪性病変数エラー
7477:心血管系奇形NOS 7788:新生児
73/140060/73796/# /071    ///1///1/40/G1878G/8903/889/01///R66R/01/1/M2509M/
140060:産業医科大学病院病院病理部 40:福岡
原発部位エラー
1878:該当コード無し 8903:横紋筋肉腫
  
```

図 10 データチェックリストの例

^BOKD
 (1048)=マイコプラズマ感染症
 (1048,1)=52/ 17
 (1048,2)=4
 (1049)=他のスピロヘータ-感染
 (1049,1)=58/ 17 -
 (1049,2)=4
 (1370)=ペーシェット病
 (1370,1)=470/ 39
 (1370,2)=5
 (1409)=口唇
 (1409,1)=1/ 1
 (1409,2)=1
 (1599)=他の消化器
 (1599,1)=21/ 1
 (1599,2)=1
 (8073)=扁平上皮癌
 (8073,1)=7/ 4
 (8073,2)=2
 (8523)=小葉癌
 (8523,1)=43/ 5
 (8523,2)=2
 (8723)=黒色腫
 (8723,1)=77/ 7
 (8723,2)=2

^BGTBHC
 (1048)=
 (1048,8262803000824)=
 (1129)=
 (1129,8210701082095)=
 (1129,8210701082152)=
 (1139)=
 (1139,8211404003007)=
 (1139,8212302007961)=
 (1139,8251203082059)=
 (1160)=
 (1160,8210102007158)=
 (1160,8211306006790)=
 (1160,8213601002324)=
 (1173)=
 (1173,8210101082069)=
 (1173,8210101082084)=
 (1173,8210101082115)=

図 11 病名辞書グローバル

図 12a 病名インバーテッドグローバル

第一ノードは病名コード。第2ノードは「1」と「2」の二種類あり、「1」は出力順位／出力頁をその内容とする。「2」は病名コードの種類（感染症、悪性腫瘍等）をその内容とする。

^BOKEN
 (8212801082770)=82/128010/82770/##/076
 (8210101082001)=82/101010/82001/##/060
 (8210101082002)=82/101010/82002/##/ 07
 (8210101082003)=82/101010/82003/##/082
 (8210101082005)=82/101010/82005/##/003
 (8210101082006)=82/101010/82006/##/ 05
 (8210101082007)=82/101010/82007/##/047
 (8210101082008)=82/101010/82008/##/063
 (8210101082009)=82/101010/82009/##/044
 ///1///1/28/G1629G/8033/579706/02///RR/07//MM/
 ///1///01/GG////////RR//1/M5697M/
 ///2///01/GG////////RR//2/M28295709M/
 ///1///1/01/G1419G/8082/000/00///RR//1/M4929M/
 ///1///01/GG////////RR//1/M0389M/
 ///1///01/GG////////RR//1/M7477M/
 ///2///01/GG////////RR//1/M4309M/
 ///1///2/01/G15191629G/81438082////////RR//MM/
 ///1///01/GG////////RR//1/M4308M/

図 12b 剖検メイングローバル

GET
 MAINTENANCE GLOBAL ? (XXXXX) = BOKEN
 MAINTENANCE JOURNAL GLOBAL (^TEMP) IRU ? (Y/N) = N
 KEY ? (END=". " / NEXT="" / BACK="-" / START="-1" / UPPER NODE=@) = -1
 82/101010/82001/##/036 ///1///01/GG////////1/M4259M/
 OLD ->4259
 82/101010/82001/##/036 ///1///01/GG////////1/M4259M/
 ^^^^
 NEW ->4266
 82/101010/82001/##/036 ///1///01/GG////////1/M4266M/
 ----- RETRY ? (Y/NO=CR) ->
 KEY ? (END=". " / NEXT="" / BACK="-" / START="-1" / UPPER NODE=@) =
 82/101010/82002/##/056 ///1///1/01/G1919G/9433/000/00///R00R/00//MM/
 OLD ->1915
 82/101010/82002/##/056 ///1///1/01/G1919G/9433/000/00///R00R/00//MM/
 ^^^^
 NEW ->191915
 82/101010/82002/##/056 ///1///1/01/G191915G/9433/000/00///R00R/00/
 M/

図 13 グローバル更新用エディター

§ 7. 日本語化による効果

データチェック過程および集計表の日本語化により得られた効果を作業別に見ると人手によるチェック作業の時間が最も多い(図14)。剖検輯報のコンピュータ処理の場合、文部省への提出期限等が決められており、限られた時間内で処理しなければならない。剖検輯報の入力データのチェック作業は、単にエラーの訂正だけでなく、不確定病名(その他の肺疾患等)等の様に原稿と照合することにより、より正確な、精度の高いデータにすることも大切な作業である。これは専門的な知識が必要であり時間もかかる作業である。従って、チェック時間の減少はデータ精度向上にもつながり、その面の効果もある。

過程 区分	人手によるチェック作業	段取り	校正	滑刷作成	印刷(業者)
日本語化前	150秒/件 15人日	10人日	4人日	5人日	3人日
日本語化後	60秒/件 7人日	10人日	1人日	0.5人日	1人日

図14 日本語化による作業別効果

§ 8. む す び

本システムの様には大量粗データのバッチ入力の場合、データチェックに手間どり、結果を得るまでにかかなりの時間を要する。アンケート調査データの解析等の場合にも、解析部はパッケージソフトで効率よく出来るが、データチェックとその修正にはかなりの時間がかかる。この原因はデータが多くは数字だけで、もとのデータとその意味と連想性がないため修正作業に手間どる。連想性の改善には、原稿・人・コンピュータ・段取りのいわゆる Material, Man, Machine, Management の四つからのアプローチがあり得る。今回、日本語化により、このうちの段取りが飛躍的に向上した。

参 考 文 献

- 1) 浦野順文, 馬場謙介他: 日本病理剖検輯報記載事項の集計および検索。病理と臨床1, 文光堂。pp164-169, 1983。
- 2) 木村園恵, 木村一元, 馬場謙介: 日本病理剖検輯報主病変検索システムのデータの確かさの検定。第7回日本マンプロユーズズグループ大会, 1980。
- 3) Kimura, S., Baba, K. et al: Frequency Tabulation of Pathological Findings of more than 23,000 Autopsy Cases of Annual Collection through Japan. Proceeding of the 1979 MUMPS Users Group Meeting. 126-133, 1979.
- 4) 馬場謙介, 実藤準人, 吉村健清: 日本病理剖検輯報の癌疫学への活用のために。倉恒匡徳編: 職業がん - 疫学的アプローチ。條原出版, 東京。pp171-188, 1984。

研究用患者データ管理システム (CUIPDS) における時系列データ処理

○ 本多正幸 ・ 北村嘉章 ・ 里村洋一

千葉大学医学部附属病院医療情報部
千葉市亥鼻1-8-1

患者データを解析するためには、医療データ、病歴データの多様性に対応できるように、柔軟な設計のシステムが要求される。その一つとして、システムには時系列データを取り扱える機能が必要である。本報告では、我々の開発した“研究用患者データ管理システム (CUIPDS)”の機能の中で、時系列データの処理に関する部分について報告する。時系列データのデータ構造は、3次元構造を考える必要があり、処理に関しても独特の処理機能を必要とする。

§ 1. はじめに

各種医療機関において、医療統計や診療データの解析の要求はますます増大し、いくつかの医療機関及び医療関係者において医療統計解析システムの構築が報告されている。千葉大病院においても医療データの編集、解析を目的とし、研究者別にデータ管理を行う、個人別データ管理システム、すなわち研究用患者データ管理システム (「CUIPDS (Chiba University Integrated Patient Data System) 」) を設計開発し、82年の秋より、運用中である。

本システムでは、病歴データ、医療データの多様性に対応するために、データ導出機能としてデータ項目 (変数) 指定のための“Data Dictionary”や症例範囲指定のための“自動ルーチン作成機能を用いた柔軟な編集処理機能を準備している。(これらの機能に関しては昨年のMUG大会報告集を参照して戴きたい。)本報告では、研究用患者データ管理システムのいくつかの機能および特徴のうち、特に時系列データ (time-dependent data) の取り扱いに関するものについて報告する。

なお、本システムはMUMPS言語ですべて書かれている。

§ 2. データ構造について

本システムでは、時系列データを取り扱う場合には3次元構造 (“症例”, “項目”, “時間”) のファイルを考え、その他の場合は2次元構造 (“症例”, “項目”) を考えることにした。ユーザーは主にファイル単位で編集解析処理を行うわけであるが、同一ファイル上に2次元データと3次元データが共存したファイルを取り扱うことになる。時系列データとその他の (2次元) データとを区別するために、項目名の頭に特殊記号 “#” を付加することにした。(例えば、“#GOT”) 具体的なデータ構造は次のようになる。ただし、第一ノード (DCOD) はユーザー識別番号であり、第二ノード (FILE) はファイル名である。(↑REPDはグローバル名である。)

2次元構造: ↑REPD (DCOD, FILE, “症例番号”, “項目番号”) = data

3次元構造: ↑REPD (DCOD, FILE, “症例番号”, “項目番号”, “時系列番号”) = data

右辺のdataに関しては、5種類 [categorical type, numerical type, date type, literal type, code type (ICDコード, ICDの手術コード, SNOPコード)] の属性を付加して病歴データの多様性に対応している。3次元構造の“時系列番号” に関して同様の理由から次の3種類のタイプを準備している。

①月日依存のタイプ (YYMMDD形式…例、84年11月23日=>841123)

②時間依存のタイプ (HHH:MM形式…例、14時15分=>14:15,

MMM:SS形式…例、175分30秒=>175:30)

③その他 (sequential no.形式)

また、本システムではデータを一意に指定し、ファイル管理を機能的にするためにユーザーが登録したデータ名をそのままシステム中で使用できるようにデータの辞書を作成しておく。このData Dictionaryには、グローバル定義を行い、グローバルの形式をグローバル名、各サフィックス名、レコードの中のディスクリミネータとデータコード名をMUMPSの記述形式そのままに入力し、システムはこれを解釈して各データコード名に対するデータの概念的位置を記録する。またDictionaryに記録されるパラメータはデータのタイプ(categorical、numerical、etc)及びタイプ別に必要な情報を記録しておき、時系列データについては、“時系列番号”のタイプ(日付依存、時間依存等)を記録しておく。

以下はDictionaryの内容である。(GDIPは辞書のグローバル名で、REPDはデータのグローバル名であり、“ID”は症例番号を表わす。)

- ↑GDIP(DCOD, FILE, “項目名”)
 - = “項目名 ¥ “REPD” ¥ ↑DCOD ↑FILE ↑ “ID” ↑ “項目番号”
- ↑GDIP(DCOD, FILE, “項目名”, 1) …numerical typeの場合に、論理的な最大値、最小値、グラフ表示の際のパラメータ等の情報
- ↑GDIP(DCOD, FILE, “項目名”, 2) …categorical typeの場合に、カテゴリー名の情報
- ↑GDIP(DCOD, FILE, “項目名”, 3) …データのタイプ(numerical, categorical, date, literal, code)の情報
- ↑GDIP(DCOD, FILE, “項目名”, 4) …code typeの場合に、コードの種類(ICDコード、ICDの手術コード、SNOPコード)の情報
- ↑GDIP(DCOD, FILE, “項目名”, 5) …項目名に“#”マークが付いている、時系列データのタイプ(日付依存、時間依存、その他)の情報

このようにして登録したファイルの項目を、ユーザーはファイル管理表によって確認することができる。(表1.参照) 例えば、「DIAGNOSIS」は非時系列のliteral typeのデータ、「#GOT」はYYMMDD typeの時系列でnumeric typeのデータである。なお、numeric typeの項目では入力チェック用の最大値、最小値や、グラフ用のパラメータを、categorical typeの項目では各カテゴリーの表現名を表示している。

§3. データ処理について

時系列データの取り扱いも、その他の2次元データとほとんど同様に扱うことができる。例えば、項目の定義やデータの入力、修正、削除等の処理は同じ形式で処理することができる。しかしデータの表示(リスト、グラフ)や編集処理には独特の処理が必要である。

[表 1]

***** PRIVATE FILE MANAGEMENT TABLE (2) ***** 84.11.01

ITEMS OF "LIVER 1"

NO)	NAME	TYPE	PARAMETERS
1)	DIAGNOSIS	L	
2)	WEIGHT	N	MIN=0 MAX=150 SPAN=20 0 POINT=0 UNIT=KG.
3)	#GOT	N-YYMMDD	MIN=0 MAX=5000 SPAN=250 0 POINT=0 UNIT=U
4)	#GPT	N-YYMMDD	MIN=0 MAX=10000 SPAN=400 0 POINT=-200 UNIT=U
5)	#LDH	N-YYMMDD	MIN=0 MAX=50000 SPAN=100 0 POINT=50 UNIT=U
6)	#CRP	0-YYMMDD	1=(-) 2=(+-) 3=(+) 4=(++) 5=(+++)
7)	AFP	N	MIN=0 MAX=100000 SPAN=500 0 POINT=0 UNIT=
8)	HBS-AG	0	1=(-) 2=(+-) 3=(+) 4=(++)
9)	HBS-AB	0	1=(-) 2=(+-) 3=(+) 4=(++)
10)	#OGTT(B)	N-HHH:MM	MIN=0 MAX=1000 SPAN=300 0 POINT=0 UNIT=MG/DL
11)	#OGTT(U)	N-HHH:MM	MIN=0 MAX=1000 SPAN=300 0 POINT=0 UNIT=MG/DL

(ii) 編集処理

次の2種類の処理を準備している。

①各症例において、指定された期間における基本統計量の算出及び結果の保存

データタイプでnumerical typeのデータを対象とする。期間の指定に関しては、直接端末に2つの時点（固定時点）を入力する方法で指定するが、時系列番号が日付依存タイプの場合にはdate type のデータ項目名でも指定できる。

項目名で指定した場合には症例によって異なる特定の日付（生年月日や初診日、手術日、死亡日等）が指定でき、処理の幅が広がる。また、導出する基本統計量としては、平均、標準偏差、最大値、最大値に先に到達した時点、最小値、最小値に先に到達した時点の6種類である。これらの情報をリストアップし、更に2次元データとしての新たな項目名をつけファイルに保存できる機能がある。（表3. 参照）

②各症例において、ある時点から、ある条件を満たすまでに要する期間の算出と保存

条件に用いる項目を指定し、その項目の“時系列番号”のタイプに応じてスタート時点を端末より入力する。①と同様に日付依存タイプの場合にはdate type の項目名でも指定できる。先に指定した項目名を用いて条件の設定をする。条件はユーザー自身で論理式を作成して、その設定条件を初めて満足する時点を検出し、スタート時点との期間を算出する。（例えば、条件式としては、numerical typeの項目を指定した場合には、“ある一定値を初めて越える、または下回る”、“ある範囲を越える”、“逆に”ある範囲の中に初めて含まれる”等が考えられる。）さらに、算出された期間、及び初めて設定条件を満たした時点を2次元データとして新たな項目名を付けファイルに保存できる機能がある。

【 表 3 】

***** LIST OF STATISTICS (SUMMARY OF TIME-DEPENDENT DATA) *****

FILE NAME=LIVER 1
DATA NAME=#GOT

*** 【カンパ センク ノ ショウケン (PATIENT SELECTION)】

ショウケン シキ (CONDITION) (A=3)

WHERE A=HBS-AG

*** POINTED TERM BETWEEN (810601 - 810610)

NO	ID	SIZE	MEAN	SD	MAX (TIME)	MIN (TIME)
1	2	6	125.66	97.222	224 810604	26 810610
2	3	1	105	***	105 810606	105 810606
3	4	5	95.4	81.25	195 810601	50 810609

参 考 文 献

- 1) 神沼、他：医療の為の統計パッケージSPMS利用の手引、東京都臨床研（1980）
- 2) 池田、他：統計データベースの問題点と課題、日本ME学会第2回病歴データベース研究会資料集（1982）
- 3) 本多、里村：マンプスを用いた会話型データ解析システム（CUIPDS）とその時系列データへの拡張、第23回日本ME学会論文集（1984）

MUMPSよりのVMS及びDSM資源の利用

田久 浩志 産業医科大学振動室
馬場 謙介 産業医科大学第二病理学教室

北九州市八幡西区医生ヶ丘1-1

概 要

MUMPSは、生産性の高いプログラミング言語である。しかも、みずからMUMPSを使いやすく出来る柔軟な言語である。MUMPSをより便利で使いやすくすることが方々で努力されている。産業医科大学ではそのため、Supervisory ProgramとしてMUMPS Speed Programing Package (MSPP)を開発し、プログラマにより良いプログラミング環境を提供している。MSPPはX^MSPPで起動し、“MUMPSにひと皮かぶせた”状態で動く。MSPPはZDEBUGを用いたTRACER、ZCOMPILEを用いたERROR CHECKER、\$ZCALL(%SPAWN)を用いたEDITORを持つ。またこの他に、MUMPSの命令は勿論のこと、VMSが持っている、DCL (DEC Command Language) 命令が発行でき、直接にルーチン行の入力もできる。

1. はじめに

プログラムの生産性向上は、業務で頻繁にプログラムを開発する者に課せられた大きなテーマの一つとしてここ数年来急激に注目をあびるようになった(文献1)。このことは、研究用にVAX-DSMを使用している我々とても例外ではない。現在のVAX-DSMのシステムは、\$ZCALL関数を使用してDSMの中よりVMSの資源を利用することもできる強力なシステムであるが、筆者らのようなプログラム開発担当者にとっては、この環境でのプログラム生産性は決して良いものであるとは言い難い。何故ならば、プログラム開発に頻繁に使用するエディタなどは、一々そのプログラムを起動しなければならず、また、普段VMSの豊富な資源をDCL命令によって使用して統計計算、信号処理などを行っているため、DSMの中にもついでDCL命令を入力したくなるからである。これらの経験より、DSMより、自由にDSM,VMSの資源を利用する方法の開発が望まれた。

2. MSPPとは

MUMPS Speed Programing Package (MSPP)とは、DSM,VMSの資源を自由に利用するため人間の入力を識別し、的確にDSMやVMSへ人間の入力を与えるための一種のSupervisory Programである。これにより、一般に開放されているDSMの資源はもちろん、豊富にあるVMSの資源を全てMUMPSの世界より使用し、プログラムの生産性が向上すると考えられる。

資源といっても色々な資源が存在するが、まずプログラマが日常の業務に必要なソフトウェア資源を利用しやすくすることよりがけられた。この日常の業務に必要なプログラムとして、Editor,Tracer,error Checkerを選び各々E,T,Cの一字が^MSPPの命令で使用できるようにした。この実現にあたってはメーカーが供給しているユーティリティを改良して使用した。Editorに関しては、\$ZCALL(SPAWN)を用いてEDTを起動する^%EDTを、TracerはZDEBUGを使用した^%TRACEを、そしてerror CheckerはZCOMPILEを使用して、MUMPSの中間言語よりエラーチェックを行なう^%ERRCHKをそれぞれ改良して使用した。なお、MSPPの終了にはQを、ヘルプメッセージを見るには?を命令として発行する。

E,T,CのほかにもMUMPSの命令や、DCLの命令がX^MSPPの命令として発行でき、もちろん、Tabを用いて直接ルーチン行の入力もできる。この場合MUMPSの命令は、XECUTE命令で、DCLの命令は\$ZCALL(SPAWN)で、そして直接のルーチン行の記述に対してはZINSERT命令で対処した。

X ^MSPP への命令に関して、DSM,VMS の命令は全て受け付け、省略形、非省略形ともに発行できるようにし、使用者の入力に制限を付けて無用な混乱を起すことを避けた。ただし、KILL命令にだけは制限を付けた。つまり、KILL命令を用いてすべての変数もしくはシステム変数、および MSPP を記述しているグローバル変数を消去しようとすることは禁止した。

MSPP の応答を早くするため、命令解釈の順序に優先順位をつけた。MSPP はその命令の使用頻度を考慮して、ルーチン行の入力 > MSPPの命令 > MUMPS への命令 > VMSへの命令 の順とした。この結果、MUMPS も DCL も同じ名前の命令である、READ,WRITE,OPEN,CLOSE,IFに関しては、MUMPS の命令としての解釈が優先されることになった。

メッセージの出力方法に対しては少々の工夫をこらした。つまり、一回エラーが発生してもユーザーのタイプミスと考え、安易にヘルプメッセージを表示することは避けた。ただし二回連続してエラーが発生した場合は、使用者が使用方法を知らないと判断してヘルプメッセージを表示するようにした。こうすることにより、たくさんのヘルプメッセージによってエラーメッセージが画面上端より消えてしまうことを避けた。

4. 使用例

>X ^MSPP	DSM より MSPP を起動する。
*	MSPPのプロンプト*が表示される。
*A TA SET A=1	通常のDSM でのようにルーチンを作成できる。
* TA WRITE A,!	
*DO	ルーチンを実行する。
1	実行結果が表示される。
*E	EDITORを起動する。(詳細は略す)
*DIR	VMS の DIR命令を実行する。
Directory SYS\$SYSDEVICE:(TAQ)	ディレクトリが表示される。
MT.GBL;7 T.DAT;1	
*DEL T.DAT;1	VMS の DELETE 命令によりファイルを消去する。
*Q	MSPPを終了して、DSM のモードにもどる。
>	

5. むすび

MSPP は可能なかぎり、人間の入力を解釈して理にかなった動きをするようにしたため、ユーザーに非常に使いやすいプログラム環境を提供した。特に DSMの世界より簡単に VMSの資源を利用できるようにしたため、デバイスの初期化、マウント、ファイルの表示、印刷、消去などに威力を発揮している。

この MSPP を使用した時と、そうで無い時でのプログラム開発効率の比較は正確には表現できないが、個人的感覚として、開発時間が 20-30% は短縮されるように感じている。しかし、すべて良いことばかりがあるわけではなく、MSPP の中で既存のユーティリティを起動した時に、システム変数の衝突の発生を何回か経験した。この苦い経験をもとに、決して変数の衝突の発生しない方法として、自前のスタックポイントを利用するネスティング可能な Small Concept (文献 2) のアイデアが生まれたのである。

参考文献

- 1) 国友義久: 効果的プログラム開発技法 近代科学社
- 2) 馬場 謙介 田久 浩志: Small concept 第11回MUG 学術大会にて発表予定

「システム デザイン 2」

座長 河村 徹郎
(大阪成人病センター)

SMALL CONCEPT

馬場 謙介(1) 田久 浩志(2)

(1) 産業医科大学第二病理学教室 (2) 産業医科大学振動研究室

(1)(2) 北九州市八幡西区医生ヶ丘 1-1

要 約

「汎応用の処理・演算の単位」と定義できる「Small concept」の概念を解説・紹介し、Small concept の実例を示した。Small concept の概念をシステム化に導入すると、複雑・高度な処理が簡素化できることを述べた。Small concept は、ルーチンやユーティリティよりはるかに流通性が優れていること、等の多くの利点について述べた。Small concept の MUG による促進と利用技術の MUG による蓄積は、MUG らしい活動の一つであろうことについても触れた。

はじめに

MUMPS は、「装備に拘らず、MUMPS で書かれたルーチンであれば、どんなメーカーのどんな装備の MUMPS にも移送できる」ことを、重要な行動規範として来た。他の言語の貧弱な移送に苦しめられていた標準 MUMPS 出現当時、一般ユーザーは、「ユートピアの到来」とさえ受け取った。装備は、この規範に添って多大の努力を傾けてきてくれた。そのかいあって、MUMPS は「移送性の高い言語」との評価を得るに至るまでにはなかった。しかし、現実には、他人が作ったルーチンを、自分のシステムに組み込むことがどれだけ実現しているであろうか。当初意としたユートピアが本当に実現しているであろうか。ほど遠い感がする。どうしてであろうか。その原因として、(1) あるルーチンの一部に、優れた機能が組み込まれていても、そのルーチン全体が自分の目的に合った設計になっていなければ、せっかくの優れた機能も自分のシステムとは無縁であること、(2) 言語上、移送できても、システム構成、システムの規模、情報環境、人的環境の差によって、事実上実際に移送できないこと、(3) ルーチン程大きくなると、著作権に強く係って、現場の技術者の意のままに移送はできないこと、等々、が上げられる。これらは、すべて、「ルーチンと云う単位」が、大きすぎるからである。「ルーチンよりは小さく」かつ「ルーチンより柔軟で、汎応用性の高い」"Small concept"と云う概念を、積極的に展開すると、上述の事情が好転すると予想される。

概 念

Small concept に対する明確な定義はないが、「汎応用の処理・演算単位」と云える。"Subroutine"と似た概念であるが、汎応用性が基本条件である点が、"Subroutine"と趣を異にする。"Subroutine"は、あくまで、ある Routine が先ずあって、そのルーチンの記述に都合良くまとめられた「処理・演算単位」であるのに対し、Small concept は、どのルーチンでも広く"応用"されることを意図していることが重要で、これが概念の根幹を成している。"Utility"とも似た概念であるが、Utility が広く単独で使えること、つまり「汎用性」を使命とするのに対し、Small concept は、広くルーチンの中で

応用されること、つまり「汎応用性」を使命としている。従って、Utility には、それ単独でまともな仕事を完成することが必然的に要請されるが、Small concept は、それ単独でまともな仕事を完成するものでなければならないと云う束縛からは解放される。Utility では、汎用性を増す為に「パラメーターを増やしたり、面倒な操作手順を要求するしたり」せざるを得なくなってしまうのに対し、Small concept では、応用し易さを追及しているのが必然的に、記述規則は普遍的かつ簡明である。Small concept は、普通は、Utility より小さいが、本質的なことは、コーディングサイズの大きさをなく、「簡便に応用できる程度にしか機能を拡大しない」ところにある。

実 例

無意識の内に慣用している Small concept の例

例 1) `^ZP="0 1 U 1 W # ZP W ## C 1"`

例 2) `^ZS="ZS @$P($T(1)," """,1)"`

例 3) `^ZP="0 ""SPOOL"" U ""SPOOL"" W # ZP W ## C ""SPOOL"":SPOOL"`

以上3例は、初心者にも作れる簡単な、Small concept の例である。これらは、Small concept が、ルーチンに組み込まれることはまずない。これに対して、次の例 4は、普通、ルーチンの中で応用される。

例 4) `SET %HD=X DO ^%HD SET Y=%HD`

この例は、16進数から、10進数に変換する "SET-DO-SET ユーティリティ" で、ルーチンにしばしば組み込まれる Small concept である。この場合、Small concept 単独では、殆どなんの "まともな仕事" をしてくれない。

や、複雑な例

UOEH Small Concept Project Team が推進しているいくつかの実例を次に示す。ここでは、紙面の関係で、大部分の例についてはルーチン内での指定方法を示すに留める。

例 5) カラーディスプレイの色制御：

XECUTE ^%O で色制御 Small concept を起動しておいて、例えば、
WRITE %OL を発行すると、その後は、LEMON色（黄色）で印字
WRITE %OA を発行すると、その後は、APPLE色（赤色）で印字され、
WRITE %ON を発行すると、その後は、NORMAL 色（白色）印字にもどる。

例 6) グラフィック制御：

グラフィック制御の Small concept は、日本電気 PC-9800 / CP/M-86 の環境で、エスケープシーケンスによるグラフィック制御機能の付加を前提にしている。

XECUTE ^%O でグラフィック制御 Small concept を起動しておいて、つぎに例えば、
WRITE %OE,"100,100,200,300,2,2",!
を発行すると、点座標(100,100)に、横 200 点の径、縦 300点の楕円を、赤で描き、赤で埋め；
WRITE %OB,"100,100,200,200,2,2",!
を発行すると、点座標(100,100)と、点座標(200,200)を対角点とする、矩形を赤で描き、赤で埋める。

WRITE %OQ を発行すると、グラフィック画面がクリアされる。

例 7) 文字列編集 :

文字列編集 Small concept は、任意の変数の値を編集する Small concept である。変数 %ULに問題の文字列値を渡し、「XECUTE ^%UL」を発行すると、スクリーンの下部の所定の場所に問題の文字列値が表示され、表示された文字列値の範囲をカーソルを走査して、目的の場所で文字を挿入したり消去したりできる。挿入モードで、挿入キーが打たれると、漢字入力モードになり、(グローバル変数から)指定された読みのレター文字(2バイト文字)が呼ばれ、このなから文字を選択して、元の位置に選んだ文字を挿入できる。編集が済んだ値を、%UL に設定して、当該 Small concept を終了する。

バイピングとネスティング

多数の Small concept を並列に組み合わせて使うことを考えると、Small concept の受渡し変数は、1 つであることが望ましい。例えば、この受渡しの変数を、「%」とすると、例えば、「SET %=X DO ^%HD,^%DO SET YZ=%」で、10 進数を、8 進数に変換できる。Small concept の受渡し変数を、% と約束しておく、システムを超えて、% によるバイピングが完成できる。

前述の例 7) で示したように、Small concept が Small concept を呼んで一つの新しい Small concept を構築することがおこる。その場合、Small concept 内で使用される変数名の第 1 添字の使用の工夫、スタック変数の工夫によって、変数の重複使用による混乱を統一的に回避できる。これらの工夫については、後述する。

自分で、自分と呼んだり、自分を他の Small concept に呼ばせても、後述する局所変数の採用によって変数の重複使用による変数の混乱は生じない。スタックの管理、局所変数の採用に加えて、(1) 他ルーチン等による Small concept の変更禁止方策、(2) 無限ループ予防策、を実施することによって、実質的にリエントラントな Small concept がつくれ、リカーシブルコールが、可能になる。

Small Concept の構造

バイピングとネスティングを考慮した Small concept の仕様を説明する。まず、変数の使い分けについて述べ、ネスティングを前提とする場合とネスティングを前提としない場合に分けて解説する。

変数の使い分け

(1) Localized 変数 :

Small concept の枠を越えて使用するローカル変数を、「Localized 変数」又は、「限局変数」と呼ぶ。英大文字か %英大文字ではじまる変数を、限局変数として使用し、タック値を第 1 添字に持たせる。(限局変数が、限局変数として正しく機能するには、第 1 添字が正しく設定されていなければならない。例えば、英小文字で始まる非配列変数は、真の限局変数たり得ない。)

(2) General 変数 :

Small concept の枠を越えて使用する変数を、「Generalized 変数」又は、「一般変数」と呼ぶ。限局変数以外の変数を、一般変数として使用する。

(3) 受渡し変数 :

% 1 文字だけの変数を、受渡し変数とする。受渡し変数は、Small concept における値の受け口の変数として使用し、Small concept の結果のルーチンへの渡し口の変数と

しても使用する。(受渡し変数は、受け口と渡し口で本来の意味で用いられるほか、後述のごとく、受渡しに関係ない Small concept のメインの処理部分では、スタック値を持つ変数としても用いられる。)

(4) スタック変数:

スタック変数は、Small concept に於ける、システム変数の役割を果たす。変数 %9 をスタック変数として使用する。

(付) 受渡し変数の値の構造:

受渡し変数の値を、「受渡し引数」と呼ぶことができる。受渡し引数の最大長は、255 バイトである。「受渡し引数」は、「,」を区切り文字として、複数の「副引数」で構成することができる。副引数は、更に、「:」を区分文字として、複数の「副々引数」で構成することができる。

ネスティングを前提とする場合

1. Small concept に於ける開始処理の定式:

Small concept が起動されたときに、はじめに行なわれる処理を定式化すると、次のごとくである。

- (1) 受渡し変数が未定義なら、任意の処理をする。 IF '%D(%) 任意の処理
- (2) 受渡し変数の値を、任意の限局変数に渡す。 SET:%D(%)=1 任意の限局変数=%
- (3) スタック変数が未定義なら、スタック変数に、1 を設定する。 SET:'%D(%9) %9=1
- (4) スタック変数に、1 を加算する。 SET %9=%9+1
- (5) 受渡し変数にスタック変数値を設定する。 SET %=%9

最後の処理は、限局変数の記述を最も短縮する配慮による処置である。

2. Small concept のメインの処理:

Small concept のメインの処理では、(1) 局所変数と一般変数を使い分けること、(2) Small concept に於ける開始処理で、% が事実上、スタック値になっていること、の2点を念頭に置けば、任意のネスティング化したコーディングができる。

3. Small concept に於ける終了処理の定式:

Small concept を終了するときに、行なわれる処理を定式化すると、次のごとくである。

- (1) 受渡し変数を持つスタック値を、スタック変数に設定する。 SET %9=%
- (2) スタック変数から、1 を減じて、スタック変数に設定する。 SET %9=%9-1
- (3) ルーチンに渡す値があれば、その値を受渡し変数に設定する。 SET %=結果
- (4) 使用した限局変数を、すべて KILL する。 KILL 使用した全ての限局変数
- (5) 必要ならば使用した一般変数を、KILL する。

ネスティングを前提としない場合

1. Small concept に於ける開始処理の定式:

受渡し変数が未定義なら、任意の処理をする。 IF '%D(%) 任意の処理

2. Small concept の実質部分の処理:

Small concept の実質部分の処理では、一般変数だけで書く。

3. Small concept に於ける終了処理の定式:

ルーチンに渡す値があれば、その値を受渡し変数に設定する。 SET %=結果
必要ならば使用した一般変数を、KILL する。

Small concept の拡張

Small concept の目的は、ユーザールーチンを作るときの“道具”を提供することである。しかし、Small concept の思想自体は、ユーザーによるシステム構築にも適用できる。この方法の導入で最も効果的なことは、通常の方法で起こる「ルーチンないサブルーチンの間での変数の混用」によるトラブルが回避できることである。Small concept の思想を、ユーザーによるシステム構築に体系的に取り入れたシステムの構築が進められている。

一々、変数に、「(%)」を付けることは、はなはだ煩わしい。これが唯一の欠点である。この欠点の改善は、MUMPS の言語規定の骨格の変更を伴わない限り、不可能と考えられがちであった。その為の言語規定の変更は、切望されてきた。事実、ローカル変数に「"local"」、"global"」の別を区別させる案が、これまでに何度となく公式の場で検討されてきたが却下されている。

後に述べる結論を引き出す為に、もう一つの注目しなければならない重要な思想がある。それは、変数名に小文字を使用する件に関するこれまでの議論に現われている思想である。変数名に小文字を使用することを移送規定で否と規定されている真の理由を、「小文字があつかえない装置がある」こと、「変数テーブルの設計に難点がある場合もありうる」ことに求めることは、(現在の技術水準に照らしてみても、)到底承服できないことである。真の理由は、“小文字変数”に装置者が定義できる余地を残すことを意識しているものと受け取らざるを得ない。

一見別ものにみえる "local"・"global" の問題と、変数名の小文字の問題は、実は、深く関わっている。もし、ある装置者が、「小文字変数は、\$zstack 特殊変数で場合分けされる」と定義して、小文字変数を装備したら、どの規定に違反するであろうか。該当する規定は、見当たらない。\$zstack 特殊変数が、標準採用されて、\$stack 特殊変数となるまでは、暫定的に \$zstack 特殊変数を装備して、小文字変数を上記のように定義することは、装置者に委ねられている。この形の小文字変数を、改めて“局変数”と呼ぶことにする。左に、局変数、右に前述の局所変数を並べて、その意味を例示すると、

u	u(%9)
a("X")	a(%9,"X")

である。(WRITE 命令での出力の形式は、装置者によって様々であろうが)、局変数を左のごとく書いたとき、右の局所変数の意味に定義することは、移送基準の思想に照らし合わせて、装置者に委ねられていると解釈するのが妥当であるとせざるを得ない。

む す び

多くステップを要する処理を、出入り口のはっきりした一つの“concept”にまとめて上げて、事象を単純化することが、昨今、人口頭脳分野などで盛んに取り沙汰されている。MUMPS は、これを実現するのに適した機能を持ちながら、Small concept の思想に注目する者はいなかった。Small concept の思想とその効用について述べ、MUMPS 上で組織的にこれを扱う具体的な方法を示した。更に、Small concept は、あるユーザーが開発したよい利用技術を、ユーザー間で共用するのに適した単位であることを強調し、Small concept は、これをユーザー間の共有財産として、MUG に蓄積する単位としても適当であるとの見解を述べた。MUMPS のユートピアの到来を目指して、Small concept の思想が、MUG で結実することを念願する。

謝 辞

MUMPS への Small concept の導入の切っ掛けを与え、貴重な御教授を下さった、長崎大学教育学部の西岡幸一助教授に、深謝します。

この研究の一部は、文部省試験研究助成金(58870028)、厚生省特定疾患調査研究助成金による。

参 考 文 献

Kernighan, B.W. & Plauger, P.J. 著, 「Software Tools」

木村 泉訳: 「ソフトウェア作法」, 共立出版

Winston, P.H. 著, 「Artificial Intelligence」

長尾 真・白井 良明共訳: 「人工知能」, 培風館

大量データのバッチ処理

田久浩志 産業医科大学振動室
馬場謙介 産業医科大学第二病理学教室

北九州市八幡西区医生ヶ丘1-1

概 要

日本病理学会は、1959年より剖検データを全国的規模で収集し、1979年度剖検分より、剖検データの集計表を電子計算機で作成する様にした。この電子計算機による集計表作成の作業は過去10年間にわたり三施設で別個に行なわれてきた。今回この別々のシステムが持っている合計30万件余のデータを一つのデータベースに統合する仕事を経験した。このように大規模のバッチ処理をするにあたって 種々のトラブルを回避し円滑な処理を行なうために講じられた方策の一部を報告する。

1. はじめに

貴重な病理剖検データを互いに利用する目的で、1959年より日本病理学会は剖検データを全国規模で収集し、日本病理剖検輯報として毎年発行する登録制度を発足させた。現在では、104の医学部、医科大学と383の病院が参加し、年間四万件に近い症例が集められている。日本病理剖検輯報に収録された剖検例は年毎に疾病別に集計され剖検輯報の巻末にその結果が表1に示すような集計表として記載されてきた。この集計表を作成するための作業は、1974年から電子計算機化され、集計表作成のデータは年々保存されてきた。

電子計算機による集計表作成の作業は1974年から1978年は、主として独協医科大学が(参考文献1)、1979年より1981年にかけては神戸大学総合情報処理センターが、1982年と1983年のデータに関しては産業医科大学が(参考文献2)が担当した。これらの異なるシステムのデータを合計すると全体で30万件もの大量のデータが存在することになる。しかし、これらのデータは現段階ではデータの内部書式がまちまちである。事業が開始して10年になるのを機に新規に共通の書式を設定して、検索の便を考慮したデータベースを生成することになった。

ここではデータの形式が不統一で、かつ大規模なデータをすい上げ、あらたに統一したデータベースを作成するために我々がとった方法について述べる。トラブル回避のために講じた諸方策について重点を置いて報告する。

2. 採用した書式について

各症例データは全体としては可変長であるが、可変長データの前の部分に固定長部分を置いた。データの前半の固定長部分は性別、年齢などの必ずデータが存在するアイテムより成り、後半の可変長部分に症例によって数が異なる病変データを入れた。固定長部分のデータの中でも統計計算に使用される年齢、月令等の項目に関しては項

目にデータ存在する時は数字の頭に0をうめた。年令、月令等の項目で、空白部分が生じうるが、VMSがそれらを圧縮してくれる。

(データを目で見るとき) 視覚的に、(データを処理する上で) 論理的に優れたデータ形式にするため、可変長部分には4種類のデリミッタ文字を使用して、データを一種の階層構造にした。視覚的にどのようなデリミッタをより上位にしたらよいかについては事前に小さい実験を行った。その結果、悪性腫瘍病変と非悪性病変を区別するデリミッタとして\$マークが、個々の病変を区切るデリミッタとしては@マークが最適であることがわかり、これを採用した。しかしながら、データの検索を単純にするにはこのデリミッタにさらに1つデリミッタを増した法が良いかもしれないが、ここではディスク効率を考慮して病変毎にデリミッタをつけることはしなかった。(データ構造の詳細については表2より表5に、病変グローバルの実例については図1に示す)

病変グローバルとは別に、将来の検索の便を考慮して、病変グローバルの第一ノード、第二ノード、および病変グローバルデータの中の4桁コードより逆ファイルを生成了。逆ファイルの例とその構造を図1と表5に示す。

3. 処理にあたっての対策

いかなる場合でも、用意周到の準備をして、処理にあたるのが望ましいが、大規模データのバッチ処理を行なう時には、これをおこたると、莫大な時間、資源、経費の浪費に直接つながる。それらの遅延要因は、おおきく分けてデータに起因する遅延要因、時間的遅延要因、物理的遅延要因の3つに分類できる。今回の処理を行なうにあたり、どのような対策を事前に行なったかについて、上記の3つに分けて以下に述べる。

A データに起因する遅延要因に対する対策

データ構造の検査とプログラムのテスト： 実際の処理をする前にデータ構造の検査が必要となった。何故ならば、前もって各施設のデータに関するドキュメントは入手していても、同一施設でも年度によって、内部のデータ構造が微妙に異なっていることがあるからである。しかし、何十万ものデータを目で検査することは不可能である。そこで原データよりデータを100個おきにサンプルし、原データの性質を良く反映していると考えられる新しいテストファイルを作成した。

このテストファイルを使用することにより、年度による微妙なデータ構造の変化、たとえば、74年度には、非悪性腫瘍の数の項目が存在しないことや、74年度より78年度にかけてメモの項目が存在することなどを、前もって把握することができた。また、テストファイルは、実際の大量データを処理する前にプログラムのデバッグ用の入力ファイルとして使用し、その処理時間より実際の処理時間を推定するのに用いた。

エラーデータの取扱い： 前述のテストファイルでも把握できなかったデータ構造、磁気媒体の変換に伴う異常文字の付加、データが整理中でデータ本体が存在し

ない、などの理由でデータが処理からはずされる事がある。データの量が大規模になると、エラー訂正の件数費がばかにならない。そこで、後の人間の作業労力を軽減する一助としてエラーメッセージを漢字で出力する方法を採用した。

* 割検年度がおかしい
72-----

漢字の採用によって、エラーの意味が見てすぐわかり、作業全体の効率を高めた。これと同時に、プログラムのデータ入力部分に、*印のある行は無視するような論理をいれたため、一部のエラーデータを修正した後、エラーファイルを再度おなじルーチンに投入し、エラーファイルの大きさを縮小することが可能となる。また、漢字でエラー内容が表示されるため、エラーファイルを調べることにより、書式変換ルーチンの論理ミスを検査することが簡単になる。

B 時間的遅延要因に対する対策

CPU 処理時間の高速化： 書式変換プログラムは DISK I/O をのぞき、大きく分けて各項目への分解、ID部分の生成、固定長部分の生成、可変長部分の生成の4つのモジュールより構成されている。これらのモジュールの CPU 処理時間の最適化をはかるため、各モジュールの処理時間を測定した。幸い VAX DSM には Elapsed CPU Time を測定できる \$ZHorolog関数があるので、これを利用した。その結果、各モジュールの処理時間の比率は、それら4つの処理時間の合計に対し

1. 各項目への分解	23.3%
2. ID部分の生成	14 %
3. 固定長部分の生成	28.1%
4. 可変長部分の生成	28.5%

なる割合を示すことがわかった。通常どこか一つのモジュールがボトルネックになっていれば、そのモジュールを改良するが、この程度の割合ではその必要がないと考えた。ちなみに、一般に言われているプログラム高速化の方法である、一行にできるだけ多くの命令を書く手法を、各項目への分解をするモジュールに程こしたが、0.2%程度しか時間の改善が見られなかった。これは VAX DSM が単なるインタプリタでなくプレコンパイルを行なっているためと考えた。

Disk I/Oの高速化： プログラム上での Disk I/O の高速化について述べる。主なプログラムの入出力ファイルは、原始データ、生成されたグローバル変数、逆ファイル生成用ファイル、の3つである。単純に一つのディスクの上で作業を行なうと使用したカートリッジディスクのアームが繁雑に動いて、処理速度が低下してしまうので、10MBのカートリッジディスク各々に前述のファイルの一つづつわりあて、ヘッドアームの移動回数を減じた。また生成されたグローバル変数、逆ファイル生成用ファイルに使用する出力用ディスクは使用直前に一度イニシャライズして、物理的にもファイルが連続して書かれるようにした。

つぎに、システムパラメータによる Disk I/O の高速化について述べる。VAX DSM

は他のスタンドアロンのシステムとことなり、VAX/VMS Record Management System を利用して Disk I/O を行なっているので、VAX/VMS のコマンドを使用して、システムパラメータで複数ファイルを利用するように指定した。

```
$ SET RMS/BUFFER=10/INDEXED  
$ DSM/NOOPT/NOSH/KEY= ---
```

つぎにVMS ユーティリーの利用による Disk I/O の高速化について述べる。実際の処理にあたり、原始データは年度、施設番号に関してソーティングされたもの入手できたので、症例グローバルの生成に関してディスクのアームの動きはなめらかであると言えよう。しかし、各原始データの中の4桁コードをもとに生成される逆ファイルに関してはそうではない。そのままデータを作成してはインデックスファイルの構造がきたなくなり、アクセス速度の低下にもつながるので何らかの対策が必要となった。そこで、すべて MUMPS 世界のみで仕事をするのではなく、VMS の SORT ユーティリーを利用する方法を採用した。つまり逆ファイルは一時的に VMS の シェンシャルファイルにおとし、書式変換処理終了の後、VMS の SORT ユーティリーを利用して4桁コードに関してソーティングし、その後これをもとに MUMPS により逆ファイルのグローバルを作成するようにした。こうすれば論理的にも物理的にも連続したグローバル変数入手できることとなる。

C 物理的遅延要因にたいする対策

マルチボリュームの採用：我々の使用しているカートリッジディスクは10M byte のものであるため、10万件もの症例グローバルと逆ファイルを作成しようとする、ファイルがディスクよりあふれてしまう。そこでディスク2台を論理的に結合し20M byteのディスクとする、マルチボリュームを採用した。この結果10万件近いデータを一度に扱えるようになった。

コンパクトなファイルの設計：マルチボリュームを使用しても、ややもするとファイルの大きが大きくなりすぎてしまい、あつかいが不便になる。そこで、よりコンパクトなファイルを作成することをこころみだ。VAX DSM のグローバル辺数はB ツリーでなく VAX/VMS Record Management System の上に構築されたインデックスファイルを使用している。Record Management System によるファイルは VMS の上の ANALYZE コマンドを利用して、ファイルのパラメータの解析ができる。ANAYZE コマンドの解析結果は、FDL ファイルとして出力され、その中にはパケットの大きさ、キーの大きさ、キーの部分圧縮の有無、データ部分圧縮の有無、その他、色々なパラメータが表示されている。この解析パラメータを自分の用途に応じて変更して、FDL ファイルよりファイルヘッダを生成すれば、ある意味での最適化が行なわれたファイルができることになる。そこで、テストファイルから生成された症例グローバルを ANALYZE し、キー部分とデータ部分の圧縮をするようにファイルパラメータの調整をしたのち、これをもとに新規に症例グローバルを作成した。実際に調整したパラメータは以下のとおりである。

```

FILE
  BEST TRY CONTIGUOUS  no -> yes
  CONTIGUOUS          no -> yes
KEY 0
  DATA KEY COMPRESSION  no -> yes
  DATA RECORD COMPRESSION no -> yes

```

4. 実際の処理時間

実際の処理は処理時間を少しでも短縮するため、他のユーザーのいない深夜に実行した。1979年より1981年にかけてのデータ108000件に関する処理時間と、ディスクの大きさは以下のとおりであった。

	時間	件数	Disk Block
剖検グローバルの生成	2:35	108000	12479
ソーティング	0:23	212000	
逆ファイル生成	2:50	212000	15702
キー長	32byte		
平均データ長	69byte		
ディスク効率			$(32+69)*108000/(512*12479) = 1.66$

5. むすび

個々に最善であっても全体としては最善であるかどうかには疑問が残るものの今回は、可能な最善と思われるほとんど全ての方法を併用して処理に当たった。これで過去10年間の剖検データが統一された形式で1つのグローバルとその補助逆グローバルにデータベースとして格納できた。その件数は、30万件に達し全国の年間全死亡者の半数に匹敵する数に達している。WHOを始めとする国際機関もこれに注目しはじめている。検索し易い形でデータベース化されているとは言え、国際的にも、このデータベースが活用されるには、いくつかの問題をかかえている。今後は検索機能に重点を置いて、システムの改善につとめるが、MUMPSはこれに充分答えてくれることであろう。

参考文献

- 1 : Kimura, S., Baba, K. et al : Frequency tabulation of pathological findings of more than 23,000 autopsy cases of annual collection through JAPAN
Proceeding of MUMPS USER'S GROUP MEETING, 9: 126-132 U.S.A
- 2 : 福永泰明、馬場謙介、田久浩志：剖検輯報のデータベースを例としてデータチェックにおける日本語処理の効用
第11回日本MUMPS ユーザーズグループ学術大会発表予定

表 1 剖検集計表の例

主 表：主病理診断の性別・年齢階級別集計表
1. 悪性腫瘍（原発部位別）
2. 悪性腫瘍（組織型別）
3. 良性腫瘍
4. 感染症
5. その他の疾患
副表 I：悪性腫瘍の組み合わせ集計表
a) 原発部位別・組織型別集計表
b) 原発部位別・臓器転移別集計表
c) 原発部位別・リンパ節転移別集計表
副表 II：多重がん組み合わせ集計表
a) 2重がん
b) 3重がん
c) 4重がん
副表 III：微小癌、早期癌
副表 IV：住所別集計表
a) 悪性腫瘍（原発部位別）
b) 良性腫瘍
c) 感染症
d) その他の疾患

表 2 病変グローバルのノード部分の構造

ノード	桁位置	内容
第一次ノード	1 - 2	剖検年次
第二次ノード	1 - 5	施設番号の枝 施設番号の枝 剖検番号の枝 剖検番号の枝 J O B
	6	
	7 - 11	
	12	
	13	

表 3 病変グローバルの固定長部分の構造

桁位置	内容	桁位置	内容
1 - 3	年齢	13	備考 住所 職業 職業 悪性 腫瘍 進展 機能 腫瘍 の数 の 数
4 - 5	月	14 - 15	
6 - 7	日	16 - 17	
8	不詳記載なし	18	
9	児区分	19	
10 - 11	胎令	20	
12	性別	21	

表4 病変グローバルの可変長部分の構造

	デリミッタ	内容	桁数
悪性腫瘍部	\$	悪性腫瘍の原発臓器コード	4桁
	/	同組織型コード	4桁
	:	同転臓器の数	2桁
	/	同転臓器コード	3桁
	:	同転リンパ節の数	2桁
	:	同転リンパ節コード	2桁
	@	上記のくりかえし	未定
非悪性腫瘍部	\$	悪性腫瘍以外の疾病コード	4桁
	/	死因	未定
	@	上記のくりかえし	
予備	&	メモ	

表5 逆ファイルの構造

ノード	ノードの内容	データの内容
第一次ノード	病変グローバル中にある4桁の	第一次キー下位ノード数合計
第二次ノード	ノード	第二次キー下位ノード数合計
第三次ノード	剖検年度 病変グローバル第二次キー	ヌル

^BOKEN(74.1010107400100) =	11	1	01	##0098/
^BOKEN(74.1010107400200) =	012	1	01	##2869/
^BOKEN(74.1010107400300) =	022	1	02	##2959/65320/
^BOKEN(74.1010107400400) =	047	1	10	\$16992039/://:/\$
^BOKEN(74.1010107400500) =	025	2	01	##4249/
^BOKEN(74.1010107400600) =	039	1	02	##5189/64859/
^BOKEN(74.1010107400700) =	042	1	10	\$16318583/://:/\$
^BOKEN(74.1010107400800) =	02	2	01	##5199/
^BOKEN(74.1010107400900) =	053	1	01	##0707/
^BOKEN(74.1010107401000) =	083	1	10	\$16198073/://:/\$
^BOKEN(74.1010107401100) =	08	2	01	##7472/
^BOKEN(74.1010107401200) =	027	1	10	\$19199393/://:/\$
^BOKEN(74.1010107401300) =	02	2	01	##7479/
^BOKEN(74.1010107401400) =	05 2381	2	02	##4859/67788/
^BOKEN(74.1010107401500) =	01	1	02	##7587/67507/

図1 病変グローバルの例

```

^INVERS(1049) = 9
^INVERS(1049,74) = 1
^INVERS(1049,74.6141100062700) =
^INVERS(1049,75) = 3
^INVERS(1049,75.1131400399100) =
^INVERS(1049,75.5070100065500) =
^INVERS(1049,75.5100100037300) =
^INVERS(1049,77) = 2
^INVERS(1049,77.1020100389300) =
^INVERS(1049,77.6370100015400) =
^INVERS(1049,78) = 3
^INVERS(1049,78.1040107818300) =
^INVERS(1049,78.1100100511300) =
^INVERS(1049,78.5090400102800) =
^INVERS(1129) = 103
^INVERS(1129,74) = 18
^INVERS(1129,74.1010107413300) =
^INVERS(1129,74.1040107409000) =

```

図2 逆ファイルの例

VAX-DSMへ漢字システムとグラフィックスシステムの統合化

○ 馬場 勝美^{*}, 水沢 博士, 宮野 雅司, 安松 範郎, 重松 仁, 野口 正雄

日本専売公社 中央研究所

横浜市緑区梅ヶ丘6-2 〒227

*メディカル・エンジニアリング(株)

東京都府中市宮西町5-8 〒183

「要約」

VAX-DSMは、パソコンでは常識となっている日本語処理機能とグラフィックス機能が装備されていない。そこで我々は、DSMを漢字VMSのユーティリティーを利用して日本語処理が可能になるように拡張した。さらに、数値統計処理、グラフィックス機能に優れたRS/1(BBN社)をDSMへ統合することで、両機能を利用したたばこ用香料開発支援システム「調香システム」の開発をした。

1. システム構成

CPUはVAX11-750(3MB)、コミュニケーションインターフェイスはRS-232Cである。漢字・グラフィックス端末として、第2水準漢字、外字登録ユニット及びVT80エミュレーターを付加した日本電算機(株)製KG1403を使った。プリンターはブラザー工業(株)製M5024CGで、第1、2水準漢字と外字登録ユニットを装備している。プロッターはヒュレットパッカー社製HP7220Tである。ソフトウェアとして、漢字VMS(日本DEC社version3.6)をOSとしたDSM(日本DEC社version2.0)とRS/1(BBN社version12.00)を利用した。

2. DSMでの日本語処理

DSMでの日本語処理は、VAX上のDSM以外のシステムとの互換性を保つため、

また日本語処理の装備を容易にするために日本語処理機能をOS内部に組み込んだDEC社の漢字VMSを利用することとした。

漢字VMSで採用されているDEC漢字コード体系は、7ビット2バイトのJIS区点コードに準拠し、これに1ビット加えた8ビットのバイト構成をとっている。このコード体系では、1バイトコードの半角文字と2バイトコードの全角文字とを区別するために、先行する1バイトの最高位ビットのオン、オフを用いている。DEC漢字コードでは、半角カナ文字を用いる時を除いて、半角文字、全角文字混じりの日本語データフィールドにデータ以外の漢字シフトキャラクターを必要としない。これは基本的に8ビット/バイトの内部処理が可能なDSMは、日本語処理で必要となるいくつかの機能を付加することで日本語化ができる。

日本語処理で必要となるいくつかの機能は、漢字VMSのランタイムライブラリーを、DSMのユーザ定義機能である\$ZCALLに組み込み利用した。

入出力は日本語処理で特に重要である。日本語入力については、`^KINPUT8`という漢字変換モード付きの汎用漢字入力ルーチンを作った。このルーチンは、全角と半角の文字識別ができ、カーソル移動によるスクリーンエディット機能を備える。半角文字はそのまま入力する。全角文字の入力はTABキー入力により、漢字単語変換機能を備えた「挿入」モードに入る。このモードは、漢字VMSの標準漢字エディタであるJEDIの「挿入」モードと同一のインターフェイスを持っている。

この`^KINPUT8`ルーチンの利用により、アプリケーション中で容易に漢字かな混じりのデータ入力ができる。このルーチンにより入力したデータは、半角文字と全角文字が任意に混在できる。このためDSMの組み込み関数である\$LENGTHの返す値は、文字数ではなくデータ長をバイト数で返す。同様に\$EXTRACTで引数に使う値は文字数がバイト数となる。このことは画面表示、文書印刷で全角文字が半角文字の2倍の領域を占めることを考えると、書式を整えるためには都合が良い。しかし、こうしたデータの部分文字列を扱う時、1文字の全角文字を構成する2バイトの中間でデータを切らないように注意を要する。これにも、漢字VMSのランタイムライブラリーを組み込んだ\$ZCALLを利用し解決した。

次に出力であるが、画面出力については問題は生じない。また全角文字、半角文字の混在した日本語文書の印刷は、まず印刷すべき文書を漢字VMSの順次編成ファイルとして出力し、この出力ファイルを漢字VMSのユーティリティーKPRINTにより印刷書式を指定したのち、プリンターに合ったコードに変換して出力する。これにはDSMから\$ZCALL関数の%SPAWNにより、漢字VMSのDCLコマンドインタープリターを実行した。

3. DSMからRS/1の数値統計処理、グラフィックス機能の利用

Bolt Beranek and Newman 社のソフトパッケージであるRS/1 (Research System 1) はDSMが持っていない数値統計処理、グラフ化処理機能に優れ、これらの処理を対話形式で実行できる。このRS/1の機能をDSMから利用する時に、まず問題となるのはデータの受け渡しである。RS/1ではデータを全てテーブルとして扱うが、VAX/VMSの順次編成ファイルとデータのやり取りを行う機能がある。そこでDSMとRS/1間のデータ交換には、中間ファイルとしてVAX/VMSの順次編成ファイルを使い、DSMのグローバルデータをRS/1で数値統計処理、グラフィックス処理を行う。

実際の処理の流れとしては、RS/1で処理すべきDSMのグローバルデータを、RS/1のレイアウトテーブルに沿った形でVAX/VMSの順次編成ファイルとして作成し、DSMの\$ZCALL(%SPAWN, "RS1")により、RS/1を実行する。RS/1では、DSMで作成された順次編成ファイルからRS/1のテーブルヘデータを読み込み、数値統計処理、グラフ化処理を行う。RS/1での処理を終了すると、処理はDSMへ戻る。このRS/1操作は全て対話形式で可能である。さらに処理が固定していればDSMと、RS/1のプログラム言語であるRPL (Research Programming Language) により、プログラミングできる。プログラムで処理を実行した時の制限として、RS/1はDSMのようなプロセスの共有ができないため、1つのRS/1処理が完了するまで次のRS/1利用はできない。

4. 調香システムの開発

これらの結果を踏まえて、たばこ用香料のデータベースを基本としたたばこ用香料の開発を支援する「調香システム」を開発した。

 * 調香システム *
 * 日本専売公社中央研究所 *
 * *****

- A. 調製標準作成システム
- B. 香素・香料検索システム
- C. 香素・香料入力システム
- D. 終了処理システム

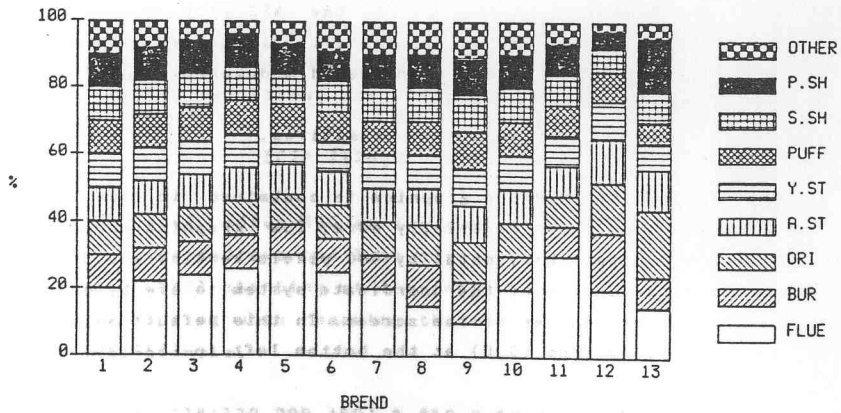
いずれか選択してください *

「調香」システムのメインメニュー画面

BA4

01-NOV-84 8:45 Page 1

DATA FROM BA3



DSMのグローバルデータをRS/1で利用したグラフ例

「システムデザイン2」 ④

Micro-MUMPS Enhancement Report

○ Donald A. Smith and Ichiro Wakai, M.D.

MUMPS System Laboratory, Nagoya

Enhancements to UCD micro-MUMPS include color graphics, screen image filing, ZCALL facility, mouse interface, 8087 mathematical co-processor support, and \$ZEDIT function for screen editing.

At the 1983 Japan MUG meeting in Tokyo, we reported on the beginnings of a graphics MUMPS implementation for selected micro-computers. In this paper we continue the discussion of last year, describing the completed first version of graphics MUMPS together with related functionality. We will also report on future plans in this area.

For sake of easy reference we outline progress in the following areas:

1. Graphics output primitives.
2. ZFG, ZDG screen image storage facility.
3. ZCALL facility.
4. MOUSE and light pen interfaces.
5. 8087 math co-processor.
6. \$ZEDIT function for screen editing.

The graphics described herein run on the NEC-9801 series micro-computer. Our next pre-compiled version will allow easy porting to any similar micro-computer, thanks to increased modularity of design.

1. Graphics output primitives.

The kernel of MUMPS's graphics is contained in the graphics primitives which we describe briefly here. The graphics screen of the NEC-9801 is 640 pixels horizontal by 400 pixels vertical. UCD graphics MUMPS defines the origin of the coordinate system to lie by default at the lower left hand corner of the screen. In this default setting, one can address pixels from (0,0) at the bottom left to (639,399) at the upper right corner.

ZLINE (0,0,320,200),

for example, draws a line from the bottom left corner to the center of the screen. If one wants to define the center or the coordinate system to be elsewhere on the screen, one can enter

```
S $ZX=320,$ZY=200 ZLINE(-320,-200,0,0)
```

for example. The line would then appear at the same position as in our first example, since the center of the coordinate system was moved appropriately. If the argument to ZLINE lies off the physical screen (relative, of course to \$ZX and \$ZY), then appropriate clipping is performed so that only that part of the line lying within the screen is displayed.

In addition to the above four argument form of ZLINE two other forms are available. The two argument form draws a line from the last drawn dot to the dot represented by its two arguments. For example,

```
ZLINE (0,0,100,100),(320,200)
```

would draw a line from the current center up and write 100 pixels, and then another line from that second point to the center of the screen assuming default window(\$ZX and \$ZY) settings. Finally, the zero argument form of ZLINE simply redraws the last line (for example "ZLINE<cr>"), but possibly with a new color or in a new window setting.

\$ZCOLOR is a special variable containing the current color for all graphics output. The NEC-9801 supports seven colors (mixtures of the three primaries, red, green and blue) plus the color black (meaning "erase"); each color is represented by an integer (0=white, 1=blue, 2=red, 3=purple, 4=green, 5=light blue, 6=yellow, 7=black). Thus,

```
FOR I=1:1:100 S $ZC=$R(7) ZLINE($R(640),$R(400))
```

draws 100 connected lines of various colors.

Similar to ZLINE, ZRECT draws a rectangle(unfilled) while ZBOX draws a filled (painted) box. Both depend similarly on \$ZX,\$ZY and \$ZCOLOR. ZCIR has two forms, for example:

```
ZCIR (320,200,100)
```

draws a circle of radius 100 at the center of the screen, while ZCIR with no arguments redraws the last circle. Finally, ZPAINT fills the screen with \$ZCOLOR starting at the point represented by its first two arguments, stopping when reaching the color represented by its third argument. For example,

```
S $ZC=4 ZCIR(320,200,100) S $ZC=2 ZPAINT(320,200,4)
```

draws a circle of color green(4) and fills its area with solid red. The ZCLEAR clears the screen of graphics data. ZCLEAR with no arguments clears all colors on the screen, while, for example

ZCLEAR 4

clears only the green pixels.

In addition to the above functions, the special variables \$ZWIDTH and \$ZHEIGHT are used to define the current window size for the x and y coordinates of the logical screen. For example,

```
S $ZX=0,$ZY=0,$ZW=640*2,$ZH=400*2 ZLINE (0,0,320,200)
```

draws a line from the bottom left corner to a point 1/4 of the screen up and to the right (since the screen is now logically twice as wide and high). Screen output can easily be inverted by defining the \$ZW and/or \$ZH to be a negative value. (We will see below how the windowing facility provided by \$ZX,\$ZY,\$ZW and \$ZH could be improved in future versions.) Finally, other special variables determine the density of graphics output (exclusive of ZPAINT); the details would be distracting here.

2. ZFG, ZDG screen image storage facility.

Any graphics image created by the above commands (or by directly writing to video RAM) can be stored on disk and subsequently reloaded by using these two commands (ZFG = "file graphics", ZDG="display graphics"). For example,

```
D ^MAKE1 ZCLEAR 4 ZFG "A:PLATE1"
```

stores all the image generated by routine ^MAKE1, except for its green VRAM page, in the file "PLATE1.GRA" on drive A:.

```
ZDG "A:PLATE1" H 1 ZDG "B:OVERLAY"
```

reloads from drive A: the first image and overlays onto the screen the second image stored in "OVERLAY.GRA" on drive B: . ZDG does not erase pre-existing screen data (it logically OR's the bit pattern). Thus, for example, variable data of one color can be readily compared to fixed reference values of another color.

Also, images are stored on disk in compressed form. Typical files occupy less than 30 kilobytes; uncompressed, they would all be 96 kilobytes, the size of graphics video RAM.

3. ZCALL facility.

Users wishing to write their own functions and commands for micro-MUMPS can do so by use of the ZCALL facility. At the operating system prompt "A>", instead of simply entering "MUMPS", one attaches the name of a ZCALL program file (whose assumed type is ".EXT"). For example,

"MUMPS MOUSE" initiates MUMPS and loads the external file MOUSE.EXT which is assumed to contain commands and their definition in a pre-determined format. For those who know 8086/8088 assembly language, or who have access to a compiler for PASCAL or FORTRAN which provides access to the machine level, it is very easy to create new commands. At the MUMPS System Laboratory we have ZCALL programs for PEEK, POKE, mouse interface and for writing character fonts to video RAM.

The syntax of the ZCALL command is exemplified by:

```
SET SEG=$ZHTD("A800"),OFFSET=0,BYTE=$CHAR(255)
ZCALL POKE(SEG,OFFSET,BYTE).
```

(\$ZHTD -- hex-to-decimal-- converts its string argument from hexadecimal to decimal.) The above command writes the byte 255 (OFFHex) at segment \$A800, offset 0. (BYTE can in fact be a string, in which case all its characters are POKEd.) Notice that arguments to ZCALL, if they exist, must be local variable. MUMPS passes the address of the local variables to the ZCALL program, which can then use the pre-existing value stored in the variable, or can replace the value with new data. Thus,

```
ZCALL PEEK(SEG,OFFSET,BYTE)
```

inputs the byte (or bytes, depending on the length of the variable "BYTE") at segment SEG, offset OFFSET. If the segment field is left out, MUMPS' DSEG is assumed.

In the next section we describe the MOUSE facility which uses the ZCALL command.

4. Mouse interface and light pen.

Using the ZCALL facility described in the previous section, a mouse interface has been designed for UCD micro-MUMPS. At the operating system's "A>" prompt, one enters "MUMPS MOUSE"; after so doing (and assuming the mouse hardware is properly installed) several forms of the MOUSE command are available.

```
S ON=1,ZCALL MOUSE(ON)
```

turns the mouse cursor on, positions it in the center of the screen, and activates the processing that moves the cursor in synchronization with the motion of the mouse on the desk. The cursor is prevented from moving off the screen by the interface software.

```
ZCALL MOUSE(X,Y,LEFT,RIGHT)
```

returns in X and Y the screen coordinates of the cursor on the screen;

further, LEFT (RIGHT) receives the value 1 if and only if the left (right) button of the mouse is being pressed.

```
S OFF=0 ZCALL MOUSE(OFF)
```

turns the cursor off. As an application of the mouse facility, a drawing program, which we describe next, has been developed. Along the left margin of the screen a table of colors is available for selection via the mouse (one positions the mouse cursor on the desired color and presses a button on the mouse). Thereafter, if for example one is currently drawing circles, the word CIRCLE on the menu line at bottom of the screen glows with the selected color. Similarly, one selects the current mode with the mouse: CLEAR, LINE, BOX, CIRCLE, PAINT, RECT or END.

An example of the MUMPS code to support this is in listing #1.

Finally, and in passing, let us mention the existence of a light pen facility (which, unlike the mouse, has resolution corresponding to the text screen rather than the graphics screen). One sets the special variable, \$ZPEN, to any non-null character, for example:

```
S $ZPEN="+" F I=0:0 I $ZPEN'="+" W !,$ZPEN QUIT.
```

When the light pen is depressed, \$ZPEN obtains the X and Y coordinates of the screen position in a format similar to \$HORLOG. For example, \$ZPEN contains "12,40" if the pen was depressed in the center of the screen.

5. 8087 Math co-processor.

The INTEL 8087 mathematical co-processor performs hardware arithmetic on various arithmetic types: 2, 4 and 8 byte integers, 10 byte packed decimal, and 4 8 and 10 byte reals. It is basically a hardware math accelerator for the 8088 and 8086 micro-computers, providing, for example, a single hardware instruction for the square root function. Depending on the amount of "number-crunching" being performed, significant speed improvements are possible. While MUMPS is not typically used for such number-crunching applications, the availability of the co-processor greatly simplifies and speeds the addition of special mathematical \$Z functions to micro-computer MUMPS. (To get an idea of the range and accuracy of the 8087, we mention that the the 10 byte real format can represent numbers up to approximately 10E4932 to 19 significant digits--iAPX 86,88 User's Manual).

UCD micro-MUMPS supports the following math functions: \$ZSIN, \$ZCOS, \$ZTAN, \$ZSQRT, \$ZNLOG (natural log), \$ZLOG10, and \$ZEXP (to the base e). As the 8087 is dependent only on the 8086/8086 hardware, these math functions are NOT limited to the NEC micro-computers.

6. \$ZEDIT function for screen editing.

The special function \$ZEDIT is actually a READ command in disguise. For example,

```
S X=$ZE("")
```

reads a string into the variable X. Yet the advantage of \$ZEDIT over READ is that \$ZEDIT allows "screen-editing" of the string being inputted. For example, when entering a long address or a long line of MUMPS code, one can correct an error at the start of the line simply by moving the cursor to the desired character and retyping. If one has forgotten the "P" in "MUMS User's GROUP", one can press the INSERT key on the keyboard and insert the character. Characters to the right shift to the right and to the lines below. Lines below scroll down if necessary. (Also, an "I" appears on the status line at the bottom of the screen when insert mode is on.)

Similarly, one can delete the unwanted "U" in "MUMPUS User's GROUP" merely by positioning the cursor via the arrow keys, and then pressing the delete key. Characters to the right scroll left, and lines scroll back up if necessary. Other cursor controls (TAB, return to first position, delete to end of line) are also supported. Control-c reverts to the original input string. Indeed, \$ZEDIT is more powerful than READ also in that one need not start with the blank string.

```
W "Enter <return> or edit please: " S ADDRESS=$ZE(ADDRESS)
```

allows the user to accept the old address (by hitting return) or to edit it on the screen. By using escape sequence cursor positioning, user friendly data entry screens can be built. Too, an optional second argument to \$ZEDIT can specify the maximum length to which the edited string can grow.

We at MUMPS System Laboratory have found the \$ZEDITOR to be particularly useful as the basis of a full screen editor. \$ZEDIT does the editing of individual lines, while a MUMPS routine stored in a global manages selection of lines and other commands necessary for an editor. (Normally, \$ZEDIT is exited by hitting the RETURN key. It can also be exited via the ESCAPE key, or the UPPPER and LOWER arrows; the latter two can be enabled to exit when the user tries to move off the current line. In either case, the variable \$ZB receives the position of the cursor in the line when the arrow was pressed.)

Mixed one byte (ASCII) and two-byte (KANJI) characters can be edited with \$ZEDIT. Kanji are treated as single characters despite their differing physical representation.

FUTURE PLANS

Until pre-compiled MUMPS is available and significant speed improvements in math appear, UCD MUMPS graphics will remain integer-based. By "integer-based" we mean that the graphics primitives such as ZLINE cannot reference screen dots between integer coordinates; thus, for example, one can draw a line "S \$ZW=640,\$ZH=400 ZLINE (0,0,100,100)" in physical screen coordinates or "S \$ZW=2000,\$ZH=1000 ZLINE (0,0,100,100)" in a window LARGER than the physical coordinates. In both cases no loss of accuracy appears. However, if one draws a line "S \$ZW=1.0,\$ZH=1.0 ZLINE(0,0,0.1,0.1)" in a window smaller than the physical screen coordinates, the arguments are truncated to integers--i.e., one cannot address points between integer values.

Because UCD graphics MUMPS does its graphics clipping and transformations in integer arithmetic (using hardware multiply, divide, add and subtract instructions) output is fast. The attached listing (#2) shows a MUMPS routine (WINDOW) which does window and viewport transformations (the way many Basics do) in two dimensions for the current UCD MUMPS graphics implementation. Based on algorithms in (Newmann and Sproull), the routine returns x-coordinate and y-coordinate factors necessary for windowing and viewport transformations such that only two multiplications are necessary (one for each dimension). With this routine one defines the left, right, bottom and top values of one's "window" into the world coordinate system; further, one similarly specifies in what rectangular portion of the physical screen (what "viewport") the window should appear. Once defined, all arguments to, for example, ZLINE need only multiplied by the factors outputted by the WINDOW routine. (For example, ZLINE (X*FACTORX,Y*FACTORY). \$ZX and \$ZY handle the offsets, while ZLINE handles clipping.

In our next graphics version we plan to support such windowing transformation as built in commands. Further, the local symbol table structure will then include an internal real format; the 8087 real format will be optional. (The current MUMPS stores all numbers in ASCII.)

Much discussion is needed, however, before the MDC can decide on a standard for MUMPS graphics. For example, DataTree's DT-MUMPS (Beaman and Lison) does all graphics via ZCALL's to a standardized graphics library. While "ZCALL LINE(...)" is not as direct and transparent a syntax as "ZLINE(...)", the trend in other computer languages seems to be in the direction of defining interfaces ("bindings") to external graphics systems like Graphical Kernel System (American National Standards Institute). As the power of graphics hardware increases, this trend will probably continue; but for simple and limited graphics functionality the more direct and transparent syntax is to be preferred.

BIBLIOGRAPHY

American National Standards Institute. dpANS Graphical Kernel System. June, 1983.

Beaman, Peter D. and Lison, Herbert G. "A MUMPS Color Graphics Workstation". MUG Quarterly, Vol. XIII, Number 3/4, pp. 54.

Intel Corporation. iAPX 86,88 User's Manual. August 1981.

Newman, William M., Sproull, Robert F. Principles of Interactive Computer Graphics. McGraw Hill, 1981.

Listing #1

```
MLINE      ;DAS,MOUSE,GRAPHICS-ZCALL;7/6/84; DRAWS LINES FOR MOUSE.MMP
L          ZCALL MOUSE(X,Y,LEFT,RIGHT) I LEFT+RIGHT=0 G L
           I LEFT Q
           I Y>40,X<35 D COLOR^MOUSE,H^MSETUP G L
           S XO=X,YO=Y F K=1:1:30
L2         ZCALL MOUSE(X,Y,LEFT,RIGHT) I LEFT+RIGHT=0 G L2
           F K=1:1:30
           I LEFT Q
C2         I Y>40,X<35 D COLOR^MOUSE,H^MSETUP G L2
           ZLINE (X0,Y0,X,Y) G L
           F K=1:1:30
```

Listing #2

```
WINDOW     ;DAS,,;11/6/84;Sets window, viewport factors FACTX & FACTY
           ;Set VL,VR,VT,VB and WL,WR,WT,WB and enter at label IN to
           ;bypass prompts
           S: '$D(WL) WL=A S: '$D(WR) WR=B S: '$D(WT) WT="" S: '$D(WB) WB=""
           S: '$D(VL) VL=0 S: '$D(VR) VR=$ZW S: '$D(VT) VT=400 S: '$D(VB) VB=0
WL         W !,"Enter world x left: " S WL=$ZE(WL) I WL' =+WL W *7 G WL
WR         W !,"Enter world x right: " S WR=$ZE(WR) I WR' =+WR W *7 G WR
WB         W !,"Enter world y bottom: " S WB=$ZE(WB) I WB' =+WB W *7 G WB
WT         W !,"Enter world y top: " S WT=$ZE(WT) I WT' =+WT W *7 G WT
           ;
VL         W !,"Enter view x left: " S VL=$ZE(VL) I VL' =+VL W *7 G VL
VR         W !,"Enter view x right: " S VR=$ZE(VR) I VR' =+VR W *7 G VR
VB         W !,"Enter view y bottom: " S VB=$ZE(VB) I VB' =+VB W *7 G VB
VT         W !,"Enter view y top: " S VT=$ZE(VT) I VT' =+VT W *7 G VT
IN         S FACTX=VR-VL/(WR-WL);NEED TO MULT X BY FACTX LATER
           S FACTY=VT-VB/(WT-WB)
           S $ZX=-WL*FACTX+VL
           S $ZY=-WB*FACTY+VB
```

Listing #3

```
GPLANET    ;DAS,,graphics;8/22/84;Sun and planet in orbit.
W # ZC S $ZX=320,$ZY=200,$ZC=6 F R=1:1:20 ZCIR (0,0,R)
F I=-3.2:.05 S $ZC=7 ZCIR S $ZC=6 ZCIR($ZCOS(I)*170,$ZSIN(I)*100,5)
```

Micro MUMPSの使用経験

○木村 一元 独協医科大学 総研 ME
馬場 謙介 産業医科大学 第二病理

要約

NECのPCシリーズでのMicro MUMPSの使用経験並びにそこで開発されたプログラムについて紹介し、各システムでの問題点を述べMicro MUMPSの動きにも言及した。それを通じて今後のMicro MUMPSのあり方について述べた。

はじめに

マイクロプロセッサ技術及び周辺機器の発展により増々マイクロコンピュータの利用が増加してきている。我々は、1979年よりカルフォルニア大学のDr.R.F.Waltersによって開発されたUCD Micro MUMPSをNECのPCシリーズのマイクロコンピュータで使用してきた。当初はPC-8001で1982年からはPC-8801で、昨年からはPC-9801で使用している。この間にUCD MUMPSもVersion upされV.4.04にまでなった。また、8bitの80系MPUだけでなく16bitの8086系MPUでも使用できる様になった。今回、マイクロコンピュータ各機種におけるMicro MUMPSの使用経験並びにそこで開発されたプログラムについて述べる。

システムの概要

これまで、使用したシステムの概要およびその機器構成は下記の如くである。

1. PC-8001 システム

CP/Mはマイクロソフトウェア・アソシエイツ(MSA)提供のVersion 2版(5 inch)で、その容量は1ドライブ当たり130KByteである。MUMPSの転送(8 inch片面 -> 5 inch片面)は、CromencoとPC-8001とをRS-232Cで接続し行なった。MUMPSのVersionはV 3.50であった。本体(PC-8001)、フロッピーディスク(PC-8031, 5 inch)、拡張ユニット(PC-8011)、CRT(PC-8041)、プリンタ(M80 11, 千代田情報機器)、CP/M(V 2.0)

2. PC-8801 システム

CP/MはMSA提供のVersion 2.2A版(5 inch)で、その容量は1ドライブ当たり300KByteと先の2.3倍となっている。MUMPSはPC-8001で使用していたものを転送(5 inch片面 -> 5 inch両面)した。MUMPSのVersion upが行なわれた時は8 inch片面 -> 5 inch片面 -> 5 inch両面の手順でプログラムを転送した。

本体(PC-8801)、フロッピーディスク(PC-8031-2W, 5 inch)、CRT(PC-8851)、プリンタ(PC-8822)、CP/M(V 2.2A)

3. PC-9801 システム I

PC-9801(128KB)にPLUS-80カード(クロック6MHz, カノーブス電子)を付けたシステムで

CP/M-80 のソフト (XXX.COM) が CP/M-86 (NEC提供) の基で稼働するように成っている。
MUMPS の Version は V 3.51 で、このシステムのフロッピーディスクは 8 inch 両面 (980KB)
である。

本体 (PC-9801) , Z80 カード (カノーブス電子, PLUS-80) , フロッピーディスク
(PC-8881, 8 inch) , CRT (PC-8851) , プリンタ (PC-8822) , CP/M-86 (V 1.1)

4. PC-9801 システム II

標準構成の PC-9801 システムで、MUMPS は V 3.01 から V 3.60 まで使われた。

本体 (PC-9801) , フロッピーディスク (PC-9881, 8 inch) , CRT (PC-8853n) ,
プリンタ (PC-8822) , CP/M-86 (V 1.1)

5. PC-9801 システム III

10 MByte のハードディスクと 8 inch 両面のフロッピーディスクとを備えた PC-9801 システムで、
CP/M は日本語 CP/M-86 (V 1.1) である。MUMPS は V 3.60 である。

本体 (PC-9801) , フロッピーディスク (PC-9881, 8 inch) , ハードディスク (PC-98H33) ,
CRT (PC-8853n) , プリンタ (PC-8822) , 日本語 CP/M-86 (V 1.1)

Micro MUMPS の実装

Micro MUMPS の実装と使用法については、日本での Micro MUMPS の領布の委託を受けている
マンブスシステム研究所で書かれた "UCD micro MUMPS の User's manual (日本語)" と "CP/M
V 1.4 または V 2.2 で走る 8080 および Z80 用マイクロ MUMPS について 若井 一朗" を基に
行なった。実装において注意する点はグローバルファイルをどちらのドライブに置くかであるが
これもマニュアルの SETMUMPS, SETGLOB の説明に述べてあるのでこれに従えばよい。

これまで Version up に伴い問題となった点は、CP/M ファイルと互換性を取るため MUMPS ルーチン
ファイルの格納形式が変わった事と、グローバルファイル構造の改良がなされグローバルデータの
転送が必要であった事だけである。

Micro MUMPS でのアプリケーション

UCD MUMPS の中には数多くのユーティリティープログラム (18種) 及びアプリケーション
プログラム (5種) がこれらのシステム上で作成されたプログラムは多種に渡るが、ここでは我々が
開発した主なユーティリティー及びアプリケーションプログラムについて述べる。

1. ルーチンエディター (X ^%XE)

UCD MUMPS には J.J. Althouse が作成したエディター (X ^%) と J. Lewkowicz とが作成した
エディター (D ^%MRE) と二つあるが、これはそれらとは異なり行の指示が UP, DOWN Key
あるいは +, - Key でまた文字の位置の指示が TAB Key あるいは UP, DOWN, RIGHT, LEFT Key
で出来るようにした 1 ラインでのスクリーンエディターである。

2. グローバル・リスター

グローバルの内容 (添字およびデータ) を出力するルーチンでグローバル名あるいはグローバ
ル名と添字を指示する事ができる。添字レベルの制限はない。

3. 個人用文献検索システム

個人用の文献整理プログラムで、論文名、著者、雑誌名、キーワード、コメントのいずれもが検索の対象となるもので所謂キーワードインデックスシステムである。

4. 病理診断データ登録検索システム

病理診断データを整理する目的で作られたシステムで、患者番号、病理番号、提出年月日、診療科、医師、組織コード、組織診断、診断材料の別、数、診断者などが登録されている。

5. 細胞診データ登録検索システム

細胞診データを整理する目的で作られたシステムで、以前に BASICで組まれていたプログラムを MUMPS用に直したものである。

6. ノンパラメトリック統計サブルーチン

第七回日本 MUG 学術大会で発表した統計サブルーチンを Micro MUMPSに乗ったもので、カイ 2乗検定、スピアマンの順位相関、ケンドールの順位相関、マンホイットニーの U 検定、コクランの Q 検定、ケンドールの W 検定、フリードマンの二元配置分散分析、1次～4次の積率、ピアソンの相関とからなる。

7. DNA/RNAシーケンス解析援助プログラム

DNAあるいはRNAのシーケンス解析を支援するプログラムで、シーケンスを種々のフォーマットで出力する、相補鎖を作る、フラグメントを結合すると云った基本的な仕事を行なう。

8. 入学試験処理プログラム

入学試験成績処理プログラムで、当室で行なっているミニコンでの入試処理のダウン時のバックアップ・プログラムとして組まれたものである。

9. 命令、関数、演算子のチェック・プログラム

Micro MUMPS の動作を確認するために作られたチェック・プログラムで、52のルーチンから成り400以上の複合した項目(命令、関数、演算子)のチェックが行なわれる。

Micro MUMPS の使用上での問題点

ここで、これまでに経験した Micro MUMPSシステムの問題点及びそのシステムで気づいた点について述べる。

1. PC-8001 システム

5 inch片面のフロッピーディスクである為、マンブス (MUMPS.COM)のロードに時間がかかり過ぎた。また、ディスクの容量も130Kバイトと少なくルーチンとグローバルを1枚のディスクセットに混在させるには難点があった。MUMPS システムとしては、カーサーが現れない点、ルーチンのネスティングが深くすると CP/M に戻ってしまう点に使いにくさがあった。

2. PC-8801 システム

5 inchのシステムであるが、PC-8001 とは異なりCP/Mも安定しておりディスクの容量も300Kバイトと多く、個人用のシステムとしては充分なものである。ディスクアクセスの遅さが少し気にかかるが、これはシステムの価格面から相殺されると考えられる。また、MUMPS システムにも工夫が成されており極力ディスクアクセスの回数が抑えられている。このシステムでは、

多くのプログラムが開発されたがこの時に問題となったのはエディターであった。

3. PC-9801 システム I

Z80カードを用いたこのシステムは安心して使えるものであった。Z80のクロックも6MHzと早く、ディスクも8 inch 両面を装備しており PC-8801システムと比べものにならないシステムであるが BASICなどのスクリーンエディターに慣れると MUMPSのエディターには使いにくさが残る。現在、ワードスターのもとで MUMPSを走らせており基本的なプログラムの作成はワードスターで行なっている。

4. PC-9801 システム II

このシステムでは、8086/88用の MUMPSが V 3.01 から V 3.60 まで用いられた。Versionの低い MUMPS(SIBER, INC. 提供)ではその安定が悪くプログラムの作成あるいは実行時に頻繁にダウンしCP/Mに戻った。そこでその対策としてCP/Mから MUMPSに強制的に戻すルーチンを作製した。また、この Versionではある文字数の所で\$Extract,\$Lengthと云った関数が正常に働かない事があった。V 3.60では添字が0(ゼロ)のグローバルを作ると\$Next/\$Orderで添字が-:に化けた。

5. PC-9801 システム III

10MBのハードディスクと8 inch 両面(980KB)のフロッピーディスクとを備えたこのシステムではグローバル、ルーチンのディスクへのアクセスは全く気にならなかった。このシステムで気がついた事は、75.65E-8の値が.00000075654と僅かに異なる点とルーチンデレクトリーをView 3でフロッピーディスクにシーケンシャル・ファイルとして書きだした場合にデレクトリーが正常に書かれない点であった。このシステムでも基本的なプログラムの作成はワードスターを用いて行なっている。なお、ハードディスクの中には CP/M (5MB), BASIC (3MB), MS-DOS (2MB)の三つのOSが混在している。

む す び

Micro MUMPS の使用経験とそこで開発されたプログラムについて述べた。マイクロコンピュータでの MUMPSの使用は Dr. R.F.Waltersらの絶え間ない努力によって現実のものとなった。今後、増々 Micro MUMPSの利用者はその言語の学習のしやすさ、データファイルの作りやすさで増えると考えられる。しかし、今のままの MUMPSでは、彼らがまず直面するのはその特徴でなく先の問題点で述べた如く操作性であろう。特にこれまで BASICを使用していた人々はプログラムの作成時には当惑するであろう。今後のマイコン上での MUMPSの発展の為にはこの様な点まで考慮されなければならないであろう。

1. 地籍調査の目的、意義、趣旨、概要、キーポイント、

2. 地籍調査の歴史、現況、今後の展望、

3. 地籍調査の法的根拠、

4. 地籍調査の組織、

5. 地籍調査の業務、

6. 地籍調査の成果、

7. 地籍調査の活用、

8. 地籍調査の課題、

9. 地籍調査の展望、

10. 地籍調査のまとめ、

11. 地籍調査の参考文献、

12. 地籍調査の用語解説、

13. 地籍調査の図表、

14. 地籍調査の資料、

15. 地籍調査のその他、

16. 地籍調査の備考、

17. 地籍調査の索引、

18. 地籍調査の別表、

19. 地籍調査の添付資料、

20. 地籍調査のその他、

21. 地籍調査の備考、

22. 地籍調査の索引、

23. 地籍調査の別表、

24. 地籍調査の添付資料、

25. 地籍調査のその他、

26. 地籍調査の備考、

27. 地籍調査の索引、

28. 地籍調査の別表、

29. 地籍調査の添付資料、

30. 地籍調査のその他、

31. 地籍調査の備考、

32. 地籍調査の索引、

33. 地籍調査の別表、

34. 地籍調査の添付資料、

35. 地籍調査のその他、

36. 地籍調査の備考、

37. 地籍調査の索引、

38. 地籍調査の別表、

39. 地籍調査の添付資料、

40. 地籍調査のその他、

41. 地籍調査の備考、

42. 地籍調査の索引、

43. 地籍調査の別表、

44. 地籍調査の添付資料、

45. 地籍調査のその他、

46. 地籍調査の備考、

47. 地籍調査の索引、

48. 地籍調査の別表、

49. 地籍調査の添付資料、

50. 地籍調査のその他、

51. 地籍調査の備考、

52. 地籍調査の索引、

53. 地籍調査の別表、

54. 地籍調査の添付資料、

55. 地籍調査のその他、

56. 地籍調査の備考、

57. 地籍調査の索引、

58. 地籍調査の別表、

59. 地籍調査の添付資料、

60. 地籍調査のその他、

「ネットワークキング」

座長 里村洋一（千葉大医）

1984年度 第11回 日本マンプスユーザーズグループ学術大会
「ネットワーキング」 ②

藤田学園保健衛生大学病院におけるMUMPS使用経験

○山内 實 茂木 健一 大谷 元彦
水野 康 岩田 重信 藤田 啓介
藤田学園保健衛生大学病院
藤田学園コンピュータ管理センター

本大学病院は、1977年より、学内各部署からの出向者からなるチームでMUMPSによる総合医療情報システムの開発を独自にすすめてきた。

現在、DEC社9台、TANDEM社6台のミニコンと端末装置175台で医師自らが全オーダを入力する、いわゆる情報の発生源入力が実現している。

1. システム開発の経緯

大学病院開設（1973年5月）の半年後、記憶容量16KBのコンピュータ2台、オンライン端末2台、オフライン端末3台で、窓口会計、レセプト業務を稼働させた。1977年からMUMPSを採用して、外来医事務、検査システム、入院医事務そして病棟業務が次々と開始された。

1980年、外来診察室で医師がオーダを入力するシステムを開始、検査部では、オーダの受付、検体ラベル・ワークシートの作成、自動分析装置の検査結果のオンライン収集、検査結果報告書の作成、そして診察室での検査結果の検索が完成した。

1982年、病棟・ナースステーションで医師が処方オーダ、検査オーダ、X線オーダ注射オーダを入力するシステムが始まり、院内各所に配置した175台のディスプレイ装置でオンライン情報処理、部門間連絡システム、レセプト作成などを含むバッチ処理も完成した。

1984年現在、外来、入院とも24時間オンライン運転である。

表1に導入経過を示す。

表1 医療情報システム導入経過

昭和48年5月 開院

- 49年9月 NEC 2台 端末装置 5台 外来医事システム
 52年3月 コンピュータ管理センター発足 DEC社にてマンブス研修
 6月 教育・開発用のPDP 11/40 米国マンブスユーザー視察
 10月 看護部入力練習(200名) 外来用 PDP 11/70
 12月 入院用 PDP 11/34 検査用 PDP/34
 53年4月 外来医事システム稼働 [検査]~[外来] CPU-CPU通信
 7月 入院医事システム稼働 検査システム完成 欧米ユーザー視察
 54年5月 外来医師入力用 PDP 11/70 2台 DSM-11
 7月 第一回医師入力研修 10月 第二回医師入力研修
 55年1月 外来医師入力開始(内科) 5月 整形外科 精神神経科
 11/70のメモリ512KBを768KBに増設
 7月 小児科 外科 泌尿器科 放射線科 歯科 耳鼻咽喉科
 8月 眼科 皮膚科 産婦人科
 12月 NIHメディカル・センター研修 (看護長 5名)
 56年6月 TANDEM社 NonStop System I 6台
 11月 入院医事システムをDSM-11 (standard MUMPS) に書換え
 57年5月 TANDEMで外来システム稼働
 12月 病棟医師入力開始(処方オーダー)
 58年1月 " 検査オーダー X線オーダー 3月 注射オーダー
 12月 NECと契約 ACOS 3台 (コボル・日本語処理システム)
 59年5月 インテリジェント端末装置 4台
 8月 " 7台 ACOS 550
 (12月 ACOS 550 ACOS 410)
 (60年4月 新システム稼働)

2. 機器構成

DEC社のPDP 11/70 3台、11/34 5台、11/40 1台の計9台と
 TANDEM社のCPU 6台がある。

病棟はPDP 11/70 3台、外来はTANDEMが受け持つ。検査には11/34
 2台があり、11/70およびTANDEMと情報交換を行っている。

11/34 2台は、第三教育病院で1984年から日本語MUMPSで稼働しており
 残り2台の11/34は、CAIと人事管理に使用している。

図1に機器構成を、表2に端末装置の配置場所と台数を示す。

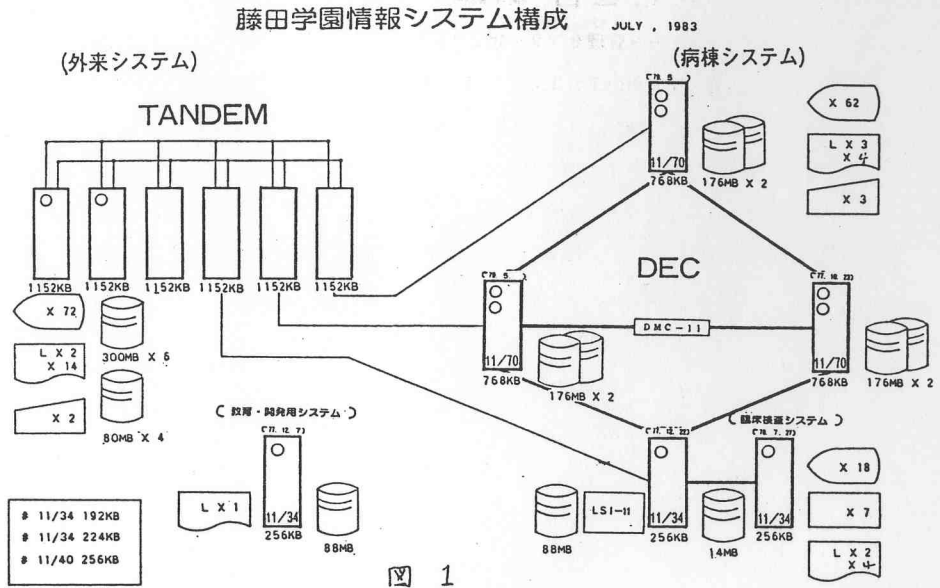


表2 藤田学園保健衛生大学病院 端末機の配置図 (昭和59年8月)

1号棟		2号棟		3号棟	
全科病棟(50) ③	精神神経科(35) ②	孔拝堂礼楽の間			15
循環器・神経内科(126) ③	消化器外科(73) ③	呼吸器内科(39) ②			14
こども病棟(113) ③	産婦人科(48) ③	膠原病内科(39) ②			13
整形外科(114) ④	耳鼻咽喉科(72) ③	消化器・腎内科(39) ②			12
脳神経外科(129) ③	眼科(66) ③	消化器・腎内科(39) ②			11
教授・助教・講師室	中央材料室	機 械 室			10
コンピュータ管理センター □ ⑦	腎センター 手術室	皮膚科・泌尿器科(39) ②			9
検査室 ⑥ □ □ □ □ □	薬剤部 ② 外来会計 ④	内分泌外科(39) ②			8
コンピュータ室 ⑤ □ □ □ □ □	内科・初診③ 小児④ 産科① 耳鼻科④ 内科・再⑥ 脳外② 特診室① 精神科③ 皮膚科④ 眼科② 歯科②	産婦人科(31) ②			7
食養部 ②	放射線科① R I ① リハビリ① 東洋医学診療センター①	血液内科(31) ②			6
		胸部外科(32) ②			5
		HCU(40) ②			4
		ICU, CCU(22) ③			3
					2
					1
					B,1
					B,2
					B,3

3. 対象業務と処理形態

本システムでは、情報の発生源である医師が自ら診察室で入力する。その情報は即刻必要な部署に転送される。

“検査”はCPU-CPU通信で検査室のコンピュータに取込まれる。“X線”は放射線部のプリンターに依頼票として、“投薬”は薬局のプリンターに処方箋として出力され“会計”は医師入力のデータをそのまま利用する。

また、病棟の予定オーダは、実施前日にバッチ処理で各部にプリントアウトする。

投薬、注射の緊急オーダは、入力後ただちに薬局のプリンターで緊急処方箋、緊急注射箋を出力する。

全診療行為について、過去のオーダ内容のオンライン参照が可能で、参照の結果、それを新規のオーダとすることも、あるいは、一部の変更、追加もできる。

表3に対象業務と処理概要を、表4に医師入力の初期画面、Doctor's Ordering Guideを示す。

“検査”、“薬剤”、“看護”システムは別に発表するので、ここでは医師入力業務と食養部業務について説明する。

1) 医師入力業務

医師は、自身のIDカード、つづいて患者IDカードをリーダーから読みます。すると前述の Doctor's Ordering Guide が表示される。そして該当する番号を選んで入力をすめることになる。

一般的には、この画面から開始するが、科によっては、いきなりLab. Test (検体検査) 入力の画面へ、あるいは前回処方画面へとぶものもある。また、いくつかの科で外来の場合は、よくオーダする検査、注射、処置などの項目が混在してひとつの画面に表示して、そのなかから必要なものをオーダする。

入力方法は、セットを多数つくり、画面の構成、表示方法、項目名検索法などは操作が容易でさらにミスを少なくするよう種々の工夫がなされている。

入力したデータの追加、変更、取り消し、確認の機能は、共通して全プログラムが持っている。

2) 食養部業務

病棟でナースが入力する給食から、一日3回、食数表を出力する。
 一年間の献立のメニューが登録されていて、それらを適宜変更して実施献立とする。
 食数と献立を把握しているため、それらから材料の発注伝票を作る。
 また、栄養計算、各種統計数の作成を行っている。

表3 適用業務と処理概要 (昭和59年8月)

業務名	処理概要	業務名	処理概要
患者登録	個有情報と保険情報の入力 受診登録カード発行 照会、変更、再診患者登録	放射線部	フィルムの種類、枚数 薬剤の入力、日報、月報
オーダー業務	検査、X線、リハビリ 処方、注射、処置、保険 初・再診を医師が入力 給食、転科、転室、外泊、 退院は看護婦が入力 入院の処置・手術は医事課	食養部	食数表、献立表 発注表の作成 カロリー計算、諸統計
		診療費請求	外来窓口会計、退院患者の 料金計算、入院患者の定期 請求、未収金管理
臨床検査	検体ラベル、採取表、ワー ク・シート、報告書の作成 検査結果入力	レセプト	外来 19033枚 入院 1877枚
薬剤部	納入伝票の入力、在庫管理	統計	行為別診療点数、病床利用 状況、診療圏

表4 Doctor's Ordering Guide

<p>*** Doctor's Ordering Guide ***</p> <ol style="list-style-type: none"> 1. Lab. Test (検体検査) 2. Physiology (生理検査) 3. X-Ray (X線検査) 4. Medicine (投薬) 5. Injection (注射) 6. Treatment (処置・手術) 7. Reservation (予約) 8. Health consultation (健診) 9. Blood transfusion (輸血) (9. CT) 10. ショサイシン (初・再診修正) 11. ホケンキリカエ (保険切替) 12. STAT lab. order (緊急検査指示) 13. STAT lab. review (緊急検査参照) 14. Lab. result review (検査結果参照) <p style="text-align: center;">Select [1-14 or E]</p>
--

病院情報システムによるLANの利用例

○田川順通，竹谷利夫

三井造船株式会社 システム事業本部

システム・エンジニアリング事業部 医療事業室

1. はじめに

近年、OA、FA等の分野においてLAN (Local Area Network) の導入が急速に高まりつつある。医療情報においても同様であり、特に病院情報システムでは、これまでも各種のネットワークを利用している。従つて、LANを利用することにより、さらに効率的なシステム構築が考えられる。そのため、当社においても医療情報分野におけるLANの応用について研究を進めてきた。今回は、病院情報システムによるLANの利用例として、大阪府立成人病センター敷に導入し、本年10月より本稼動しているシステムについて報告する。

2. システム概要

2.1 ローカル・エリア・ネットワークについて

本システムに使用したLANは、イーサネット仕様の同軸ケーブルを使用し、ベースバンド方式のNET/ONE (米国アンガーマン・バス社製)である。

NET/ONEは、同軸ケーブル、タップ・トランシーバ、NIU (Network Interface Unit) およびNSM (Network Storage Module) から成る。NET/ONEの主機器であるNIUは、ソフトウェア・コントロールによる記憶装置付のマイクロ・プロセッサーを内蔵している。ソフトウェア・ツールは、代表的なパーソナル・サーキットがある。これは、コンピュータと各種端末機間で、ポイント・ツウ・ポイントの通信ができるような機能である。従つて、これはLANの一般的な目的を満足させている。

同軸ケーブルは、データ伝送媒体であり、ネットワークはバス・トポロジーで作られている。1本のケーブルは500mであり、最高100個までのトランシーバが接続可能である。また、ローカル・リピータで1.5km、リモート・リピータで2.5kmまで拡張可能である。但し、1ネットワーク当りのノード数は、1024個である。

1ネットワーク当りの転送速度は10Mbpsであり、ライン・アクセスはCSMA/CD方式である。

本システムでは、NIUと端末機間ではカレント・ループにより接続であるが、これは、当社製のRS-232C/カレント・ループ仕様のI/Oインターフェースを組み込むことにより可能としている。

2.2 コンピュータ・システムとネットワーク

今回導入したシステムは、図1に示すようなイーサネット・ケーブル上に、25台のNIUを接続し、約250ポートのホスト・ライン及び端末機を接続可能にしている。

コンピュータ・システムは、PDP-11/44を5台で、OSはDSM-11 (DEC Standard MUMPS)の漢字版である。

端末機は、漢字CRT、漢字シリアル・プリンター、ラインプリンター、分析装置用インタ

一フェース、パソコン、およびその他特殊機器等の約110台がLANに接続されている。以上のコンピュータ・システムにより、DSM-11のDDPによるコンピュータ・ネットワークとLANによる端末ネット・ワークを構築している。(1)

3. MUMPSによるLANコマンドの利用

NET/ONEのバーチャル・サーキットには、端末機の切替え用としてコマンドが用意されている。このコマンドをMUMPSのプログラムから発生させることにより、ホストと端末機の切替えをプログラムにより制御することになる。

この機能を利用すると、LANに接続されている全端末装置が、ホスト・システムの仮想端末として取扱いが可能となる。また、利用者からみれば5台のうちいずれのコンピュータを使っているかを意識することなく使える。従つて、①システムを管理する側でコンピュータ間の負荷分散をはかりながら運用できること、②限られた端末装置の有効利用が実現される。③一台のコンピュータの障害時には、端末群、業務システムを他のコンピュータに切り替ることにより全体のシステムを停止させることなく業務が継続できる(1)、以上がLAN利用のメリットとに考えられる。

4. MUMPSによる端末装置の切替えシステム

前述したように、MUMPSのプログラムにより障害時の端末群の切替え可能である。

図2に示す内容について、一つの例として以下にその方法を説明する。

①外来会計システムに接続されている端末装置から、切替えプログラムを稼動する。

②切替え選択画面で、“臨床検査システム 総合管理システム”の項目を選択する。

以上で、プログラムが実行され、切替えが完了する。但し、切替えが不成立の場合は画面上に、エラーメッセージが出される。

切替えプログラムの実行は、次の方法で実施される。

①臨床検査システムのホストと端末装置群を切り離す

②総合管理システムのホストと端末装置群を切り離す

③総合管理システムのホストと臨床検査システムの端末装置群を接続する。

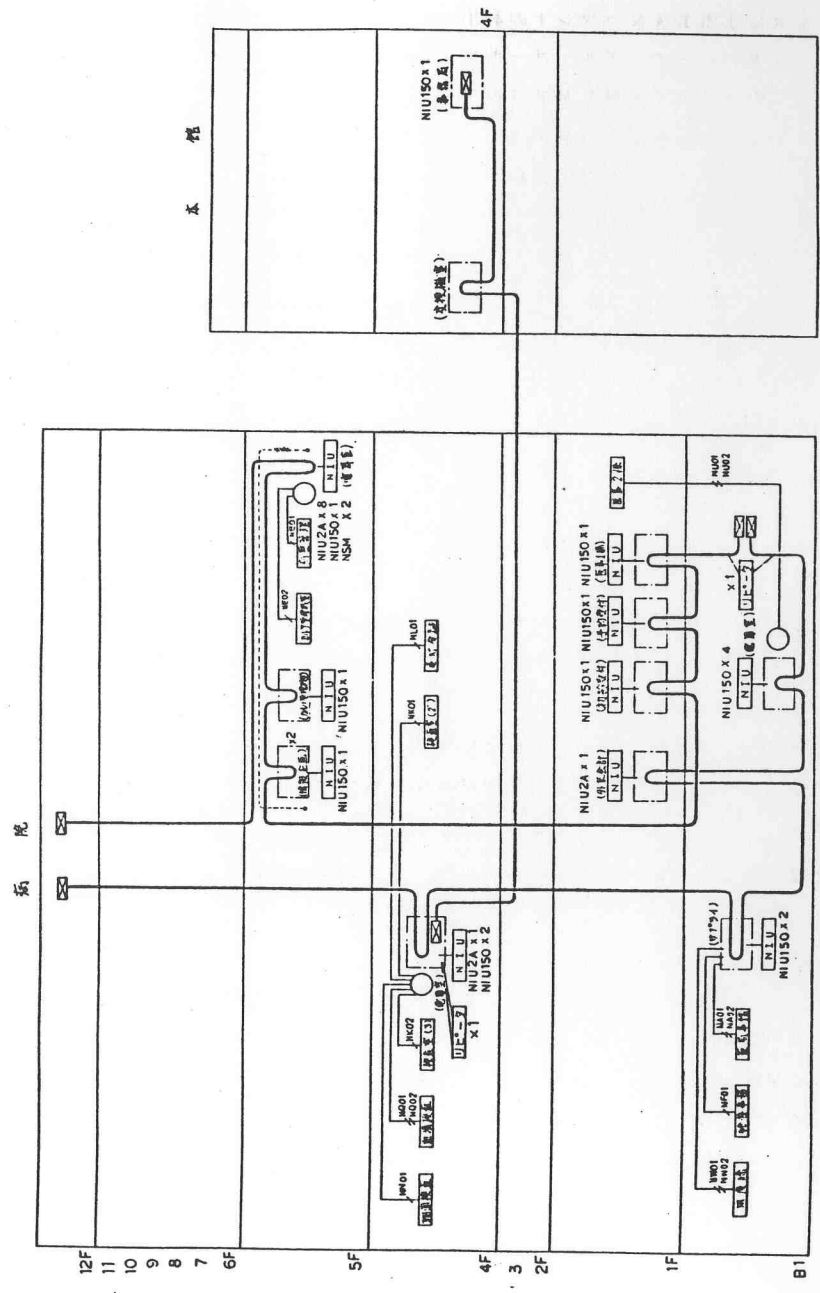
5. おわりに

以上のように、LANの利用の一例として述べて来たが、医療分野においては各種の活用方法が考えられ、またデータの種類により光ファイバーの検討も必要であろう。

今後、さらに発展するLANの利用について当社の導入例が1つの参考になれば幸いである。尚、今回の導入に当り、大阪府立成人病センター 情報企画室 野村室長、河村主査、他皆様に多大なご協力、ご援助を頂いたことに感謝する次第である。

参考文献

- (1) 河村徹郎：病院内LANと診療支援システム，映像情報 MEDICAL，
Vol. 16，No. 17，P879，P880，(261)1984



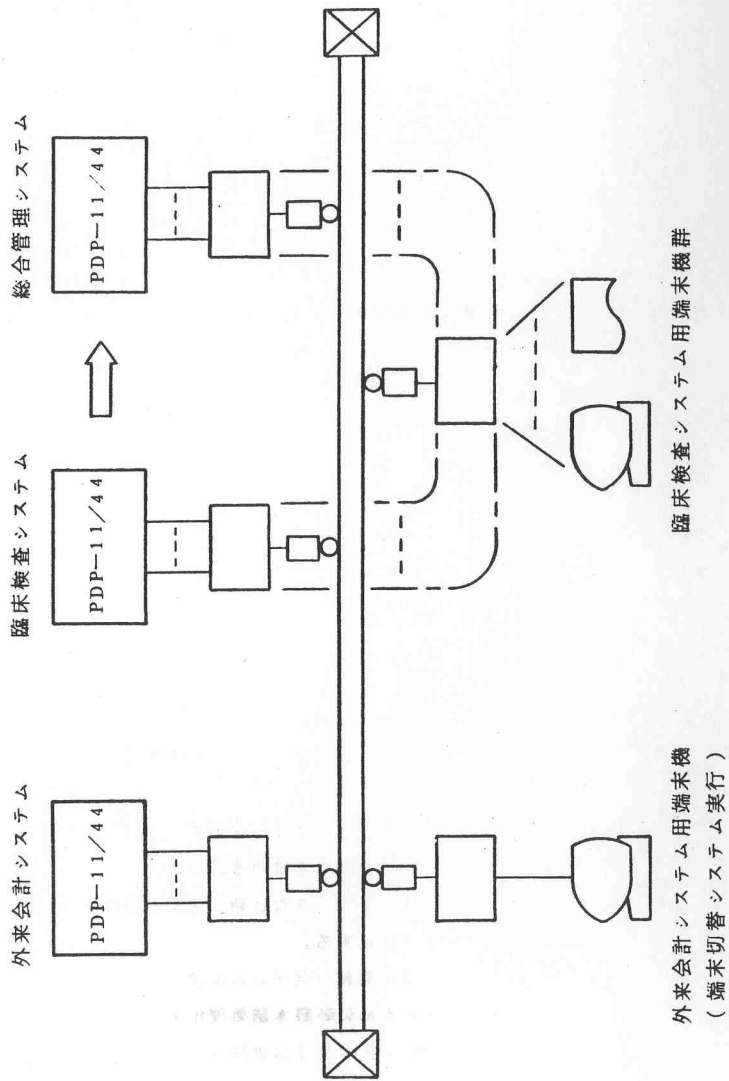


図 2 端末機切り替え方法の例

LANを用いた
病院情報システム

河村 徹郎、北川 謙、三嶋 博昭、藤江 昭、永井 利広、
上田 清治、服部 敏夫、青木浩二、野村 裕

大阪府立成人病センター 病院 情報企画室
大阪市東成区中道1丁目3-3

1. はじめに

大阪府立成人病センターでは、1977年より病院情報システムを導入利用してきた(1)。7年を経て新しいシステムに入れ替えるべく、第二期システムを企画し準備をすすめてきた。10月末現在、コンピュータを中心とするハードウェアを新機種に更新している。ここではこのような第二期病院情報システムの概要を紹介し、その特徴、システム構築にあたっての考察等について述べる。

2. システムの概要

2-1. 基本構想

第二期病院情報システムの設計は、旧システムに対する評価の結果(2)を基に次のような基本構想で設計した。

(1) 旧第一期システムのソフトウェア資源やデータ資源、要員の経験とノウハウを最大限に生かすため、第一期システムと同様に、ミニコンピュータと汎用中型コンピュータからなる分散処理とデータの集中管理構成を踏襲する。また、ミニコンピュータによる即時処理システムでは、プログラム言語としてMUMPSを継承する。

(2) CPU間を高速データ通信回線で接続し、コンピュータ・ネットワークを構築する。これによりCPU間でデータ転送を行ない、データの相互利用をはかる。

(3) 分散システム構成、コンピュータ・ネットワークなど新システム技術を活用し、システムの拡張性、柔軟性をもたせる。かつ、障害対策を強化する。

(4) コンピュータ・ネットワーク上で、適用業務システムおよびデータベースの再編成を行なう。

(5) 利用者の使い易さ、読み誤りの防止のために、日本語処理化をはかる。

2-2. システム構成

1) コンピュータ システム

新システムの構成を、図1、表1に示す。コンピュータ構成は従来の7台のPDP 11を、5台のPDP 11/44に置き換え、さらに次のような新技術を取り入れ、ネットワークを形成している。

- ① PDP 11/44 システム間は通信制御装置DMR-11を使い、CPU間の高速データ通信をはかる。
- ② PDP 11/44 システムとIBM 4361システム間はBSCによるデータ通信をはかる。
- ③ 5台のPDP 11/44と端末装置間の低速データ通信回線としてLOCAL AREA NETWORK (LAN)を採用する。
- ④ ソフトウェアシステムとしてDSM-11 V3.0 JBを採用する。

2) ローカル エリア ネットワーク (LAN)

LANの利用範囲として、CPU間のファイル転送も考えられるが、今回は次の理由で端末装置との間の低速通信回線に限定して利用することにした。

LANとして、表2に示すNET/ONEを採用したが、ベースバンド方式、ラインアクセスメソッドはCSMA/CD方式である。このLANの伝送速度としては10Mbps程度と公称されているが、いわば瞬間搬送速度であり、端末装置との間の瞬発的な送受信には9600bps程度は可能であるが、メッセージブロックが連続するファイル転送の場合には、さらに転送速度が低下し、実用的ではないと考えられる。また、PDP 11システムでは、制御装置DMR-11とDSM-11のDDP (Distributed Data Processing) 機能を使ったCPU間のデータ通信の実績と経験もある。さらに、PDP 11システム側からみれば通常の端末装置が接続されただけであり、既存のDSM-11でサポートされる。NET/ONE (表2) は、Ethernetコンパチブルであり、マイロプロセッサを装備したプログラマブルなインターフェイス (N I U) を採用している。N I U間でソフト的にバケット交換を行なうバーチャルサーキット機能を備えており、コマンド入力により簡単に端末装置の接続・切断ができる。このコマンドをMUMPSシステム側より発生する事により任意の端末装置を任意のコンピュータに切り替え接続ができ、後に記す端末ネットワーク、障害対策などに活用している。

3) パーソナルコンピュータの利用

新システムの機器構成に関する決定は約一年前に行なったが、まだホスト側の日本語MUNNPSと言語仕様、通信機能の上で互換性のあるものはなく、次のような利用形態を前提にして3台を導入することにした。

①VT100エミュレータを導入し、PDP 11/44の日本語ターミナルとしての利用。

②日本語ワードプロセッサとしての利用。

③高機能簡易言語システムを導入し、クローズドな処理への利用。

③の利用例として、事務局での備品管理のシステム化がある。表形式のデータであること、データ量も10MBのハードディスクで足りること、日々の業務のため、身近に設置されたパーソナルコンピュータを使い、多様な条件検索処理が必要とされること、たの業務とのデータのやり取りもなくクローズドな処理であること、などから最適であると考えられる。まだ日本語ワードプロセッサ機能の利用を始めたばかりであるが、事務吏員の関心も高く、頻繁に利用されている。また、この簡易言語システム利用のコンサルテーションと指導体制をつくることにより、事務局業務のオフィスオートメーションへの道を開くことも期待している。

2-3. 障害対策

新しく導入したミニコンピュータは同一機種で統一し、総合管理システムのハードウェア構成は、他のPDP 11/44の最大規模のものと同等にしている。他のPDP 11/44の障害時には総合管理システムで処理を代替し、端末装置もLANによりこのシステムへ繋ぎかえることにより、障害時のバックアップを行なう。切り替え作業は、約10~15分で可能である。

2-4. ログオン管理システム

前記のように、MUMPSシステム側からコマンドを送ることにより、端末装置の接続先のシステムを切り替えることが可能である。この機能をログオン管理システムに組み込み、端末ネットワークを構築した。そして図2に示すような、LANコントローラ(LAN.C)、LANエータ(LAN.M) アプリケーションプログラムサーバ(AP.S)から成るサイト・アクセス・プロセッサで処理される。Aシステムに接続された端末装置から、接続

先をBシステムに切り替えの指示が出された場合には、AシステムのLAN.CからNET/ONEのNIUに対し、切り替え先のBシステムのI/Oポートを捜し、新たに回路を設定するように指示を出す。回路の設定が完了し、新I/Oポートのアドレスが判明すると、DSM-11のDDP機能を使い、Bシステムにログオン時の全情報とポートのアドレスを送信する。Bシステム上のLAN.Mはこの情報にもとずき必要なアプリケーションプログラムスタートを起動する。このようにして、全端末装置を仮想端末として、全適用業務システムの処理に使用が可能となる(I/Oポートの余裕の範囲内)。

このような方法により、ハードウェア、ソフトウェア、データベースなどのシステム資源の有効利用を図るよう設計している。しかし一方では、データ保護、プライバシー保護の観点から、無制限に利用することは許されないし、システム監査のため、システム資源の利用状況の厳密な把握も必要とされている。このためログオン管理システムのメニュー・スル・ハイパー(MSV)、ログオン・スル・ハイパー(LSV)により(図2)、次のような管理を行なっている。

- ①個人単位に、システムへのログオンの是非を制御すること
 - ②個人単位に、適用業務システムなどの利用の是非を制御すること
 - ③個人別、業務別、端末別、システム別の利用記録を自動的に残しレポートを作成すること
- 利用者は、パスワード、パスワードを入力し、システムでは登録されているパスワード、パスワードと対比しログオンの是非、適用業務システム利用の是非を判定している。またシステム利用記録として7表の月報が作成されている。

2-5. 適用業務システム

適用業務システム(APS: Application System)の外部仕様は第一期システムのもの踏襲し、大幅な追加、拡張は行なわれない方針である。そして、再開発の内容は次の各点に絞っている。

- ①各APSの機能を、各利用部門の業務運用と内容に、より適合させるべく改良する。
- ②コンピュータ・ネットワークによるデータ転送の自動化と効率化、データベースとAPSの再編成など内部仕様の改良と充実をはかる。

例えば、各コンピュータ間では、患者基本情報、検査指示情報などのトランザクションデータをCPU間で転送し適時利用をはかる。また、医事会計処理は従来の窓口計算だけでなく、一ヶ月分の患者データを保有し、データの照会や修正、レセプト作成用の編集までミニコンピュータで行ない、出力処理は高速ラインプリンタを有する汎用中型コンピュータで行なうなど、業務処理運用に最適な負荷分散を図る。

2-6. 移行計画

現在、PDP 11/44 と IBM 4361 コンピュータ、LAN、端末装置の入れ替えなどを完了し、APSは第一期システムのもの新ハードウェア用に修正し利用している。第二期システム用のAPSは、現在開発中であり、1985年2月本稼働を目標に準備を進めている。

3. ローカル エリア ネットワーク利用への期待

LANは、信号線が同軸ケーブル1本ですみ工事の簡素化がはかれること、さらに端末装置の移設や増設の際も工事が少なくすむこと、などの効果を予測して導入した。また、コンピュータ・ネットワークとLANを組み合わせることにより、全端末装置を仮想端末装置として扱うことが可能である。従って利用者からみれば、対象コンピュータを意識することなくシステムが使える、かつ次のよ

うな利点がある。

①限られた端末装置の有効利用が図れる。

②システムを管理する側でコンピュータ間の処理負荷分散を調整しながら運用できる。

③障害時には端末装置群を、他のコンピュータに容易にかつ迅速に切り替えることが出来、障害対策としても有効である。

当システムではLANを、端末装置との間の低速通信回線として使っているが、一般の技術動向として、光ファイバーの採用など高速通信回線向きのLANも実用化されつつあり、画像や文書データ、CPU間のファイル転送も可能となろう。多様な形態のデータをもち、かつ分散処理指向の病院情報システムにとって、LANは有効なツールになると考えられる(4)。

4. おわりに

第二期病院情報システムのハードウェアの導入、稼働は完了し、次の段階として、昭和60年2月本稼働を目標に適用業務システムの再開発を進めている。新システムでは、外部仕様は旧システムのものの改良にとどめ、ネットワーク化、LANの採用、APSとDBの再編成、システム管理機能、標準化など内部仕様の充実に重点をおいた。利用者には直接見えない面であるが、将来の情報処理サービスの充実には欠かせないものと考えている。

参考文献

- (1) 河村 徹郎、北川 護、三嶋 博昭、藤江昭、他：大阪府立成人病センター、病院情報システム、医典社、p.336、東京、1983。
- (2) 河村 徹郎、他：大阪府立成人病センターにおける病院情報システムの評価、第2回医療情報学連合大会論文集、p.111、東京、1982。
- (3) 服部 敏夫、藤江 昭：Log on管理システム、第6回日本MUG 学術大会論文集、p.1、名古屋 1979.9。
- (4) 河村 徹郎：病院内LANと診療支援システム、映像情報 MEDICAL、16、7、p.874、1984。

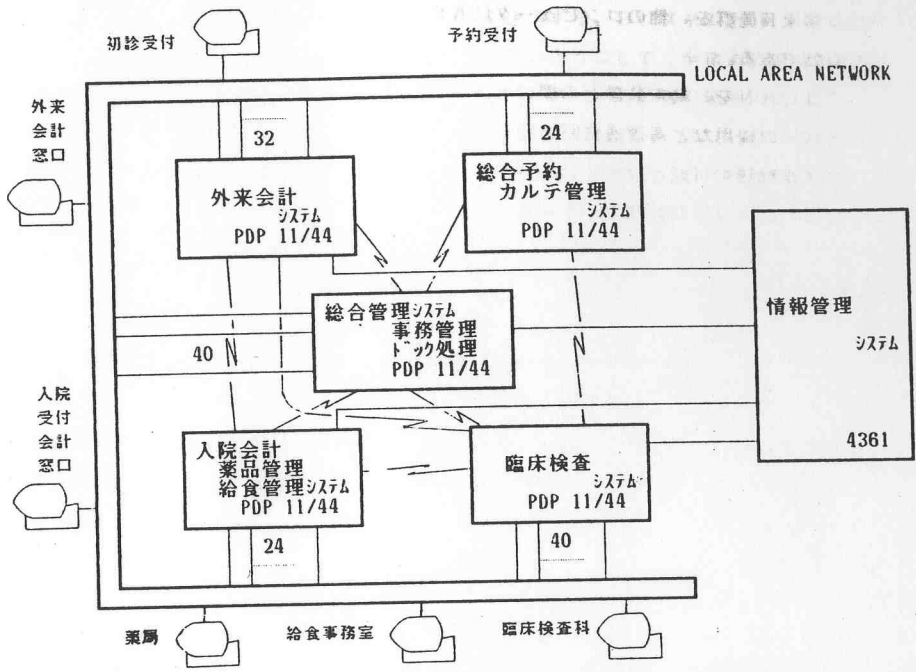


図 1. 大阪府立成人病センターの第二期専用システムの構成

表 1. 第二期専用システムの機器構成

CPU	: DEC PDP 11/44 (1MB)	× 5	
DISK	: RA60 (205MB)		
	外来会計、総合管理	各々×2	
	予約、入院、検査	各々×1	
MT	:	各々×1	
漢字VDT	: × 5	LP(ローカル)	: × 4
漢字ワーケーションVDT	: × 4	LP(リモート)	: × 3
PC9801E	: × 3	漢字リフトプリンター	: × 2
磁気カードリーダー	: × 2	ドットマトリックスプリンター	: × 1
検査分析装置用コンソール	: × 10	ラベルプリンター	: × 1
カルテ管理機	: × 1	オートインキプリンター	: × 1
OS: DSM-11			
LOCAL AREA NETWORK : Ungermann-Bass NET/ONE			

表2. NET/ONEの仕様

項目	内容
伝送形態	ベースバンド
伝送媒体	同軸ケーブル
伝送方式	CSMA/CD
伝送速度	10 Mbps
ネットワーク長	500m/セグメント
NIU数	最大 100 NIU/セグメント

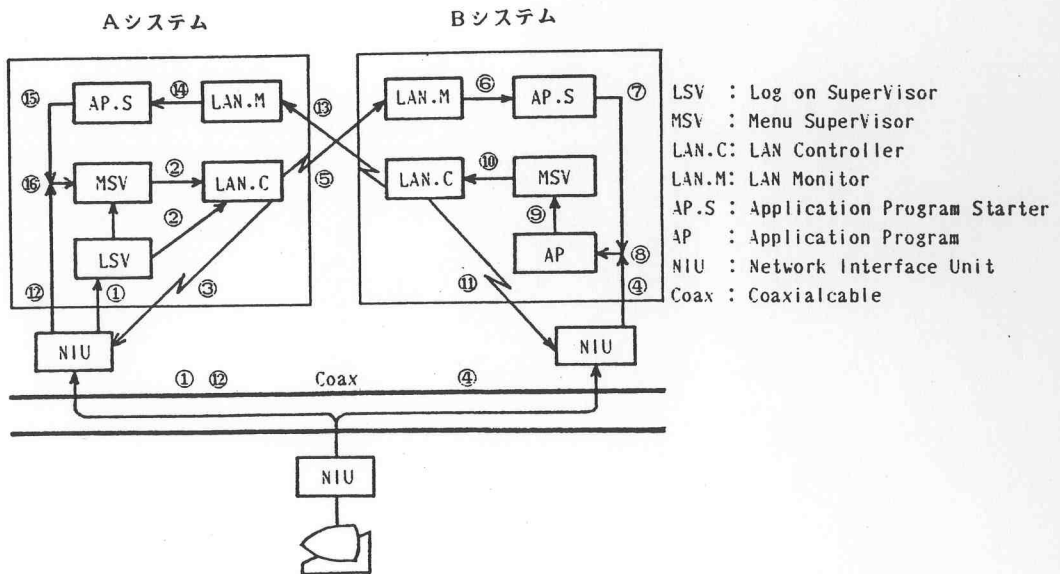
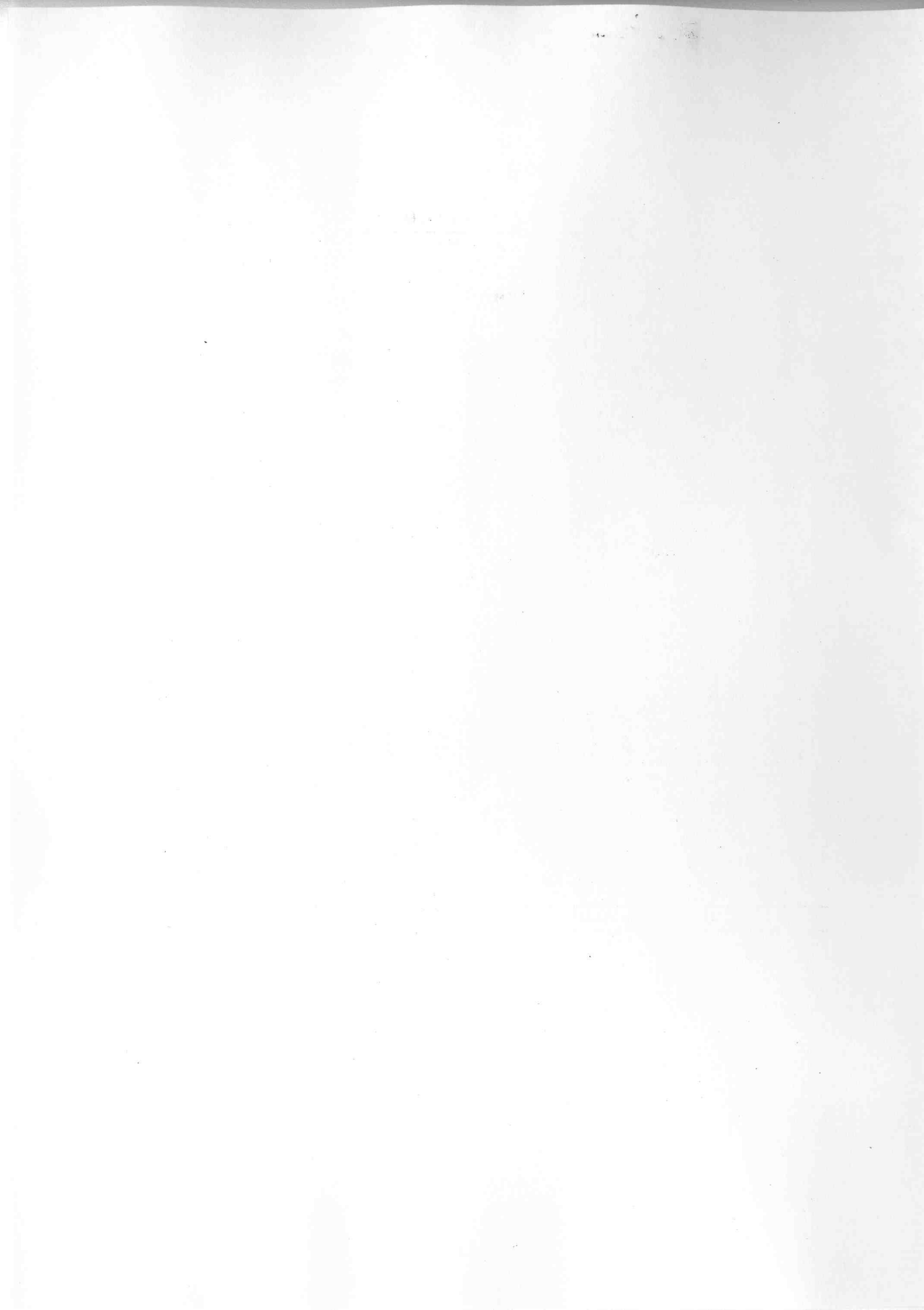


図2. サイト・エクステンジ・プロセッサ(SEP) によるコンピュータシステムの切り替え



「アプリケーション」

座長 八日市谷 隆 (東北大.医)

3 品質管理査定
3.1 品質管理査定の対象
A. 基本設計書

プログラム構成図と全体的な処理の流れの表現の仕方。ファイル設計用紙を用いての概念図と個々のデータの属性、長さ...とノードの書き方について。

B. 詳細設計書

プログラム条件書(使用ファイル群、共通ルーチン群概要説明書)、HIPO design sheet (1), (2), (3), 変数表, 入出力形式用紙とプログラムリストとの関係について。

C. プログラムリスト

プログラム・リストはドキュメントの一部である(プログラム条件書の一部とプログラムに記述すること、Coding規準(ラベル名, 変数名の略名, 制御の方法... etc) について。

D. テスト仕様書・成績書

テスト仕様書(テスト環境, テストデータ操作方法, 起動条件), テスト成績書(予想結果, 判定, 入出力結果リスト...) について

E. 操作手引書

起動条件、ユーとの対話、ステップと利用者がかかる様に書かれていること、GLPIのSOP(標準操作手引書)の一部となる。

項目	内容	検査日	検査者
1	起動条件		
2	ユーとの対話		
3	ステップと利用者		
4	GLPIのSOP		

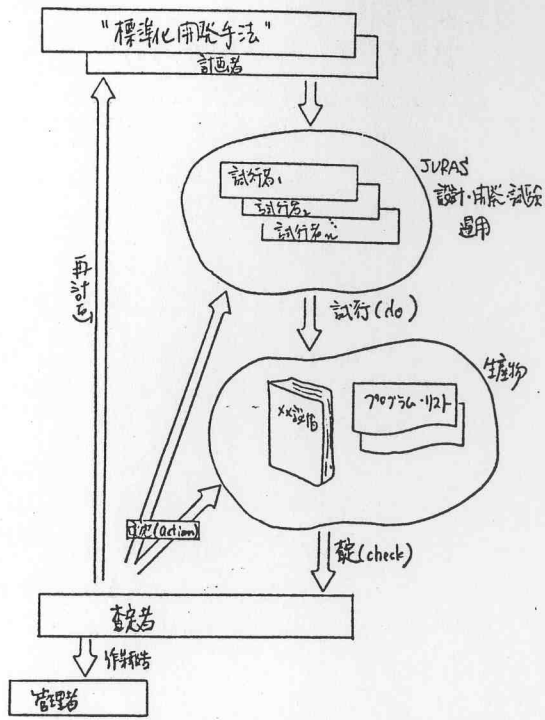


図1-2: 品質管理査定の流れ

図1-3: 品質管理査定 (Quality Control Inspection)

3.2 品質管理の運営

- 計画を立てた時は、計画倒れにならない様に、次の点に注意した。
- a. 計画(SDM)をプロジェクト全員に説明し、理解を深める。
 - b. 再計画が発生した場合は、原本を修正(承認者必要)し、プロジェクト全員に説明し、修正部分を理解させる。
 - c. 試行者(do)と査定者(check)は、査定される生産物に対して同一人物であってはならない。
 - d. 査定者は、全ての試行者に対して、心理的に変化せしめ、問題提起をする事によって質をアップする事に努める。
 - e. 必ず Walkthrough に、QCI report に書き、前もって配印する。
 - f. Walkthrough は 査定者、試行者、開発メンバーの3名以上で行う。

3.3 品質管理査定の報告

QCI report の内容と Walkthrough の結果などは、Walkthrough 対象生産物、査定者、... を記入し、品質管理査定表(Summary)として、プロジェクト管理者に提出し、ドキュメントの出来具合、改訂、他の工程との関係等の7ヶレベルに役立てる資料とする。

表1-1: 品質管理査定表(Summary)

QCI No.		QCI type	内容 (日本語)	予定日	完了日	担当者	評価者	承認者	Walkthrough 実施	Test Management Brief Comment
90	1-2	1-4	Sample Process (ソフトウェア)	10/20/90	10/20	W. M. H.	W. M. H.	W. M. H.	実施	...
91	1-6		実施	...
92	1-6		実施	...
93	1-6		実施	...
94	1-2		実施	...
95	1-6		実施	...
96	1-6		実施	...
97	1-6		実施	...
98	1-6		実施	...
99	1-6		実施	...
100	1-6		実施	...
101	1-6		実施	...
102	1-6		実施	...
103	1-6		実施	...
104	1-6		実施	...
105	1-6		実施	...
106	1-6		実施	...
107	1-6		実施	...
108	1-6		実施	...
109	1-6		実施	...
110	1-6		実施	...
111	1-6		実施	...
112	1-6		実施	...
113	1-6		実施	...
114	1-6		実施	...
115	1-6		実施	...
116	1-6		実施	...
117	1-6		実施	...
118	1-6		実施	...
119	1-6		実施	...
120	1-6		実施	...
121	1-6		実施	...
122	1-6		実施	...
123	1-6		実施	...
124	1-6		実施	...
125	1-6		実施	...
126	1-6		実施	...
127	1-6		実施	...
128	1-6		実施	...
129	1-6		実施	...
130	1-6		実施	...
131	1-6		実施	...
132	1-6		実施	...
133	1-6		実施	...
134	1-6		実施	...
135	1-6		実施	...
136	1-6		実施	...
137	1-6		実施	...
138	1-6		実施	...
139	1-6		実施	...
140	1-6		実施	...
141	1-6		実施	...
142	1-6		実施	...
143	1-6		実施	...
144	1-6		実施	...
145	1-6		実施	...
146	1-6		実施	...
147	1-6		実施	...
148	1-6		実施	...
149	1-6		実施	...
150	1-6		実施	...
151	1-6		実施	...
152	1-6		実施	...
153	1-6		実施	...
154	1-6		実施	...
155	1-6		実施	...
156	1-6		実施	...
157	1-6		実施	...
158	1-6		実施	...
159	1-6		実施	...
160	1-6		実施	...
161	1-6		実施	...
162	1-6		実施	...
163	1-6		実施	...
164	1-6		実施	...
165	1-6		実施	...
166	1-6		実施	...
167	1-6		実施	...
168	1-6		実施	...
169	1-6		実施	...
170	1-6		実施	...
171	1-6		実施	...
172	1-6		実施	...
173	1-6		実施	...
174	1-6		実施	...
175	1-6		実施	...
176	1-6		実施	...
177	1-6		実施	...
178	1-6		実施	...
179	1-6		実施	...
180	1-6		実施	...
181	1-6		実施	...
182	1-6		実施	...
183	1-6		実施	...
184	1-6		実施	...
185	1-6		実施	...
186	1-6		実施	...
187	1-6		実施	...
188	1-6		実施	...
189	1-6		実施	...
190	1-6		実施	...
191	1-6		実施	...
192	1-6		実施	...
193	1-6		実施	...
194	1-6		実施	...
195	1-6		実施	...
196	1-6		実施	...
197	1-6		実施	...
198	1-6		実施	...
199	1-6		実施	...
200	1-6		実施	...

3.4 品質管理査定のツール

品質管理査定をするのは、生産物(ドキュメント)で、静的方法(チェック)と動的方法(計算機)チェックに分かれるが、操作手順書とプログラムリストの一部は動的方法によってチェックが可能である。JURASでは、プログラムリストに対するツールを開発し、Coding規準などを査定するのに利用した。(図1-4: Coding規準チェックプログラム実行結果)

4. 品質管理査定導入の現状と今後について

4.1 品質管理査定担当者

品質管理査定者は、開発者側の開発思想を考慮して step ごとに生産物を check する事により、開発上好ましくない点の指摘が可能でかつ把握出来る。

査定の対象となる順序は phase ごとに round-robin 形式でこなして、再作業量を削減する事に努めた。

4.2 保守体制へ向けたコスト削減

開発メンバーと保守メンバーが異なる場合は、アプリケーション・プログラムとして重要となるのはドキュメントであるから、プログラムの制作の品質と同等、それ以上にドキュメントに品質を設定した事は、保守コスト削減へ向けるポイントになるであろうと考える。

```
NO.=1 Name =AUG121
*****
AUG121 ;
R I,"Y=","Y
K I,"Z=","Z
S S2="(Y,Z)" Y !,"SUS=","S2
S S("X_Z2")="S2S23"
QUIT
NO.=1 AUG121 Progrm
USAHAV3-I-NOTFNDOFF, Not found label+offset
*****
NO.=2 Name =JUTXT2
*****
GOTXT2 ; This is text of GOTJ offset statement NOT
;21-FEB-1984 ;V.1.4A1ZUMI
K
GOSTAT ;
S A=1,S=2,C=3,D=4
IF A=J S GOSTAT
I C=4 G.GOSTAT
GOSTAT ;
S A=1 G=2 G=1
S B=0+1 G=3=4 GEXT1+1
G.GOSTAT
GOSTAT ;
S D=3=3=2
W I," THIS IS TEXT Label=GOSTAT",!
GEXTIT ;
QUIT
; Routine save
SAVE S RSAVE="ZS GOTXT2,GOTXT2" X RSAVE QUIT
NO.=2 GOTXT2 Progrm
GOSTAT ; IF A=J S GOSTAT
GOSTAT ; S A=1 G=2 G=1
*****
```

図14: Coding標準化による保守体制

参考文献

- 1) Gerald M. Weinberg and Daniel P. Freedman
Reviews, Walkthrough, and Inspections
IEEE Transactions On Software Engineering,
Vol. SE-10, No. 1, PP68~72 (JAN 1984)
- 2) Fletcher J. Buckley and Robert Poston
Software Quality Assurance
IEEE Transactions On Software Engineering,
Vol. SE-10, No 1, PP36~41 (JAN 1984)
- 3) 松田亀松 「QCのよみがえり」 (日本実業出版社)
- 4) Glenford J. Myers 「Reliable Software through Composite Design」

健康予測システム

* 田岡達朗 ○ 浅野公美子 ** 柿崎賢一 佐々木一郎

* (株)ライフチェッカー

** (株)高岳製作所

健康予測システムとは慶応大学医学部公衆衛生学教室近藤東郎教授のところで開発された健康教育の手法である。このプログラムはFORTRANで記述されているが、今回このプログラムのデータベースの管理、統計、検索処理をMUMPSで実現したので健康予測システムの内容と合わせて紹介する。

1. システムの概要

成人病のごとく生活習慣の偏りに原因がある疾患については、疾患の発症以前から、習慣の偏りを是正するための対策が必要である。このような状況で、疾患の発症前からの健康管理活動の手法として注目されてきたものが、「健康予測」と呼ばれる健康教育法である。「健康予測」とは、現時点での生活習慣をもとに、その受診者の成人病への発症のしやすさを算出し、それをもとに習慣の偏りによる健康への損失を理解してもらうものである。

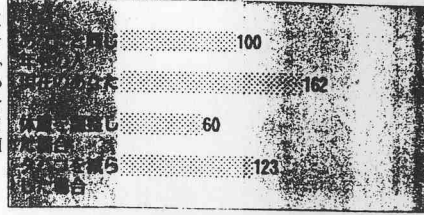
健康予測システムという概念から、健康予測指標として現在次の3項目が使用されている。

- ・第1は、疾病発症危険度予測である。これは、現在の生活様式を続けた場合、脳卒中、虚血性心疾患および悪性新生物の3大成人病疾患に関して受診者がどのくらいそれぞれの疾病にかかりやすいかを示すものであり、同年令の一般集団を100としてその結果を報告する。また、改善可能な危険因子については改善した際の疾病発症危険度の減少程度も示す(図1参照)。
- ・第2は、延命効果である。これは改善可能な危険因子を改善した場合、それぞれどの程度の延命効果(健康で長生きできる)があるかを示したものである(図2参照)。
- ・第3は、健康年齢である。これは、生活習慣をもとにして求めた疾病発症危険度から、何歳の平均的危険度に相当するかを算出したものである(図3参照)。

現在使用している専門の質問表は約100の質問項目から構成されている。質問の一部を図4に示す。これらの質問を大別すると、生活歴、健康診断情報、既往歴と家族歴、行動歴の4つになる。生活歴には、喫煙、飲酒、運動などに関する詳細な質問。健康診断情報には、身体計測、血圧、数種の血液生化学検査結果

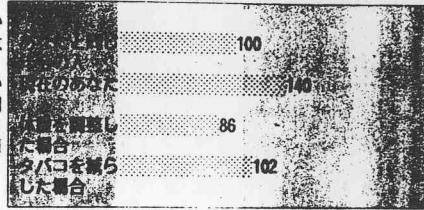
I = 虚血性心疾患(狭心症・心筋梗塞など)のあなたの危険度は

あなたと同じ年齢の人が虚血性心疾患(心臓病)にかかる可能性の平均値を 100 とすると、あなたのその危険度は 162 となっています。
 あなたのライフスタイルは虚血性心疾患を予防する上からは少し問題があるようです。しかし、体重を調整することによって大きな改善効果が期待できます。ぜひ体重を調整することを実行して、次回の値が良くなるようがんばりましょう。
 具体的改善は4の①及び5を参考して下さい。



II = 脳卒中(脳出血・脳梗塞など)のあなたの危険度は

あなたと同じ年齢の人が脳卒中(脳の疾患)にかかる可能性の平均値を 100 とすると、あなたのその危険度は 140 となっています。
 あなたのライフスタイルは、脳卒中を予防する上からは多少の問題があるようです。しかし、体重を調整することによってかなりの改善効果が期待できます。ぜひ体重を調整することを実行して、次回の値が良くなるようがんばりましょう。
 具体的改善は4の①及び5を参考して下さい。



III = 悪性新生物(ガン)のあなたの危険度は

あなたと同じ年齢の人が、悪性新生物(ガン)にかかる可能性の平均値を 100 とすると、あなたのその危険度は 105 となっています。
 あなたのライフスタイルは悪性新生物を予防する上からは、あまり問題がないようです。しかしながらタバコを減らすことによってかなりの改善効果がありそうです。ぜひがんばってタバコを減らすことを実行し、次回の値をもう少し良いものにしましょう。
 具体的改善は4の②及び5を参考して下さい。

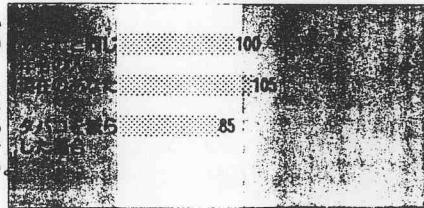


図1. 報告書の印字例(1)

あなたの延命指数をお知らせします

あなたのライフスタイル、すべての環境条件を考慮に入れて、それぞれのリスクファクターを改善したとき、どのくらい余命が延びるかを計算してみました。
 兼松さんの場合は体重を調整することによって1.8年も余命が延びるという結果ができました。
 ぜひとも以下の注意を参考にして体重を調整することを実行するようがんばりましょう。

- 体重を調整した場合 1.8年
- タバコを減らした場合 1.6年
- アルコールを減らした場合 1.1年

図2. 報告書の印字例(2)

などの項目。行動歴のなかには、虚血性心疾患の危険因子としての Type A 行動に関する特性を検出する質問群がある。

健康予測システムのデータベースは<危険因子スコア>と<理想的健康水準の

設定の2種の情報にさらに分けられる(図5参照)。前者の<危険因子スコア>は、疾病発症危険度を求めるのにもっとも重要な部分であり、疾病の発症へ寄与する危険因子の程度を数値データで収納している。後者の<理想的健康水準の設定>は、健康改善の目標をまとめたものである。この理想的健康水準は、疾病の発症危険度が最小になることを設定しており、これらの数値を用いて、生活様式を

改善したと仮定した時の健康に及ぼす効果を算出する。演算部分は受診者の質

まず、あなたの普段の生活についてお答えください。

① 現在紙巻タバコを吸いますか

- 1=はい
- 2=いいえ(④の質問へ)

② 1日どのくらい紙巻タバコを吸いますか

- 1=10本未満
- 2=10本から19本
- 3=20本から39本
- 4=40本以上

③ 紙巻タバコの煙を吸い込みますか

- 1=全然吸い込まない
- 2=わずかに吸い込む
- 3=普通に吸い込む
- 4=深く吸い込む

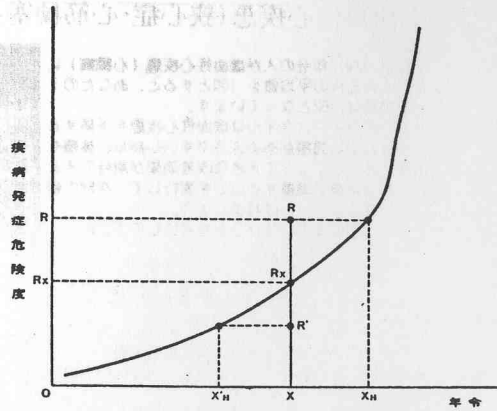
④ この1年間、吸っていた紙巻タバコの種類は何ですか

- 1=フィルターなし(両切りタバコ) (ショートピース・シンセイ・ゴールデンバット等)
- 2=フィルター付 タールやニコチンの多いもの (ロングピース・ショートホープ・ハイライト等)
- 3=フィルター付 タールやニコチンの中程度のもの (セブンスター・キャビン・マイルドセブン等)
- 4=フィルター付 タールやニコチンの少ないもの (キャスター・ジャスト・カレント等)

⑤ 紙巻タバコを吸いはじめたのは何歳の時ですか

- 1=14歳以前
- 2=15歳から19歳
- 3=20歳から24歳
- 4=25歳以降

図4. 質問項目の例



危険度 R が R_x より大きい時、健康年齢 X_H 才は、暦年齢 X 才より高くなる。
危険度 R' が R_x より小さい時、健康年齢 X_H 才は、暦年齢 X 才より若くなる。

図3. 健康年齢

問票に従いシステムの中央コンピュータに組み込まれたプログラムにより、データベースを活用して健康予測指標を算出する。

疾病の原因を追及するため、現在まで相当数の疫学調査が行われてきている。循環器に関する代表的疫学調査として、米国の Framingham heart study、Seven country study などが挙げられる。

最近の疫学調査では、多くの因子を同時に調査、解析し、因子間相互の影響についても検討することを目的としたものが多くなってきている。それと関連して多因子を同時に解析し、より精度の高い疾病の発症と原因の関係を示す予測式を求めするため、多くの統計手法が

開発されてきている。この代表的な手法が、多重ロジスティック関数で近年この関数を用いて Framingham heart study が解析され、その結果が健康評価に有用であることが示されている。次に算出された健康予測指標を用いて報告書を作成し受診者に個々に返送する作業である。報告書は、

個人宛に戻す個人報告書と、対象集団の健康予測結果をまとめて地域や職域の健康管理を担当する医師やパラメディカルワーカーの人々に戻す集団報告書の2つに分けられる。報告書の一部を図1、図2に示す。集団報告書は、対象集団の健康特性をまとめたものであり、今後の保険活動で何をすべきか、何が効果的であるかを理解するのに役立つ資料を提供するものである。一つの例として、健康予測の結果、その集団が全体として虚血性心疾患に対する危険度が高い集団であることがわかれば次回から健康診断の項目に負荷心電図等の検査を追加して、効果的に疾病の早期スクリーニングに努めることが可能になる。

演算プリントアウトのデータ処理は、1日7時間稼動で500人、月20日間で10,000人、1年間に120,000人を予定している。

健康予測システムの取り扱う情報のなかには個人の秘密に属するものも少なくないので、秘密の保護という問題があるが、それについては、十分注意している。個人を認識する情報は、質問票の情報と分けて扱い、ファイルも別々に管理しているので、外部から情報を閲覧することはできない。

2. システム構成

図6にシステム構成を示す。ホストCPUには米国DEC社のVAX-11/750を使用し、FORTRANをVAX-FORTRAN(FORTRAN77)MUMPSを米国InterSystems社のM/VXを使用している。入力媒体としてはフロッピーディスクからデータを入力する方式と、漢字ターミナルKJ-100からオペレータが直接入力する方法をとっている。報告書の出力には漢字プリンターKJP-150を使用している。

3. MUMPSとFORTRANとのリンク

本健康予測システムはデータの技術計算処理をFORTRANで行い、その処理結果の管理、統計、検索処理、データの入力処理、報告書の出力処理はMUMPSで行っている。基本的にMUMPSとFORTRANとのリンクはVMS上

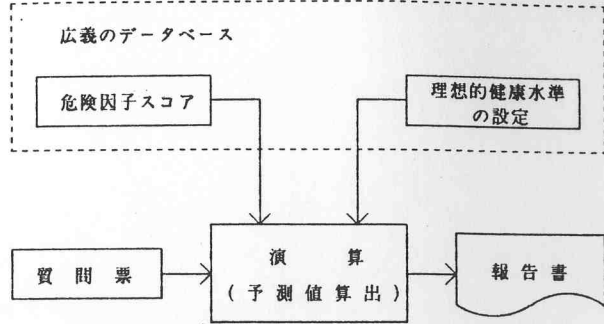


図5. 健康予測システム

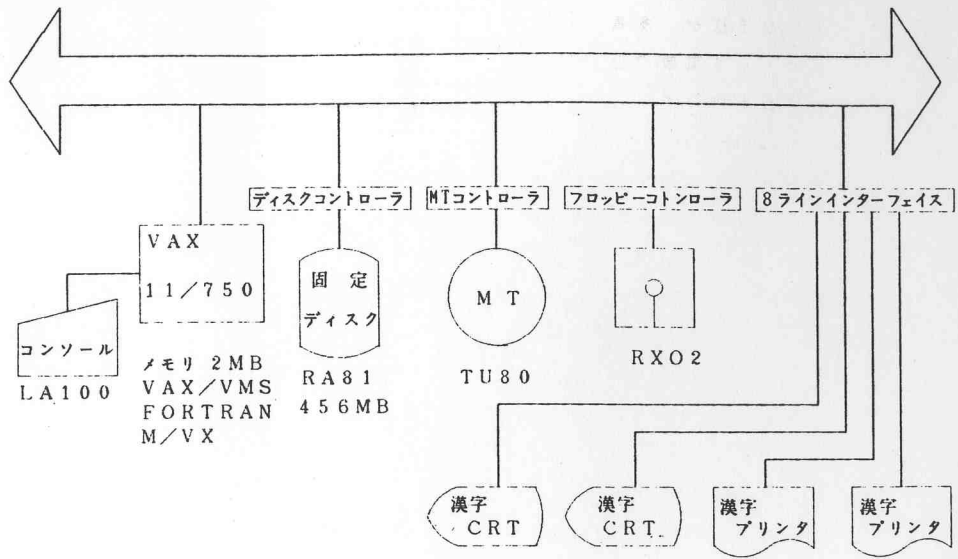
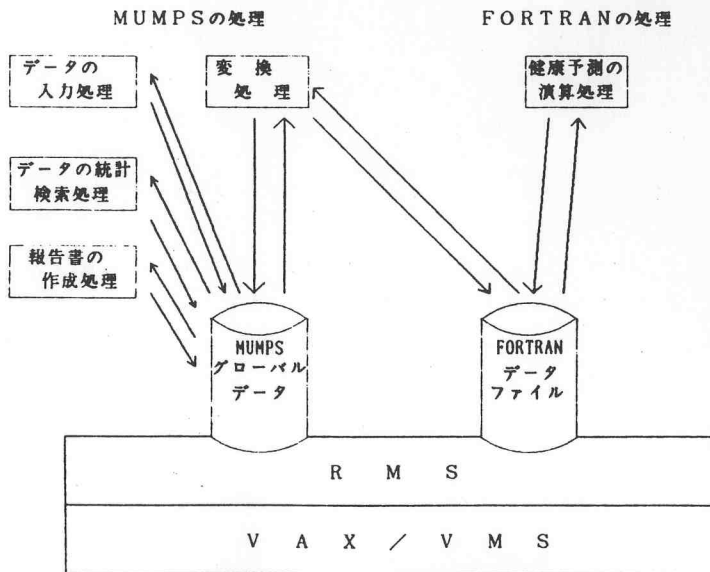


図6. システム構成図

のファイルを紹介している。MUMPSとFORTRANとのリンクを図7に示す。入力データはオペレータより入力された場合、まずMUMPSのグローバルデータとしてセットされ変換プログラムによりFORTRANのインプットファイルにセットする。フロッピーディスクから読みこまれた入力データは変換プログラムにより、直接FORTRANのインプットファイルにセットされる。このあ



とFORTRANの演算処理を行い処理結果はVAX/VMS上のデータファイルにセットされる。MUMPSの変換プログラムを使って、このファイルからデータを読みこみグローバルデータとしてセットする。このグローバルデータを使い、報告書の作成、データの統計、検索等の処理を行う。

これはMUMPS、FORTRANの各々の言語の長所を十分に引き出しておりディスク容量についてもMUMPSの可変長レコード処理で十分パフォーマンスを上げることができる。FORTRANの演算結果をもとにしたデータベースによりMUMPSで統計、検索など柔軟なデータベースの活用をすることができる。

参考文献：

「健康管理における健康予測システムの開発」

近藤東郎、吉田勝美、桜井裕 著より

（「日本医師会雑誌」第91巻 第10号掲載）

「健康予測は可能か」近藤東郎、吉田勝美 著より

（からだの科学・臨増「成人病の事典」'83.2月号 掲載）

「健康予測」-健康管理における可能性と応用について- 82年

近藤東郎、吉田勝美、古川武温 著より

「健康予測とライフスタイル」近藤東郎 著より

（からだの科学・別刷 NO.107 '82.9月号 掲載）

「健康診断と健康予測」近藤東郎 著より（「健康教育」'83.12月号 掲載）

藤田学園保健衛生大学病院における
MUMPS検査システムの特徴について

○落合 潤一 秋山 富英 白石 正司 鴻上 慶次郎 勅使河原 深雪
厚味 高広 伊藤 円 岡島 光治 藤田 啓介

藤田学園コンピュータ管理センター
藤田学園保健衛生大学病院
愛知県豊明市沓掛町田楽ヶ窪1番地の98

I. はじめに

藤田学園保健衛生大学病院では、昭和52年より、ミニコンネットワークによる病院総合情報システムが自主開発され稼働している。このシステムは情報の発生源入力を第1目標に開発され、医療に従事する全ての職員が関与し、医師の指示情報より発生する各種の業務から、診療費の請求業務までを含む一連の情報処理システムからなっている

今回、各診療室・病棟ナースステーションにおける医師の検査依頼入力から検査結果報告までの検査システムの流れ及びその特徴について報告する。

II. 外来における検査依頼

医師は診察室または病棟に設置されたディスプレイ装置より各種の指示を入力する。検査依頼入力は病棟システムが予約入力機能を持っていることを除けば、外来、病棟とも同一機能がある。

- 検査依頼画面は ① SET検査 画面
② 単項目検査 画面
③ 依頼項目参照画面
④ コメント入力画面

の4種に大別されており、目的に応じて選択する。

1. SET検査 画面

検査指示を行うにあたり、医師のKey操作をできる限り少なくする為、診療科毎にそれぞれの専門分野にあわせて、30~40種のSET検査を作成した。CRT画面左半分にSET検査名を表示し、右半分にSET検査と併用して良く使用される検査項目を約80項目表示し、特殊な検査を除けば、ほとんどの検査依頼が、この画面にて操作できるようになっている。画面上はSET検査名の表示であるが、その内容は、必要な時、いつでも簡単なKey操作にて参照できる。

また外来システムでは皮膚科、耳鼻咽喉科、眼科、産婦人科の5診療科については指示される項目が限定されるので、検査項目のみならず、処置、X線、注射等の項目が混在した画面で構成されている。

2. 単項目検査 画面

この画面は、保健診療で認められている総ての検査項目を尿検査、血液検査、細菌検査、血清検査、生化学検査、病理検査、生理検査、R I検査、緊急検査の9部門にわけてあり、医師は目的とする部門を任意に選択して、意図する検査項目をみつめる
なおSET検査画面で、すでに入力してある項目は、単項目画面にも表示されるので、二重に入力することはない。

3. 依頼項目参照画面

特定Keyを入力することにより、前後1週間の依頼内容が表示され依頼項目の認ができる。また、この画面においては、前回の指示内容と同じ検査を実施する場合日付、曜日いずれかを指定すれば、新規依頼となる。

4. コメント入力画面

検査指示に伴う各種コメントは、身長、体重、病名等の患者コメントと検体種別、採取量、測定時間等の検体コメントの2種があり、検査項目により任意に選択できる
なお妊娠反応の希釈倍率、負荷試験の時間等特定検査項目については、項目選択時にコメントを入力するようになっている。

III. 検査システムの運用

1. 外来患者受付

患者は診療終了後、検査受付に行き受診カードを提示する。受付では、カードをI Dカードリーダーにて読み込ませ、外来システムより患者の検査依頼情報を検査システムに取り込み、CRT画面に依頼項目を表示する。表示された検査項目のなかで、当日検体採取不能な項目を予約検査として、別途処理し、検査受付業務を終り、検体採取に必要な検体ラベルを発行する。

2. 病棟患者受付

病棟患者の場合、ほとんどの検査指示は前日までに入力されており、看護業務との関連より前日に各病棟単位に採血採尿指示一覧表および検体ラベルが検査部にて発行され、各病棟に送付される。看護婦はその指示にもとづいて検体を採取し検査部へ提出する。検査項目の追加、取り消しなどの情報は採血採尿指示表に記録しておく。

3. 検体ラベルと検体採取

検体ラベルは検査依頼項目に応じて採血量、採尿量および採血管の種類などが自動判別され、それぞれ必要な枚数だけのラベルが発行される。

ラベルには受付年月日、患者番号、氏名、性別、年齢、採血時注意すべき事項、採血管の種類が印字されており、検体採取時に生じやすい検体の取り間違いや分量不足などが防止できる。

4. 検体受付

病棟から検査部へ送付された検体および採血、採尿室にて採取された外来患者の検体は ①生化学、血清、R I 検査 ②血液検査 ③一般検査 ④緊急検査 ⑤その他（細菌検査など）の5部門のいずれかの検体受付にて受理する。

検体受付では、外来、入院の別なく到着順に検体を受け、患者番号を入力し、分析すべき検体数と量および検査項目を再確認する。つぎに、その患者に対する検体番号が割り振られ検体受付ファイルとワークシートファイルが作成される。

5. 血清分取分注処理

生化学、血清、R I 検査については、検査依頼項目に基づいて分取分注情報を作成し自動分取分注装置に転送する。この装置は患者血清を①自動分析装置1、②自動分析装置2、③電解質、④血清検査、⑤R I 検査用の最大5系列のサンプルカップに1本1本、可変量的に分注していき、残余血清は用手法検査に用いる。

残余血清は用手法検査に用いても、なを1ml以上の血清が残るよう採血量が指示されているので、この残余血清を1ヶ月間保存して、必要な再検査に使用する。

なを、この自動分取分注装置はオフラインでも、分取分注の指示を入力する機能を備えているので、検査CPUが故障した場合でも、この装置に直接指示を入力して作業を実施させることができる。

6. ワークシート

個々の技師は測定を開始するに当たり、作業区分、あるいは機能区分に分けられたワークシートを作業状況に応じて出力する。ワークシートは ①一般 12種 ②血液 8種 ③細菌 5種 ④血清 6種 ⑤生化学 18種 ⑥R I 2種の計51種がある。

7. オンライン接続自動分析装置

オンライン接続された自動分析装置の測定結果はCPUにリアルタイムに送られ、技師により結果の確認がなされる。確認が終了した後、これらの結果より結果ファイルと報告ファイルが作成される。

8. 結果入力

用手法検査の結果は技師により入力される。この際、測定結果に異常がないかの確認をするために前回値比較を画面上で行うことができる、これは過去3ヶ月あるいは過去5回の検査結果（ワークシート単位）を画面表示し異常結果について画面に異常であるとの表示をする。これにより検査結果の妥当性を容易に確認することができる。また、複雑な計算を必要とする結果については実測値をそのまま入力すれば自動的に計算がされるので、計算ミスを確実に無くすることができる。前回値比較および演算の確認された結果より、結果ファイルと報告ファイルが作成される。

9. 前回値比較

オンライン接続された自動分析装置より自動取り込みされた結果の妥当性を調べるために、結果入力時と同じ前回値比較を行う。

異常結果については、画面あるいはプリンター出力し確認することができる。

10. 報告書

自動分析装置の結果処理が終わった後、中間報告書を出力して各診療科、病棟へ提出する。中間報告書は自動血球算定機、生化学自動分析機、電解質自動分析機の結果で構成され血球算定結果は検査を依頼した当日中、生化学検査結果は当日午前10時までには受付された検体と、前日に受付された検体の結果を一覧表として提出している。また依頼された検査の結果がすべて入力された後、最終報告書が出力され提出される。

11. 結果参照

過去1ヶ月間の検査結果を患者のIDと検査の依頼日を入力することにより画面で参照することができる。そのため診療科、病棟より結果の問い合わせがあった場合、すみやかに画面で確認して結果を報告することができる

IV. システムの特徴

1. 本システムは病院の医療情報システムと直結しており、検査依頼、検体収集、検体受付、血清分離分注作業、自動分析装置からのデータ収集・解析、報告書作成、結果参照などの検査全体にわたっており、検査業務の省力化とともに正確化・迅速化に寄与している。
2. 検査依頼は、各診察室設置のCRTを用いて医師が入力するため、検査伝票を発行する必要はなく、転記によるミスがなくなった。依頼入力は全ての検査項目をそれぞれ入力しうるが、簡便のため、各医師の専門分野に応じて、検査項目がSET化されており、ワンタッチにて入力することが可能。
3. 検体の採取および分注時に生ずる人的ミス（検体の取り違い・採取量不足・肝炎ウイルス等の感染）による障害を、依頼情報にもとずいて、採取必要な検体数と採取量を指示する検体ラベルの発行と自動分取分注装置をオンライン接続にすることにより防止した。
4. 各種の大型自動分析装置はCPUとオンライン接続して分析結果の自動読み取りを行い、手作業をできる限り少なくし、結果報告も印字報告として、転記ミスが起こり、時間のかかる手書き報告の省力化・正確化を実現した。
5. 総ての検査結果は、過去1ヶ月間保存されているので、直接診療科設置のCRTにて参照でき、検査部への電話等による問い合わせを不要とした。

1984年度 第11回 日本マンブスユーザーズグループ学術大会
「アプリケーション」 ④

藤田学園保健衛生大学病院におけるMUMPS薬剤システムの特徴

○竹林和行 奥村義金 原田治良
岡島光治
服部敏郎 山内 実 藤田啓介

藤田学園保健衛生大学病院薬剤部
藤田学園保健衛生大学医学部
藤田学園コンピュータ管理センター
愛知県豊田市沓掛町田薬ヶ窪1-98

藤田学園保健衛生大学病院では、昭和52年より総合医療情報システムが稼働している。このうち、各診療科・病棟における処方並びに注射薬のオーダーは処方箋・注射箋として薬剤部に出力されオンライン処理後、入在庫データとして取込み、在庫管理を行っている。

I システム構成

システムの構成を図1に示す。図1の左は外来システム、右は入院システムを示し、在庫管理は入院システムを用いて行なっている。

各機能の概要・データの流れ及びその特徴を以下で述べる。

II 外来システム

1. 内外用薬

医師の1患者のオーダー終了時処方オーダーが存在すれば、処方箋として直ちに薬剤部に出力され調剤される。

処方内容の問い合わせは電話にて行われ、その後変更等が行われる。

処方箋の枚数計算及び使用実績の累計は、バッチ処理で行う。

処方歴を画面にて参照することができる。

2. 注射薬

注射薬は、外来においては定数配置制であり、その補充はバッチ処理により得た科毎の集計表により行う。定数外の薬品については、オーダーの発生時に薬品請求伝票にて払い出しが行われる。

注射箋の枚数計算及び使用実績の累計は、バッチ処理で行う。

処方歴を画面にて参照することができる。

3. 薬品出納管理

内外用薬・注射薬共に、出力された各集計表をもとに払い出し情報として入院システムの在庫管理システムに入力し、調剤室及び薬品庫の出庫データになる。

■ 入院システム

1. 内外用薬

入院処方箋は調剤業務の都合上、定期処方・臨時処方・緊急処方・退院処方箋に分けられ、各々のスケジュールに従って出力され調剤される。

処方内容の変更等は、外来システムと同様である。

処方箋の枚数計算及び使用実績の累計も、外来システムと同様である。

処方歴を画面にて参照することができる。

2. 注射薬

注射薬の払い出しは、先払い1本渡して行っている。

医師のオーダーは、1手技毎にその実施期間が入力され、実施前日の午前7時から定期薬として処理が開始され定期注射箋が薬剤部へ出力される。この処理と並行して、薬品庫からの棚出し用のワークシートが病棟別に出力される。

定期薬処理からはずれたオーダー及び取り消し・変更のオーダーは、臨時薬として臨時注射箋が出力される。払い出しの必要な薬品については(+)、返品しなければならない薬品については(-)の符号が、各々の薬品の数量の頭に記される。

一部の薬品については、緊急定数薬として定数配置しており、事後入力により出力された注射箋に従って払い出された薬品で補充する。又、インスリン・ヘパリン等の分注する薬品については、薬品請求伝票により払い出す。

3. 薬品出納管理

各々のデータは、随時オンライン処理され出庫データとして在庫管理システムに送られ、在庫日報等に反映される。

Ⅳ 薬品在庫管理システム

1. 薬品庫在庫管理

入庫データは、納入伝票・返品伝票の入力及び返品注射箋の入力により作成する。

出庫データは、内外用薬においては薬品請求伝票、注射薬においてはオンラインデータの取り込み及び薬品請求伝票・破損亡失伝票の入力により作成する。

各々のデータは、在庫の情報データとして処理すると同時に、日毎のデータとして各種データファイルに格納され各種月報用のデータとなる。

2. 調剤室在庫管理

受け入れデータは、倉庫からの出庫データを受け入れ数とする。自家製剤については、製剤室からの払い出しデータを受け入れ数とする。

払い出しデータは、各診療データからバッチ処理により得た使用実績を払い出し数とする。

3. 発注業務

基本的には月初めに一括発注を行い、必要に応じ随時追加発注を行う。

毎日在庫日報をチェックし、必要に応じて在庫月報の出力を行ない、又、薬品動向照会画面を参照して発注品目及びその数量を決定する。

4. 統計処理業務

オンラインからの取り込みデータ及び薬剤部からの入力データより作成されたデータファイルより、各種統計資料を作成する。

薬品動向一覧、薬品使用実績、業者別納入一覧、科・病棟別払出一覧、処方箋・注射箋集計表等である。

5. 在庫管理情報照会

発注の決定、在庫調査等の支援機能として、各薬品についての動向を画面又はハードコピーで参照可能としている。

薬品毎の入出庫状況（当日・当月・前月・当年度の値）

在庫数0%の薬品リスト

返品注射薬リスト

破損・亡失薬品リスト

V システムの特徴

本システムの特徴としては、医師オーダーがオンラインで在庫管理のデータとして取り込まれ、当日在庫が随時参照可能であること。

本院にて不採用の薬品のオーダーはできないこと。

処方薬の転記ミスを防ぐことができること。

各薬品についてのデータ入力及び参照がコードレスで行えること。

払い出し業務における払い出し数の手計算が不要であること。

在庫管理に関する帳簿の記入・整理が不要であること。

未来オーダーの参照が可能であること。

MUMPSシステムである為、機能拡張が容易であること等が上げられる。

図-1 システム連携図

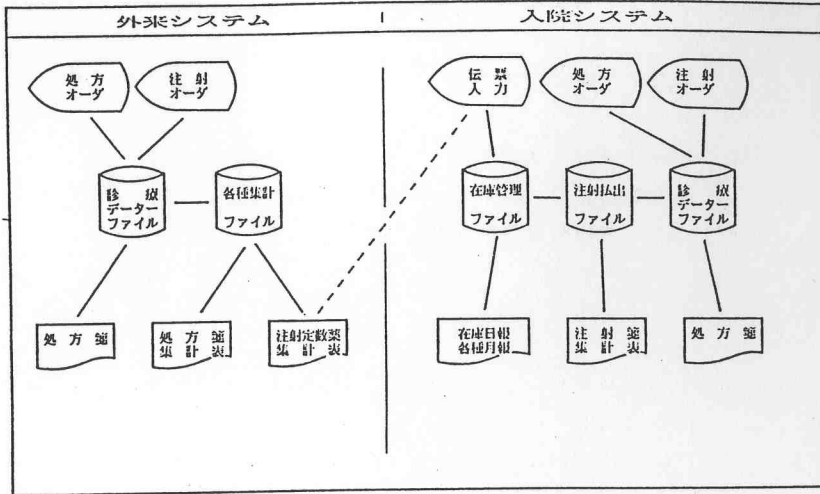
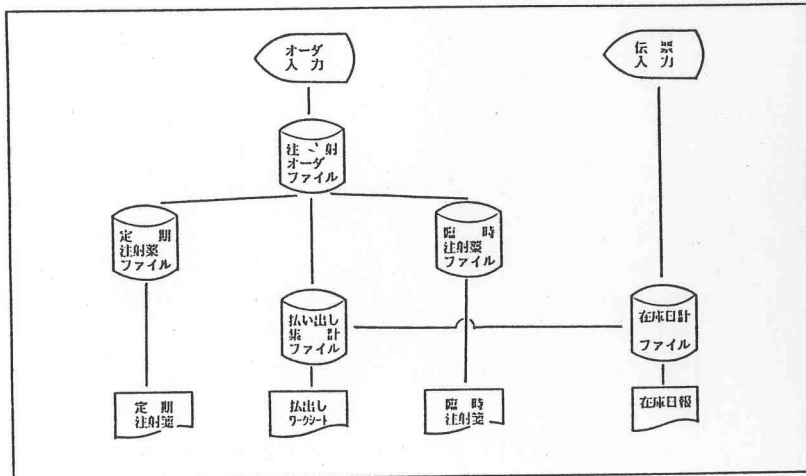


図-2 注射データの流れ



藤田学園保健衛生大学病院における病棟看護業務とMUMPSシステムの現状

○ 荻原美奈恵 松原健次 藤井美登璃 今坂とみ江 大谷元彦 藤田啓介

藤田学園保健衛生大学コンピュータ管理センター

1 藤田学園保健衛生大学病院看護部

〒470-11 愛知県豊明市香掛町田楽ヶ窪1-98

要約

本大学病院では、国内で数少ない、医師が全ての指示をナースステーションの端末機で入力するシステムが稼働している。

情報発生源入力方式による病棟システムは、S53年から看護婦による食事・転科・転室・外泊・外出を始め、S57年から医師指示入力を開始している。

1. はじめに

近年の技術革新は目ざましく、特にコンピュータを中心とした情報処理機器の進歩は著しいものがある。同時に、医療の高度化、専門化に伴ない、情報の量は飛躍的に増大し、その処理は従来の入力のみ方式では、もはや追いつかなくなりつつある。現に、CTを始めコンピュータを内蔵した各種検査機器、監視装置、病院管理システム（料金計算、レセプト作成、在庫管理、人事管理など）が次々と導入されている。

看護面におけるコンピュータ利用は、残念ながら殆ど普及していない。本大学病院はMUMPS言語を使用して、S53年から看護婦による食事・転科・転室・外泊・外出入力を行なっている。病棟看護システムの現状と将来計画について述べたい。

2. 病棟システムの現状

図1～図3はディスプレイ画面における病棟入力の状況を示す。いずれかの番号を押すと対応する業務の入力画面が次々に表示される。入力方法は対話形式で、入力しなければならないもののガイドが表示されるので、それに従って入力が進み、ほとんどが数字をキーインするのみで入力が進む。ディスプレイ装置は医師IDおよび患者ID入力のための穿孔カードリーダーを持つ。

図4は、病棟を中心とした情報の流れを示した図である。医事課で入院登録の後に、はじめて病棟で入力が可能となる。

『検査』の指示入力により、検査前日14時で一度締切り、一括出力され、16時頃各病棟に“検査一覧表”“検査伝票”“検体ラベル”が配布される。検査伝票を基に、ラベルを用い検体容器を作成する。検査一覧表は申し送り等に利用している。

『放射線』の指示入力により、検査時病棟に放射線部から電話で患者の呼出しがあり又はポータブルで病棟にて撮影している。

1. 医師指示
2. 看護業務

図1. 病棟第1画面

- 医師指示
1. 検体検査
 2. 生理検査
 3. 放射線検査
 4. 内服薬
 5. 注射薬

図2. 医師第1画面

- 看護業務メニュー
1. 入院通知
 2. 食事通知
 3. 転科・転室
外泊・外出
 4. 退院通知
 5. 処置確認入力
 6. セット食事
 7. 重症看護登録
 8. 患者情報紹介
 9. 文書料・健診

図3. 看護業務第1画面

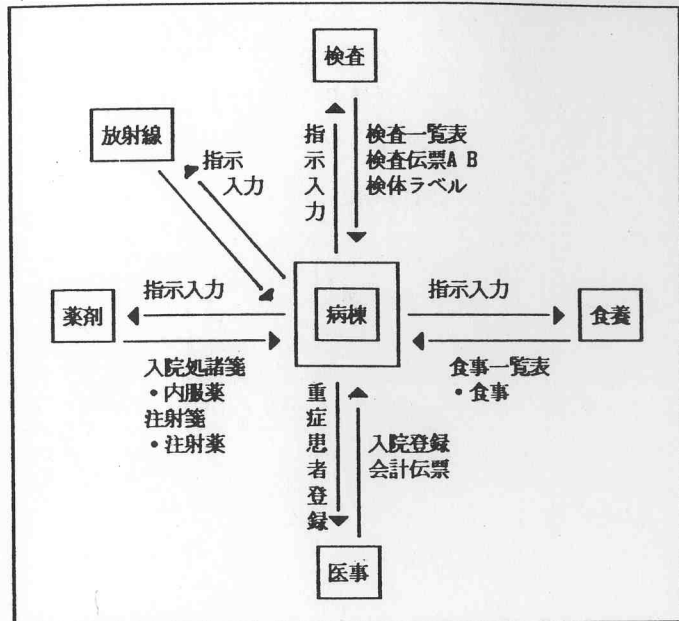


図4. 病棟と各関連部門

『薬剤』の指示入力により、“内服薬と処方箋”“注射薬と注射箋”が払出される。払出すタイミングとして、緊急は電話とともに即時、臨時は8時から17時までは随時、定期は約束日としている。

看護部で入力している、入院通知・食事通知の入力により、病棟食事別の“食事一覧表”が事前に出力され、配膳に使用している。

また、入院・退院・転科・転室の入力は患者の所在を明らかにし、24時現在の患者数を把握して“日報”・“月報”・“病棟利用状況”を作成している。

新採用者の研修の一部にコンピュータ操作が含まれ、看護婦全員が入力操作を習得し誰でも入力できる。

3. 病棟看護業務の実態

今後の看護業務の電算機利用を予測し、以下の調査を実施した。

本大学病院における看護内容区別従事時間は表1の如く、△はコンピュータ移行可能と考えられる項目であり、×は移行不可能と考えられる項目である。表2は移行可能と考えられる看護業務を示したものであり、延2681分(12.2%) + αと考えられる。看護内容区別従事時間のうち13.9%を占める報告・連絡は、申し送り時間がほとんどを占め、平均92.9分であり、患者一人あたりの、一日合計申し送り時間は平均2.2分であった。申し送り内容の分類は表3に示す。報告・依頼・相談のうち、コンピュータ移行可能と考えられる報告は89.6%、不可能と考えられる相談は0.2%であった。看護業務

の中心となるカードックス利用度は表4に示す如く、内服・検査・注射の医師指示がほとんどを占めていた。このことは医師の入力した情報を、病棟でかなり利用可能であることを示している。以上のことから、看護業務のうちコンピュータ移行可能と思われる報告・連絡・記録・単位管理の仕事、及びそれらを活用するカードックス、申し送り、看護サマリーのコンピュータ化が考えられる。

表1. 看護内容区分別従事時間(3日間)

×・身の回りの世話	12.0%
×・患者とのコミュニケーション	1.5%
×・ベットサイドの治療処置	27.0%
△・記録	17.9%
△・報告・連絡	13.9%
・教育・オリエンテーション	4.5%
・ミーティング・カンファレンス	0.6%
△・単位管理の仕事	9.9%
×・病棟外への移動・移送	4.0%
・その他	8.7%

表2. 移行可能な看護業務

	2681分	←→12.2%
		(100%)
・食札・食事箋	28分	1.0%
・病棟日誌の一部	58分	2.1%
・翌日手術検査表作成	156分	5.8%
・勤務表作成	97分	3.6%
・伝票	523分	19.5%
・薬札	201分	7.5%
・連絡・報告(50%)	728分	28.4%
・薬品管理	380分	14.2%
・検体採取	478分	17.8%

表3. 申し送り内容の分類

	報告	依頼	相談
バイタルサイン	20.0	—	—
観察事項	39.0	1.4	—
処置・治療	3.0	0.8	—
検査	6.8	1.6	—
薬剤・注射	7.2	5.7	—
患者指導	0.5	—	—
他科受診等の指示	1.8	0.8	—
その他	10.0	0.1	0.2
	89.6%	10.4%	0.2%

表4. カードックス利用度

95分間	—100%
・与薬のため	31%
・検査	28%
・注射	25%
・その他	18%

4. 将来計画

前に述べたように、本大学病院では、医師が全ての指示をナースステーションの端末機で入力するシステムが稼働している。

これらの入力されたデータの看護サイドでの利用を含め看護のシステム化を考えている即ち、

A. 患者管理に関するもの

- 1) 指示の時系列参照
- 2) 医師指示の、実施・未実施リストの出力
- 3) 基本看護入力による看護業務スケジュール表の出力
- 4) 看護観察記録情報の入・出力と参照
- 5) 看護歴の作成
- 6) 栄養管理システム

B. 病棟管理に関するもの

- 1) 看護勤務表の作成

- 2) 物品管理
- 3) 物の貸し借りシステム
- 4) 患者に必要な看護度の把握と看護力の適正配置
- 5) 申し送り業務における利用

C. 教育に関するもの

- 1) 処置手術の手順書
- 2) 看護手順書

現在そのための基礎調査と現状分析の作業を進めつつ次期看護システム計画にはいっている。

次期導入されるコンピュータは漢字が扱え、端末機そのものがマイコンの機能をもつものを計画している。したがって、フロッピー利用により、患者指導等病棟独自の巾広い看護婦による患者指導への活用が容易にできると考えている。

5. おわりに

コンピュータのもつ機能、すなわち大記憶容量、高速な通信能力、複雑な分類と統合能力、そして印刷機能を上手に利用すれば、看護の質の向上、管理の強化と効率化が図れると確信する。

看護へのコンピュータの導入は、経済的・技術的、要員の確保、院内幹部の認識、職員の理解と協力など問題が山積している。しかし、看護は院内のほとんどの部所とかかわりあいを持ち、当該部門の電算機導入と同時に看護における利用が可能となる。現時点の電算機利用は、この段階にとどまっている。さらに、看護婦間の、および患者と看護婦間の利用には効率、ソフトウェア開発等の種々の問題が存在し、なかなか実現しにくい。

本大学病院では、学長自らがコンピュータ管理センター長を兼務し、10年近くシステムの研究・開発を続けている。数回にわたる関係職員の海外研修が実施され、看護部として3回、延9名が参加している。この恵まれた環境のもとで、今後一層の努力、勉強を行ない、早期に良い独自の看護システムを作りたいと思っている。

MUMPS Poetry

○ Donald A. Smith and Ichiro Wakai, M.D.

MUMPS System Laboratory, NAGOYA

A simple MUMPS routine writes poetry at high speed.

The structure of a simple English sentence can be described by the following grammatical production rule:

SENTENCE ::= NOUN_PHRASE VERB_PHRASE NOUN_PHRASE.

A noun_phrase, in turn can be described as:

NOUN_PHRASE ::= ARTICLE ADJECTIVE NOUN; or

NOUN_PHRASE ::=

ARTICLE ADJECTIVE NOUN PREPOSITIONAL_PHRASE NOUN_PHRASE.

Notice, please that there are two choices available for expanding the syntactical unit NOUN_PHRASE and that NOUN_PHRASE is described recursively in terms of itself. Continuing in a similar vein,

VERB_PHRASE ::= VERB ADVERB.

(There are of course other possible production rules which would yield English-like sentences.)

The syntactic unit NOUN has many possible choices for the right part of the definition: for example, "keyboard", "kidney", "Measles", "altruism". If we store such nouns in a MUMPS global file (as in listing #2) and if we write a MUMPS routine embodying syntactic production rules (as in listing #1), then it's reasonable to expect English-like sentences to be generated. Indeed, if we choose our vocabulary carefully (for example, choose "flower" rather than "indigestion"), almost poetic sentences are produced.

Of course, almost all of the poems produced by the routine in listing #1 are of poor quality. Here are some examples that are typical:

A turgid outside blossoms blankly but a unreachable finger.
The indescribable skin keeps quite the shallow inside.
Her lonely over-indulgence sings shly but a tearful storm.

Yesterdays's dead hornet teases never tomorrow's flower's water.
 Yesterday's forgotten skin runs madly my fantastic hypocrisy.
 One small bones hopes for coyly a dark spring.
 But a beautiful skin after today's dispirited forgetfulness breaks slowly my forgotten disease.

My flower's hatred atop but a transparent heavy headedness quickly defies but a shallow hair.

Tomorrow's dull evening after yesterday's warm child by one understated passion destroys knowingly your vibrant beauty below today's only girl.

On the other hand, considering the fact that the routine in listing #1 is short and that it produces each poem in about two seconds (on microcomputer MUMPS), one cannot complain. No doubt, if one increased the size of the vocabulary in the global file, if one perhaps added more production rules to the grammar, and if one were willing to read through numerous bad poems, eventually one would come across a Nobel prize winning poem.

We conclude this brief comment with some of the better, or at least more amusing, untouched poetic productions.

His happy despair nightly rattles the unreachable water.
 The flat pancake surely sings the mind's lower lips.
 The mind's indescribable misfortune chases sadly above your dispirited passion.
 That morning's infatuation mourns dreaming yesterday's forgotten despair.
 The fleshy infatuation flows forgetfully after today's weird river.
 One morning's heavy-headedness flatters farther the past spirit above a vacant moon.

```

*****
Listing #1
POEM      ;DAS,;5/21/84; Writes a computer poem from globals.
;POEM:= NOUN_PHRASE VERB_PHRASE NOUN_PHRASE
S POEM=" D NP,VP D:$R(6)=5 PP D NP
I $E(POEM)71L S POEM=#C($A(POEM)-32) $E(POEM,2,254)
S POEM=#E(POEM,1,$L(POEM)-1) W POEM,! Q
GET       ;X HAS COUNT
S X=$R(X)+1,Y=$L(^X),...
S POEM=POEM,$P(^X),...,$R(Y)+1) . . . Q
NP        ;
D ARTICLE,ADJ,NOUN I $R(6)=3 D PP,NP
Q
VP        I $R(3)=2 D ADV,VERB Q
D VERB,ADV Q
VERB      S X="VERBS I ^VERBS(1)"= " D GET Q
ADJ       S X="ADJ I ^ADJ(1)"= " D GETN Q
NOUN      S X="NOUNS I ^NOUNS(1)"= " D GETN Q
ADV       S X="ADV I ^ADV(1)"= " D GET Q
ARTICLE   ;
S X="ARTICLE I ^ARTICLE(1)"= " D GET Q
PP        S X="PREPH I ^PREPH(1)"= " D GET Q
GETN      S Z=$R(X)+1,Y=$L(^Z),...
S Z=$P(^Z),...,$R(Y)+1) I POEMCZ G GETN
S POEM=POEM_Z_ . . .
Q
REP       K F Q=1:1 D POEM I Q#18=0 W !,"? " R X
    
```

```

Global name: ^NOUNS
^NOUNS      5
Listing #2  ^NOUNS(1)  frog^sun^moon^lake^pond^snow^rain^storm^wind^leaf^dew^mist^calm^
cloud^sound^sky^hand^eye^clap^sonnet^life^hope^tear^window^butterfly^hair^skin^h
eart^cheek
^NOUNS(2)  water^thunder^flower^ripple^mountain^hornet^apple^flavor^surpris
e^passion^disease^inside^outside^finger^firefly^sunflower^grasshopper^morning gl
ory^lost love^spring
^NOUNS(3)  mischief^misfortune^hatred^girl^boy^child^water^rose^bird^wonder
^flesh^spirit^hope^regret^flight^button^pancake^knife^oven^death
^NOUNS(4)  face^bones^lips^guts^veins^daughter^suffering^soul^fear^concern^
delight^infatuation^over-indulgence^hypocrisy^beauty^callousness
^NOUNS(5)  forgetfulness^heavy headedness^despair^river^stars^evening^midn
ight^lover^enemy^friend
    
```