

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

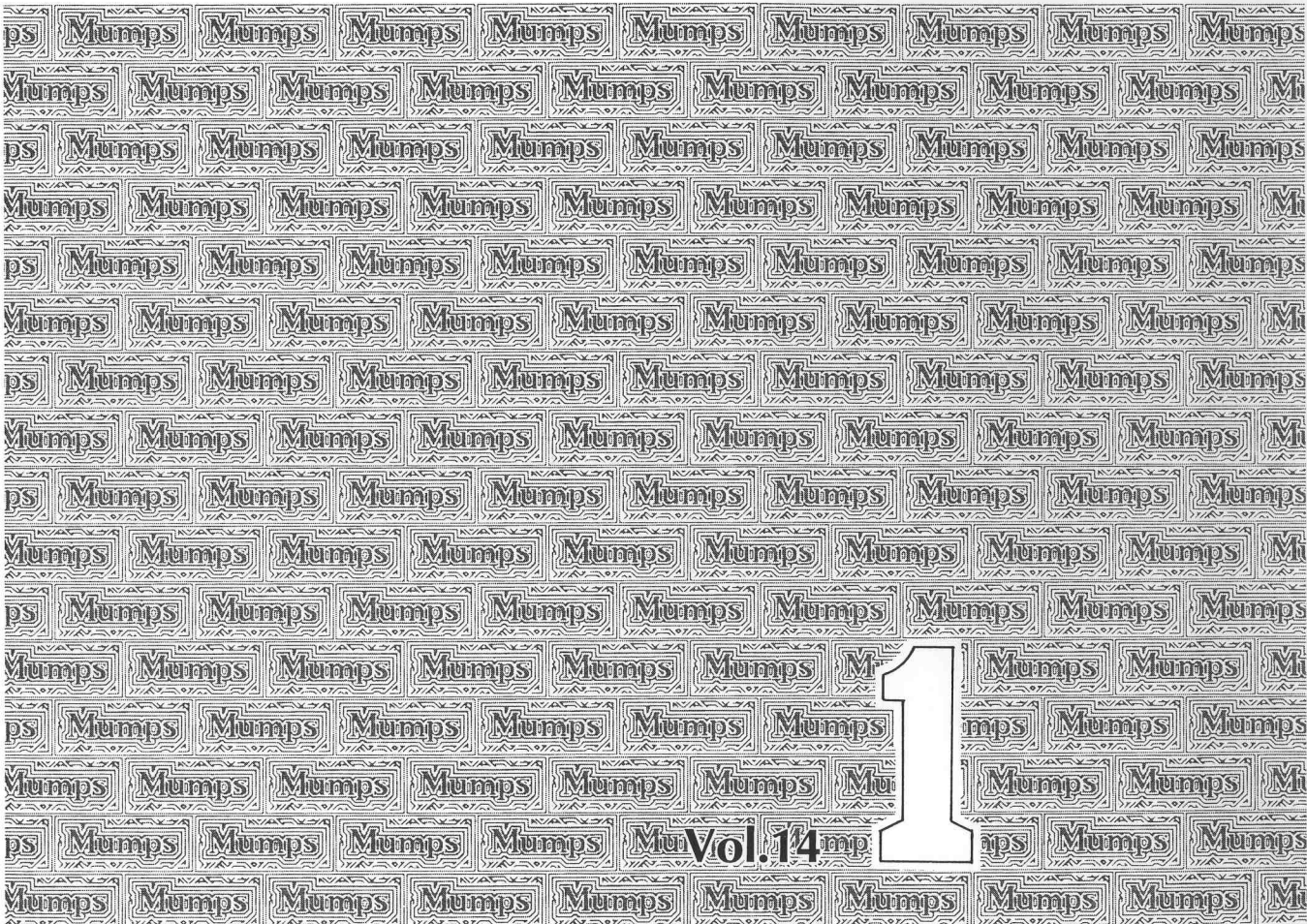
Tel: 0463-93-1121 ext. 2140

Fax: 0463-96-4301

Email: [youichi@keyaki.cc.u-tokai.ac.jp](mailto:youichi@keyaki.cc.u-tokai.ac.jp)

A Scientific Periodical on MUMPS

# Mumps



1

Vol. 14

Edited and Published by MUG-Japan





# *Mumps*

Vol. 14 – 1

*Edited and Published by* MUG-Japan



# 第14回日本MUMPS学会

於 産業医科大学ラマツィーニホール



昭和 62 年 7 月 29 日 ~ 7 月 31 日



# KIOKU—実験動物データ解析の エキスパートシステム

今泉 幸雄 藤本 浩司 小形 稔 金子 泰久 小池 敏  
アップジョン ファーマシューティカルズ リミテッド 総合研究所\*

Key words: Expert system, KIOKU, animal safety assessment test.

## はじめに

医薬品の動物安全性試験データのコンピュータ化（以下JURAS）はすでにデータの収集、解析に利用しているが、研究者が報告書にする段階では、文献情報、背景因子、標準データなどを考慮したまとめが必要である。これらの情報をデータベースとしたエキスパートシステム（Expert System, 以下ES）と JURASの結合、および対話形式による試験データの評価を目的に、専門分野の有志（コンピュータ部門、病理毒性部、薬理部、化学部）が1984年10月よりプロジェクトで、JURAS と結合するESの構築を開始した。

ESの対象は、JURAS の生データからの薬物の無影響量、中毒量などの算出、並びに症状データからの結果の推定であり、専門家の知識、経験はもとより薬物の特徴や実験動物の背景因子の情報も一元管理される。ESを構築するに当たって、調査・研究をした結果、独自の shell（ES構築ツールプログラム）開発を決定し、その shell仕様（以下 KIOKU）特に知識表現と推論方法を決めてFeasibility Model の作成をした。KIOKU を決める前後には、毒性専門家によるヒアリングを行い、shell 開発においては、proto type版を bottom-up形式にプログラミングと評価を繰り返す、MUMPS で開発中である。

## ESの概要

図1に示したように、試験の生データを shell部分でアクセス可能にするKERNELとスーパーバイザのほかに、主に時系列標準データベース部、辞書部、生データ変換部および対話処理部より構成される。

- ① 時系列標準データベース部＝動物種（ラット、マウスなど）、系統（SD系、Fisher系など）、性別、Age（日、週、月など）、検査項目（生化学、血液など）、計測機器（機器、単位、範囲など）に対して標準値を格納する。
- ② 辞書部＝対話処理部や知識表現などに用いる専門用語（同意語）とルールなどに関連する薬

---

Information for authors. Kouji Fujimoto, Yukio Imaizumi

\*〒370 群馬県高崎市大八木町168（電話：0273-63-2211）



物 情報や文献情報を格納する。

- ③ 生データ変換部=毒性試験計画、生データ、統計計算結果などを instance(知識表現) として変換し、アクセスを可能にする。
- ④ 対話処理部=利用者などへの問い合わせ文の生成と入力されたコマンドの解釈、実行をする。

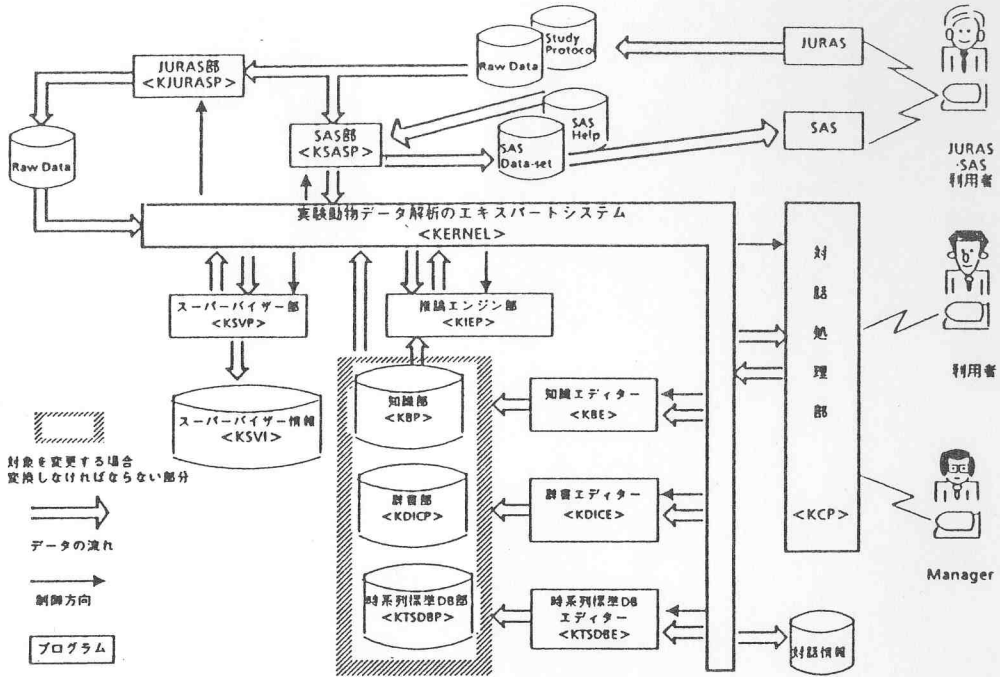


図1 実験動物データ解析のエキスパートシステム

### Shell(KIOKU)について

知識表現は Frameをもとに 2 方向性( 前向き、後ろ向き) を持つ Production Rule(以下、PR) を併用。推論部は各 Frame推論と Frameを構成する Slot推論に分けられる。

#### 知識表現

- ① Frame = Frame共通部と Slot部よりなる。前者は Frameの定義、Frame内共通情報 (Attached Procedure、上位・下位の定義)、Instance識別で、後者は Slot名、Slot値、評価値、標準値、Demon (4種類)、PRの起動条件と Rule番号で構成される。
- ② PR=Ruleの番号、前向き・後ろ向きの区別、Ruleの履歴、条件部、Action部 (前向き・後ろ向き・System用の Working Memory<以下WM>の操作命令)、後ろ向き専用の Promptや説明機能よりなる。

推論部

- ① 推論の順序 = Instance(FrameがWMにLoad) に生データがSlot値として登録され、該当Frameは起動Frameで動く。
- ② Frame推論 = まずPre-Attached Procedureが起動し、全Slot推論が終了するとPost-Attached procedureが起動する。
- ③ Slot推論 = Slot値の値に応じて該当する Demonが起動、またSlotに登録されているRuleが起動してSlot推論の一部処理を行う (RuleはAgendaに登録して照合を速くするために複合の場合がある)。
- ④ PR推論 = 事実データはすべてInstanceの中にSlot値として存在し、Slotより起動されたRuleが終了するまで繰り返し、結果をSlot推論に返す。

推論域

Frame とPRは別々である。Frame はInstanceが確保される域であり、PRは前向き用と後ろ向き用にAgendaがあり1 slot分終了すると、WMとAgendaは初期化される (図2)。

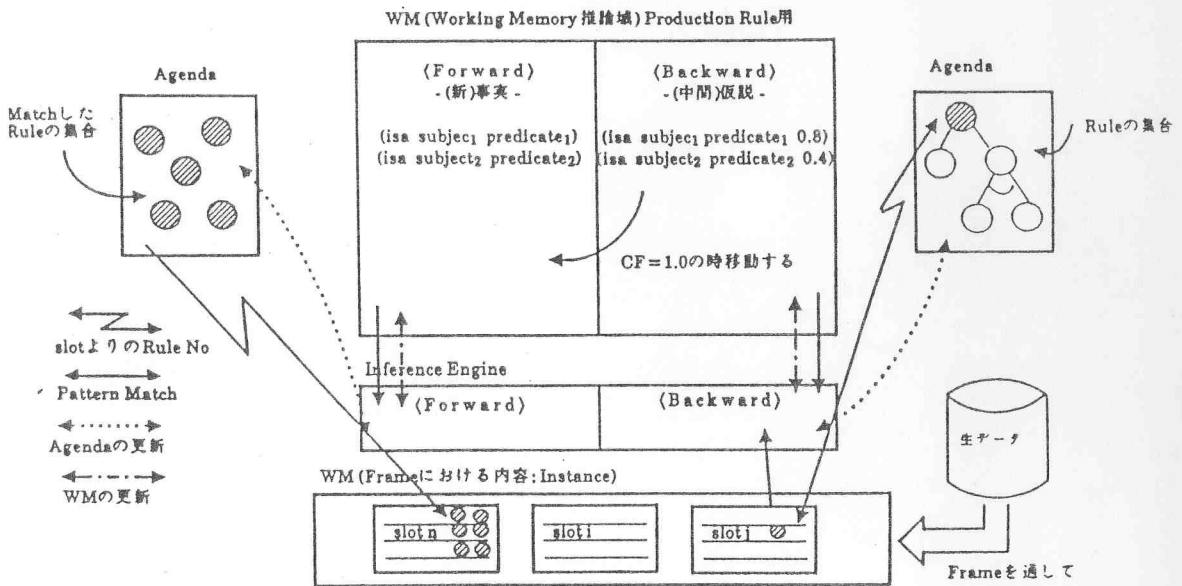


図2 PR推論

プロトタイプ版

KIOKU を実証するに bottom-up形式にてプロトタイプ版の改善を繰り返して、作成中である。KIOKU は Frame理論をbaseとし、PRを併用して、Frameには階層的であり教科書的な知識を、PRには経験的な知識を格納する。プロトタイプの初期版は FrameとSlotの知識表現を重視し、

さらに Frameと Slot(Demon)推論を加えて評価し、次にPR知識表現を加えて、さらにその推論を加えて評価している。

Frame(Slotを含む)とPRの知識表現には、各々Editorを作成してglobal fileとして格納し、推論途中のWM(PR用)、InstanceとAgenda(PR用)はlocal variableでsnap shotとして参照できるようにになっている。またPre-Attached Procedure、Post-Attached Procedure、各々のDemonは通常のプログラム名(MUMPSプログラム)にて登録しておき、DO制御にて起動するようにしている。

### 毒性専門家からのヒアリング

毒性専門家からのヒアリングはKIOKU仕様を決める前とプロトタイプ版の作成途中に助言をいただいている。

### おわりに

今回は、毒性試験を対象にしたエキスパートシステムの概要とshell(=KIOKU)、プロトタイプ版について報告し、次回以降はプロトタイプの評価とreal-use版と毒性エキスパートの知識獲得について言及したい。

### 文 献

- 1) 白井良明, 述井潤一 「人工知能」 岩波書店
- 2) Koyama T. "An Automatic Rule Generation and Evolution Schema for AI Medical consultation systems" 3rd Medical Informatics, pp 1306 - 1310 (1980)
- 3) M.Minsky : "A Framework for Representing Knowledge, in the Psychology of Computer Vision" McGraw-Hill (1975)
- 4) 「アップジョン新薬の毒性試験支援システム開発中」日経AI 1987, 5月11日号 Page 6
- 5) 今泉幸雄, 国原峯男, 小形稔, 金子泰久 "実験動物データ解析の支援システム(1)" 情報処理学会 第31回(昭和60年後期)全国大会論文集 pp 929 - 930 (1985)
- 6) 「KIOKU-実験動物データ解析のエキスパートシステム」 INTER技術論文 UPU (1987)
- 7) 今泉幸雄, 藤本浩司, 国原峯男, 小形稔, 金子泰久, 小池敏 "実験動物データ解析の支援システム(2)" 情報処理学会 第34回(昭和62年前期)全国大会論文集 pp 1429 -1430 (1987)
- 8) 和田英一 「算体主導型プログラミング」数学セミナー PP 31 - 36 (1983)
- 9) Richard O.Duda and Edward H.Shortliffe "Expert Systems Research" SCIENCE ,15 April 1983 ,Volume 220 , Number 4594
- 10) Nii,H.P. and Aiello,N. "AGE (Attempt to Generalize) : A Knowledge-Based Programs for Building Knowledge-Based Programs" Proceedings of 6th International Joint

- Conference on Artificial Intelligence , pp 645 - 655 (1979)
- 11) F.Hayes-Roth , D.A.Waterman and D.B.Lenat "Building Expert Systems"  
Addison-Wesley Publing Company,inc (1983)
  - 12) Richard Fikes and Tom Kehler "The role of Frame-based Representation in  
reasoning" Communications of the ACM , September 1985 , Volume 28 , Number 9  
pp 904 - 920 (1985)
  - 13) F. Hayes-Roth "Ruled-Based Systems" Communications of the ACM , September 1985  
Volume 28 , Number 9 , pp 921 - 932 (1985)
  - 14) 高垣善男監修「長期動物実験：第6章 長期動物実験における成績評価」清至書院  
pp 262 - 293 (1981)
  - 15) 中村正夫ら監修「ナースに必要な臨床検査マニュアル：第3章 主訴・症状と臨床検査」  
学習研究社 pp 211 - 270 (1984)

## KIOKU — Expert system of experimental animal data analysis

\*Yukio Imaizumi, Kouji Fujimoto, Minoru Ogata, Yasuhisa Kaneko, Satoshi Koike  
Upjohn Pharmaceuticals Limited, Research Laboratories\*

keywords: expert system, frame, production rule, KIOKU,  
JURAS, safty assessment test

We are using computerization of animal safty assessment test of drugs (JURAS) for data gathering and analysis. In researchers' report, they need a summary considering reference information, background factor, standard data, etc.. In order to join expert system (ES) based on the data base of above information and JURAS, and in order to estimate test data by conversational method, we, interested in this field (belonging to computer, path & tox, pharmacology and chemistry unit), started the project for making ES joined with JURAS in October 1984.

Objects of ES are to estimate non-toxic dose and toxic dose from raw data of JURAS, to infer results from clinical data and to control unitarily information of drug characteristics and background factor of experimental animals. Based on research and study for building ES, we determined the development of original SHELL (ES building tool program) and made Feasibility Model by determing SHELL specification (KIOKU), particulary knowledge representation and inference method.

ES consists of KERNEL that makes possible for raw data to be accesed, Supervisor and four subsystems, that is, Time series standard data base, Dictionary, Raw data translation and Interactive. In Knowledge rerpresentaion, it constructs the Frame theory and Production system. In Inference, Frame theory has frame inference and slot inference, Production system (PR) has forward chaining

In determing KIOKU, we made hearing from toxicology experts. We are developping SHELL with MUMPS by repeating programming and estimation of proto type with bottom-up type.

---

Information for authours. Yukio Imaizumi  
\*168 Oyagi-machi Takasaki-city Gunma-Prefecture Japan 370

# U-MUMPSの機能強化(II)

小林 勝 煙山 孝 上戸 隆

住友電気工業株式会社システム事業部 MEシステム課\*

Key words: U-MUMPS,

## はじめに

筆者らは、1984年10月住友電工製ワークステーションUSTATIONで稼動するマルチユーザMUMPS (U-MUMPS)を開発し、その後も日本語処理機能の拡充、DDP機能の開発、稼動機種拡大等の機能強化を行ない、本学会では報告してきた。1) 2)

今回、新たにこのU-MUMPSに対して、下記の機種強化を行なったのでここに報告する。

- 1) 日本語標準MUMPSイ草案準拠：旧仕様との互換性を保った変更
- 2) 各種通信インターフェースドライバのサポート：インテリジェントCCU, DRV68, SUMINET3200 (住友電工製光LAN)
- 3) 他言語インターフェース機能の開発：\$ZCALL関数の開発
- 4) LANサポート：CPU-CPU間における分散データベース機能の開発
- 5) 対象機種の拡大：U-STATION E30におけるU-MUMPSの稼動

## 日本語標準MUMPSイ草案準拠

U-MUMPSは1984年10月開発以来、日本語処理のサポートを行なってきたがバージョン1.7APより、日本語標準MUMPSイ草案に準拠した日本語処理仕様となるように下記の仕様変更を行なった。

### 1) 関数

- ・ \$ASCII, \$CHAR  
漢字に対する対象コードをシフトJISコードからJISコードに変更した。
- ・ \$ZWIDTH  
第2引数(漢字ピッチ)の指定を漢字ピッチの2倍の値から1倍の値に変更した。
- ・ \$ZPOSITION  
新規追加を行なった。

---

Information for authours. Masaru Kobayashi, Takashi Kemuriyama, Takashi Kamido.

\*〒530 大阪市北区堂島 1-2-5(ホルカビル) (電話: 06-347-7511)

・ \$ZCRC

漢字に対するコードをシフトJISコードの他にJISコードのCRC（総和，排他的論理和）を追加した。

2) 入出力

・ R \*X W \*X

漢字に対する対象コードをシフトJISコードからJISコードに変更した。

・ OPEN/USEコマンドの13番目のパラメータ（漢字ピッチ）

漢字ピッチの2倍から1倍の値に変更した。

3) 漢字パターン照合

・ 従来のTからTに加えて全を追加した。

旧仕様との互換性をとるため、標準設定は新仕様であるが、パーティション単位に新仕様で稼動するか、旧仕様で稼動するかのフラグを設け、新旧いずれの仕様でも稼動が可能である仕様とした。このため、新旧仕様選択ユーティリティ%YKMODを追加したほか、システムジェネレーションユーティリティにUCI単位の新旧仕様選択機能を追加した。

さらに、下記の旧仕様関数を右に示す関数名で新仕様へ移行した。

・ \$ASCII ( ) 関数 → \$ZZA ( ) 関数

・ \$CHAR ( ) 関数 → \$ZZC ( ) 関数

・ \$ZWIDTH ( ) 関数 → \$ZZW ( ) 関数

### 各種通信インターフェースドライバのサポート

同一機種内および異機種間通信の拡大を行なうために、USTATIONの通信ドライバに対して、MUMPS標準I/Oコマンド（OPEN，USE，READ，WRITE，CLOSE）を利用可能とした。表1にU-MUMPSで利用可能な通信ドライバの一覧を示す。

このうち、インテリジェントCCUはハードウェアにプロトコルモジュールをダウンロードする形式で各種の有手順に対応可能なように設計されている。現在、プロトコルはDT1211手順，BSC手順がサポート可能である。

また、高速パラレル転送可能なDRV68（DR11W準拠），5項で述べる住友電工製光LAN SUMINET3200が利用可能である。

表1 U-MUMPSで利用可能な通信ドライバ

4チャンネルCCU (RS232C, RS422)
8チャンネルCCU (RS232C)
インテリジェントCCU (BSC手順, DT1211手順)
DRV68 (高速パラレル通信, DR11W準拠)
SUMINET3200 (住友電工製光LAN)

## 他言語インターフェース

MUMPSの世界から、UNIXのコマンドおよびC, FORTRAN, アセンブラ等で作成したロードモジュールを実行可能とする\$ZCALL関数を開発した。シンタックスは、下記の通りである。

```
S X=$ZCALL (MODULE, PRM1, PRM2, . . .)
```

但し、

MODULE : 実行ロードモジュール名

PRM1, PRM2 . . . : ロードモジュールに対する引数

X : 実行不可能な場合は-1, その他は0を返す。

図1にUNIXのディレクトリ表示コマンドlsを\$ZCALL関数を用いて、実行した例を示す。

## LAN(ローカルエリアネットワーク)

U-MUMPSにSUMINET3200ドライバを組み込み、OPENコマンド時に相手先ノードアドレス、ポート番号を指定することにより、バーチャルサーキットモード、データグラムモードにおけるローカルエリアネットワーク通信が可能である。

分散データベース処理機能 (Distributed Data base Processing, 以下DDPと略す) はEthernetを用いたDDPを開発した。SETコマンドに加えて、KILL, WRITE, LOCKコマンドや\$ORDER, \$DATA関数にも、DDPシンタックスは可能である。

UstationのLAN機能であるTELNET (リモートログイン機能), FTP (ファイル転送機能) と相まって、システムの利用度の拡大が期待できる。

```

MUMPSの世界
.
.
.
>S X=$ZCALL (" /bin/ls", "-l")
.
.
.
                                プログラム名   引数並び

```

UNIXの世界					
9254032	Jul	8	11:58	database.mcm	
13970	Jul	30	1986	mcmf	
15610	Jul	30	1986	mctr1f	
16205	Jul	30	1986	mrdndf	
21467	Jul	30	1986	mdjncf	
22458	Jul	30	1986	mlc1cf	
28042	Jun	29	10:40	mln1ff	
998	Jul	1	15:57	mumsc.log	
197113	Jun	29	10:37	mumsmf	
10873	Jul	30	1986	mumscf	
22314	Jul	30	1986	mut1cf	

- UNIXコマンド  
(ディレクトリ表示, スクリーンエディタ等)
- 他言語作成プログラム  
(C, FORTRAN, アセンブラ等)

図1 \$ZCALL関数使用例



UNIX SYSTEM III版で稼動するUstation E10用U-MUMPS, UNIX SYSTEM V版 (release 1.0)で稼動するUstation E5, E15, E20mini用U-MUMPSに付け加えて、UNIX SYSTEM V (release 2.0)版で稼動するUstation E30用U-MUMPSを開発し、機種拡大を行なった。

表2にUstation E30仕様及び図2に外観を示す。

当社におけるMUMPSベンチマークテスト結果では、従来機種のE20の処理速度を1とした場合1.1とほぼ同程度であるが、ハードウェア技術によりコンパクトなサイズとなり、ディスクサイズも大容量から小容量まで選択可能でコストパフォーマンスに優れたシステムを提供できる。

### むすび

U-MUMPSに、他言語インターフェース、通信ドライバ、ネットワーク機能を負荷することで、MUMPSの閉じた世界から他の世界へ広がることが可能となったといえる。今後もU-MUMPSに対して、このような開放型の考えをもつ機能拡張を行い、より開かれた使い易いMUMPSを目指して開発を行なっていく予定である。

表2 Ustation E30仕様

CPU (フロック)	MC 68020 (16.6MHz)
メモリ	4MB(標準)~32MB(最大)
キャッシュメモリ	16KB
FD	1MB
I/Oポート	RS232C×2(標準) セントロニクス×1(標準) (最大 RS232×32)
ディスク	690MB/340MB/170MB (最大3.4GB,5ドライブ搭載時)
OS	Uniplus+ (UNIX SYSTEM V リリース2.0)

図2 Ustation E30外観



### 文献

- 1) 小林 勝 久江 正 上戸 隆: USTATION上のMUMPSの開発, P.11~P.14  
第11回 日本MUG学術大会 1984年12月
- 2) 小林 勝 久江 正 上戸 隆 阪倉 明: U-MUMPSの機能強化, P.53~P56  
第12回 日本MUG学術大会 1985年8月
- 3) 大櫛陽一ら: 日本語MUMPSの標準化について, P.207~P.215  
医療情報学 Vol.6, NO.2, June, 1986

Ethernetは XEROX社、DR11Wは Digital Equipment Corporation社、  
Uniplus+は、Unisoft社の登録商標です。UNIXは、ベル研究所で  
開発され、ATTによってライセンスされたオペレーティングシステムです。

## Enhancement of U - MUMPS (II)

Masaru Kobayashi, Takashi Kemuriyama, Takashi Kamido  
Medical Engineering System Section, System and Electronics Division,  
Sumitomo Electric Industries, Ltd.\*

We reported the developments of MUMPS (trade-mark:U-MUMPS)running under UNIX based workstation(trade-mark:Ustation E10),at the 11th Mumps User's Group of Japan,and the enhancement (Implementation of U-MUMPS,for Ustation E5,E15,E20, Development of DDP, and Enhancement of Kanji character handling)at the 12th Mumps User's Group of Japan.

In this paper we described the new enhancement of U-MUMPS Some of the main enhancements of U-MUMPS include :

- 1) Conforms to Japanese standard MUMPS type A
- 2) Enhancement of the U-MUMPS communication driver:  
Intelligent CCU, DRV68, SUMINET3200
- 3) Developments of the function call to the load modules without U-MUMPS language. For example : UNIX commands and the user written programs.
- 4) Development of the Distributed Data Processing(DDP) Function among USTATIONS' using Local Area Networks.
- 5) Implementation of U-MUMPS for Ustation E30 under UNIX SYSTEM V,release 2.0.

---

Information for authours. Masaru Kobayashi,Takashi Kemuriyama,  
Takashi Kamido.

\* Volkart Bld. Dojima 1-2-5 Kita-ku Osaka, 530 Japan

# MUMPSによる日本語処理と それをめぐる諸問題について

嶋 芳成

岐阜県厚生連久美愛病院\*

Key words: Japanese standard

## は じ め に

日本語MUMPS標準化委員会の努力により、MUMPSによる日本語処理機能の仕様案がまとめられた<sup>1)</sup>。そして、この仕様案に沿ったMUMPSシステムがすでにいくつか発表されるに至っている<sup>2), 3), 4)</sup>。これによって、日本語処理を含んだアプリケーションの移送性が高まり、ユーザーとして歓迎すべき状況になりつつある。

しかし、今までにまとめられた日本語処理機能は、いわば最小限必要とされる機能でしかなく、残念ながら十分に満足のできるものではない。

MUMPSは従来強力な文字列操作機能を特徴とするとされてきた。しかし最近、高機能のワープロが普及し、その影響を受けて各種のデータベースシステムの日本語処理機能も高度化している。これらと比較したときMUMPSの日本語処理機能は決して高いとは言えない。

## 漢字の大きさの問題

日本語を含むデータを処理するとき、まずぶつかるのが漢字がANK文字より大きいという問題である。漢字はCRT端末などではANK文字の2倍、プリンタでは1.5倍あるいは1.25倍の幅を持つことが多い。このため文字数だけを考えたMUMPSプログラムを用いると、出力されるデータの位置がずれるという現象が生じる。

この問題について、日本語MUMPSの標準仕様では、

- ・文字列操作の基本単位は文字とする
- ・表示幅を操作するために \$ZWIDTHと\$ZPOSITIONという関数を用意する

という方法で解決を試みている。

しかしデータの表示位置の制御のためにはまだ不十分である。

---

Information for authours. Shima Yoshinari.

\*〒506 岐阜県高山市大新町5丁目68番地 (電話: 0577-32-1115)

MUMPSの書式制御 WRITE ?n や \$JUSTIFY関数は、ANK文字の文字数を基準に書式制御する機能である。このため、漢字がANK文字の整数倍でないプリンタで、右揃えで出力するにはこれらを利用できない。例えば漢字がANK文字の1.5倍なら「日本語MUMPS」の幅はANK文字の9.5倍になる。これを左端から20文字の欄に右揃えで出力するには、左端から10.5文字分の空白を空けなくてはならない。ANK文字の空白を基準に書式制御すれば当然0.5文字分ずれを生じる。プログラムで意識して全角の空白を用いてこのずれをなくすることができるかもしれないが、望ましい方法ではない。漢字がANK文字の1.25倍の場合にはもっと複雑になる。

さらに、文字1つずつの大きさを調べると、漢字は決してANK文字の1.5倍や1.25倍ではないことが多い。文字と文字の間に何ドット分かの空白があり、それを合せて全体で1.5倍または1.25倍に調整されている。この文字間の空白が各文字の右側にある装置では、ANK文字と漢字の混在したデータを右揃えしても、右端が視覚的には揃わないことが多い。特に、CRT端末と同じように漢字をANK文字の2倍として扱うプリンタでは文字間の空白が広くなり、右端のずれが無視できないものになる。

結局、ANK文字0.5文字分あるいは0.25文字分の幅の空白を出力する機能が必要である。できれば任意の幅の空白を出力する機能が望ましい。任意の空白を出力する機能があれば、ANK文字も漢字も幅が一定である限り、自由に書式制御可能となる。

### 文字属性の管理

次に、文字の大きさが1文字毎に異なる場合について考える。

今までの議論では、ANK文字の大きさは一定であり、漢字もANK文字より大きいけれども漢字であればすべて同じ大きさであることを仮定してきた。

しかし、アルファベットの場合、プロポーショナル印字という問題がある。プロポーショナル印字では1文字毎に文字幅が違うため、漢字の問題より複雑である。その上、表示文字には複数のフォントがあり、同じフォントでも大きさの異なる文字があり、それらが、斜体、ボールド体、影付き体などのスタイルを持っているとさらに深刻な事態となる。

漢字についても近い将来、複数フォント、複数フォントサイズを用いることのできるシステムが普及すると予想される。そうなれば、アルファベットのプロポーショナル印字と同じ問題が生じる。

このような様々な大きさの字が混在したデータを、左・右・中央に揃えたり、均等割付するには、各文字の幅、高さ、隣の文字との間の空白の幅などの文字属性を、全データの1文字毎に管理しなくてはならない。

MUMPSでこのような文字属性を管理するシステムを実現する方法のひとつを次に述べる。

あらゆる文字列について、各文字の属性を表わす属性データを付随させるシステムを作ることができる。

例えば、変数Aがあったとき、その裏には A{font}, A{size}, A{style} という3つの属性変数が付随する。これらの属性変数の文字数は変数Aの文字数と同じとする。そして属性

変数の各文字は、それに対応する変数Aの各文字の属性を表わす値にセットされる。

これら属性変数は、従来の文字列処理には表面に出てこない。

例えば、A="MUMPS" のとき、

```
A{font} = "GGHHH"
A{size} = "42222"
A{style}= "BIIII"
```

のようにセットされ得る。ここで "G", "H" は対応する "MUMPS" という文字のフォントを表わすシンボルで、"4", "2" は対応する文字の大きさ、"B", "I" は文字がボールド体またはイタリック体であることを表わすシンボルとする。すなわち先頭の "M" は、"G" フォントの "4" のサイズでボールド体であることになる。

ここで、 SET B=A とすれば、同時に

```
SET B{font}=A{font}
SET B{size}=A{size}
SET B{style}=A{style}
```

という処理が暗黙の内に実行される。

もし、 SET B=\$EXTRACT(A,1,3) を実行すれば、同時に

```
SET B{font}=$EXTRACT(A{font},1,3)
SET B{size}=$EXTRACT(A{size},1,3)
SET B{style}=$EXTRACT(A{style},1,3)
```

が実行され、その結果

```
B{font} = "GGH"
B{size} = "422"
B{style}= "BII"
```

となることは容易に理解できる。

さらに、もし最初の文字をフォント "N" の、サイズ "6" にしたければ、

```
SET $EXTRACT(A{font},1)="N"
SET $EXTRACT(A{size},1)="6"
```

とすることができよう。(注意: SET \$EXTRACT は現在の標準MUMPSにはない仕様)

そして例えば変数Aの頭文字の表示幅(WIDTH)を知りたいければ、例えば次のように計算できる。

```
SET CHAR=$EXTRACT(A,1)
SET FONT=$EXTRACT(A{font},1)
```

```
SET SIZE=$EXTRACT(A{size},1)
SET STYL=$EXTRACT(A{style},1)
SET WIDTH=`fonttable(FONT,CHAR,"width")*SIZE*`styleparam(STYL,"width")
```

ここで、`fonttable`にはあるフォントのある文字の大きさについての情報が入り、`styleparam`には、文字の飾り（ボールドや影付きなど）によって、文字の大きさが変化する係数が入っているものとする。

もし文字属性について何の指定もないときにはシステムの既定値がセットされることにすれば、現在の標準MUMPSに対して、文法的に上位互換がとれるというメリットがある。

またこの方法を拡張すれば、例えば `A{color}` に文字の色を、`A{angle}` に文字の傾きなどをセットすることもできる。

こうして管理された、様々な属性を持つ文字データを、プリンタに書式制御しながら出力することは容易ではない。しかし、最近 PostScript などのページ記述言語が普及し始めている<sup>5)</sup>。これらのシステムと連携すれば、MUMPSが直接プリンタや端末のハードウェアを操作する必要がなくなり、装置依存性が極めて低くなる。多数のフォントについても、業界標準的なフォントを採用し、そうでないものについては装置へのダウンロード機能を用意することは困難ではないと考える。

### 全角文字のパターン照合

日本語データを処理する上で、文字の大きさ以外にも、いくつかの、多分日本語に特有な問題がある。

現在の日本語MUMPS標準仕様では、全角文字すべてに対して「全」というパターンコードしか与えられていない。従って、入力された文字を漢字、全角アルファベット、記号、罫線素片などに分類するには、1文字ずつ比較する必要がある。全角文字の中をいくつかの文字集合に分けて、別々のパターンコードを割付ければ良いことであるが、文字集合の分け方に議論があり統一できていない。中でも特に問題になるのは「記号」の扱いである。

1. 長音記号「ー」は記号とも言えるが、片仮名の言葉の一部として用いられるのでこれは片仮名として扱うべきだと考えられる。
2. 「、」「ゝ」「ゞ」「々」の5つの記号は、直前の文字の繰り返しを表わすものである。これらも記号として扱うと困る場面がある。例えば、名前の「弄々枝」は3つの漢字、「すゞ糸」は3つの平仮名とすべきである。さもなければ、パターン照合演算子だけでは人名さえチェックできないことになるからである。  
漢数字のゼロ「〇」も記号ではなく漢字として扱うべきである。そうすれば「西暦二〇〇〇年」という表現を漢字として認めることができる。

パターンの分類方法を定めるには、実用上の便利さを最優先すべきである。しかしそのためにかえって、立場や好みの違いが表面化し、標準化し難いのが現状である。

別の解決方法としては、アプリケーション毎にパターンコードを定義する機能を持たせる方法も考えられる。例えば、パターン定義のために PATTERN という命令を新設し、

```
PATTERN 名=ひ:カ:漢:"、ゞ、ゞター"
```

と宣言して、

```
READ NAME IF NAME'?'1.名1" "1.名 WRITE "???"
```

のように利用する方法である。この方法であれば、上記の“記号”を、記号とするか、カナや漢字として扱うかという議論を避けることができる。さらに、目的に応じたパターンコードをつくるので有用である。

### 仮名の表記法と並替えの問題

日本語のデータを、言葉の読みで検索する場合、仮名の表記方法が統一されていない、あるいはしばしば間違えられるという問題がある。

例えば「大山」という名前の読みは、「おおやま」と書かれることが多いが、ときに「おうやま」とされたり「おーやま」とされることもある。一旦別の表記法で登録されたデータは、2度と検索できなくなる可能性があり、アプリケーション開発上、難しい問題である。

ここでこれらの表記方法を変換する機能をMUMPSに組み込むことができれば、大きなメリットがあるのではないだろうか。例えば次のような関数である。

- 1) 「お」の段の文字の直後の「お」を「う」に変換する。  
例: \$ZKCONVERT("おおやま","う") --> "おうやま"
- 2) 「お」の段の文字の直後の「う」を「お」に変換する。  
例: \$ZKCONVERT("こうり","お") --> "こおり"
- 3) 「ー」記号を直前の文字の母音に変換する。  
例: \$ZKCONVERT("きーぼーど","母") --> "きいぼおど"

入力データをこのような関数で変換して、検索キーをつくることにより、データ検索の検索漏れを防ぐことができるようになる。

検索の問題と関連して、50音順のデータの並替えの問題がある。現在のJISコードの順に単純にデータを並べても、50音順には並ばない。例えば「がいこく」は、「かんぞう」の後に来る。これは一般の辞書配列とは逆である。コンピュータの中の仕組みに疎いエンドユーザにとってはこれは決して当り前のことではない。すでに一部のパソコンシステムでは、50音順に並べる機能を持ったデータベースシステムが発表され、それがそのシステムの大きな特徴として評価を受けている<sup>6)</sup>。さらにそのシステムでは、検索キーが全角の片仮名か平仮名か、8ビットのカタカナかを問わず、その上長音「ー」記号もその発音で並替えるという機能を持っている。



MUMPSで50音配列を実現するには、まず8ビットの仮名から全角の仮名に変換する。次に、濁音半濁音の仮名を清音の仮名に変換する。さらに上述の\$ZKCONVERT関数などで仮名の表記法を統一する。こうして変換されたデータと、元のデータの両方をキーにして検索グローバルをつくれれば良い。

日本語処理が当たり前のように普及すればするほどこのような、きめ細かい機能が要求されるようになるものと予想される。

こういった機能は、ユーティリティで処理すべきであるという考え方もできる。しかし、日本語処理にとって一般的で、避けられない問題であるならば、言語仕様内に含めてもよいのではないかと考える。

### バイト処理機能について

上記のようなデータの変換に加え、データをバイト単位に分解して操作する関数（特別仕様の\$EXTRACTや\$PIECE）が必要であるという議論がある。これがあれば、JISコードやいわゆるシフトJISコードなどの異種コード間のコード変換や、画面制御、バイナリデータの処理などに有用だということである。

機能として、異種コードの文字の読み込みやバイナリ処理が必要であることは確かである。しかしこれはバイト単位に分解して行なう必要は必ずしもない。シフトJISコードを採用しているシステムでも、OPEN命令やUSE命令のパラメータを工夫することで、データを1バイトずつ取り込み、それを文字として扱うことは可能である。バイナリデータも同様である。

そのようなシステムでは特別な関数は不要で、従来の\$EXTRACT、\$PIECE関数などで処理できる。

画面制御のために、VRAMメモリーを直接アクセスする場合でも、多分それにはVIEW命令や\$VIEW関数を用いるだろうが、これらの命令・関数の引数を工夫することで文字単位の処理で十分対応できる。

データをバイト単位で処理する場合には、データの構造（1文字のバイト数、コード体系、シフトイン/シフトアウトの文字）などを意識しなくてはならない。一方OPEN、USE、VIEW、\$VIEWで外界とのリンクを制御したシステムでは、そのような意識は不要となる。プログラムはより論理的で装備独立性が高くなる。

このような理由から、バイト単位の処理機能をMUMPSの文法的なレベルに持ち込むことには反対である。

### カナ漢字変換システム

最近ではコンピュータ全体の日本語処理機能が上がり、OSのレベルで仮名漢字変換システムをもつものも珍しくはない。OSレベルで仮名漢字変換するシステムは、アプリケーション側の負担を軽くし、そのコンピュータでの日本語処理に統一的なインターフェースを与えるという利点がある。

しかしデータ入力の際に、ひとつ問題点がある。すなわち、名前などの並替えのキーとなるデータを入力するとき、同じようなカナ文字列を2回以上入力しなくてはならないことである。まず漢字変換のために1回読みを入力し、その言葉の読みを登録するには再度同じ文字列を入力しなくてはならない。

一方、アプリケーションで仮名漢字変換を行なう場合には、仮名漢字変換のために入力した仮名文字列を保存しておける。そしてこのような入力の重複を避けることができる。

OSレベルの仮名漢字変換システムでも、変換前の仮名データを仮名漢字変換システムから受け取ることができ、すでに入力されている仮名データをもとに仮名漢字変換できるようなシステムも有り得る。データ入力の手間を省くことができるので、特に大量のデータを入力するエンドユーザにとっては是非必要な機能である。これについてもパソコンのデータベースシステムにはすでに実現しているものがある<sup>9)</sup>。

MUMPSの文法の中に、仮名漢字変換システムとのインターフェースを標準的に用意することはできないだろうか。

#### 終わりに

以上、大きさの違う文字の取扱の問題と、日本語データを扱う上で問題になり、MUMPS言語やシステムに絡んだ問題に付いて言及した。難しいのは、MUMPSシステムを不用意に大きくせずに、便利にすることである。また、将来的な仕様拡張に耐えられるものでなくてはならない。

ここに述べたのはまだ十分に検討が加えられていない意見ではあるが、是非諸氏のご批判を頂きたい。

#### 文 献

- 1) 大槲 陽一, 他: 日本語マンブスの標準化について, 医療情報学, 6(2):207-215(1986)
- 2) 煙山 孝: SP-MUMPSの機能拡張(その2), 第14回日本MUMPS学会プログラム・抄録集, 26(1987)
- 3) 小林 勝, 他: U-MUMPSの機能拡張(II), 第14回日本MUMPS学会プログラム・抄録集, 27(1987)
- 4) 佐藤 真美: 日本語標準MUMPSイ草案の採用に向けて, 第14回日本MUMPS学会プログラム・抄録集, 28(1987)
- 5) 下山 智明: Sunワークステーションでの図形・文書処理, 第14回日本MUMPS学会プログラム・抄録集, 134-139(1987)
- 6) リレーショナル・データベース・ソフト「日本語データベースシステム『桐』」, 日経パソコン1986年10月20日号, 291-299



# 千葉大学病院における標準化について

本多 正幸 里村 洋一

千葉大学医学部附属病院 医療情報部\*

Key words: System Standardization

## はじめに

多くの大学病院にあつては、ドクターなどの職員の数が多く、移動も頻繁に行われる。このような病院において情報システムを効率良く運用するためには、いろいろな側面で標準化が必要になる。標準化というとコンピュータ言語の仕様について言及されることが多いが、ここでは言語仕様を除く、例えばファイル設計、プログラム設計、教育面、運用面など諸々の標準化について、本院における実情を述べることを目的とする。本稿では、次節で一般的な意味での標準化の際の問題点について触れ、3節で本院における標準化やルールなどについて項目ごとに具体例をあげる。最後に今後の期待を述べてまとめとする。

## 標準化について

標準化を議論するときは、(1) 標準化の対象となる範囲および規模、(2) 標準化と特殊処理の関係、(3) 機密保護の問題、の3つの点が特に重要になると思われる。

(1) については、いくつかの病院または施設にわたる範囲を想定したものか、またはひとつの病院または施設内での標準化をねらったものかで、その難易度や標準化対象項目が違って来るであろう。しかし一病院内の標準化の場合でも、複数の病院にまたがる場合でも大切なことは、‘全体’と‘部分’の関係であり、‘全体’からみた‘部分’、‘部分’からみた‘全体’の両方向から標準化を議論することである。各部分においてそれぞれ標準化をねらって、各種試みがなされていると思われるが、それぞれの経験を相互に生かして標準化を拡張し、範囲を広げる努力をしなければならないであろう。

マンブスは比較的簡単に習得でき、プログラミングも容易な言語であるといわれる。またプログラムの修正が容易にできるため、仕様書の書きにくい(書いてもす

---

Information for authours. Masayuki Honda.

\*〒280 千葉市亥鼻 1-8-1 (電話: 0472-22-7171(内線:3291))

ぐに内容が変わってしまう) 言語であるという側面を持っている。それだけに標準化やルール作りが重要な言語の一つであろう。

(2) については、例えば一病院内の標準化を想定したときに話を限定すると、各科、各部門に共通な部分については標準化することができる。その結果互いの部門間の情報交換や、病院全体の各種統計を明確にすることができる。しかし各科、各部門に固有な処理が少なからず存在する。これらは標準化とは逆方向のものであり、標準化の立場から見れば特殊処理と考えられる。標準化が進み過ぎると、各部門間の個性が失われてくる傾向になり、逆に特殊処理や特例処理が増えすぎると、標準化の意味や価値が希薄になる。したがってバランスを考慮しつつ標準化を進める努力が大切である。

(3) の機密保護については、標準化が進むにつれてデータの利用範囲が広くなり、機密保護対策の重要性も増してくるであろう。

以下4つの項目(グローバル設計について、プログラム設計およびコーディング、運用、教育について、退院サマリーについて)については本院のシステムの現状を述べる。

### グローバル設計について

ここではデータベースの標準化が大きな課題になるであろう。データベースの構造は、やはり検索効率の良い構造が望ましい。本院では、データベースの中心的存在である患者データベースに対し、インバーテッドファイルおよびデータ辞書を作成し検索効率の向上を図っている。コンピュータ上に蓄積されているデータとしては、退院要約、患者基本情報、患者入院記録、検査結果、入退院履歴、入院カルテ管理、検索者参照記録、肺機能検査、血液型情報、輸血記録、研究者別の専用データなど種々のデータが存在する。現在トランスパースな検索が可能なデータは、退院要約、患者基本情報、検査結果、肺機能検査であり、科別患者ファイルや科別退院要約ファイル、ICDコード別ファイルなどのインバーテッドファイルを利用し検索対象患者を絞る。さらにデータ辞書を利用しデータ項目を用いた条件式を作成し、条件別患者の抽出を行なう。(図1) データ辞書の標準形

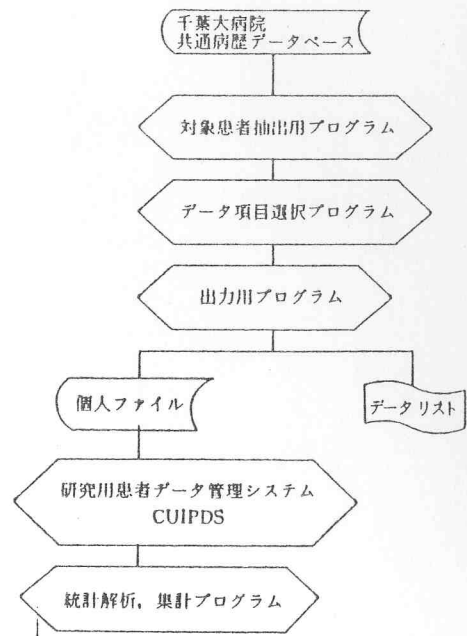


Fig. 1 The retrieval system at Chiba University Hospital

を設定して、ここにデータの多様性を吸収しておくこと、たとえデータベース自身の標準化が充分でない場合でも、かなりの比較性の向上が期待される。本院のデータ辞書には、データ自身を一意に指定するデータ名が記録しており、またそのデータが存在するグローバルデータの定義書にもなっており、システムがこれを解釈してデータの概念的位置を示すようになっている。またデータの付加的情報も辞書に記録しており、柔軟な検索の手助けとなっている。(図2)

```

^GDI("GOT") = %GOT      %PMMX%ID^10^DDATE^1^SEQ^1^:1:1:1:1:
^GDI("GOT",1) = ^7^40^30^0^MU/ML^
^GDI("GPT") = %GPT      %PMMX%ID^10^DDATE^1^SEQ^1^:1:2:1:1:
^GDI("GPT",1) = ^7^40^30^0^MU/ML^
^GDI("GTT-120") = %GTT-120 %PMMX%ID^10^DDATE^2^SEQ^1^:1:6:1:1:
^GDI("GTT-120",1) = ^60^110^40^50^MG/DL^
^GDI("GTT-15") = %GTT-15 %PMMX%ID^10^DDATE^2^SEQ^1^:1:2:1:1:
^GDI("GTT-15",1) = ^100^200^40^50^MG/DL^

```

Fig.2 Data dictionary of data-item

### プログラム設計およびコーディング

複数の人がプログラムを作成するのであるから、プログラムが作成者以外の人にとっても理解し易く、全体のシステムの中の位置付けが解るような工夫が必要であり、これらの事項は標準化以前の問題であろう。

本院のルーチンにおける1行目は、必ず次の形式になっておりルーチンを識別する際のコメント情報を与えている。

ルーチン名; SAVED ルーチン名; 内容を示すコメント; 作成日; 作成者名

また多くの業務的なルーチンに共通に用いる変数については、約束変数として共通に用いる工夫をしている。例えば、CRT:端末番号、DEVICE:プリンター番号、DAT,DATE:本日日付け などである。ルーチン名については、関連業務に関する一連のルーチン群については頭4ないし5文字を同一にしている。例えば、看護管理業務の場合には^KANGK00,^KANGK01,^KANGK02,... などとしている。

### 運用、教育について

大学病院のように、人の異動(特にドクターの)が頻繁にある施設においては、コンピュータ利用者が簡単な操作によって目的の情報を得られる様、操作性の向上を企らなければならない。また一方では、独自のプログラムを開発し利用したい研究者に対するコンピュータ利用法も考えておかななくてはならない。本院では、操作の簡便さおよび機密保護、データ破壊の危険性回避等を考慮し、業務的な内容についてのコンピュータの利用が中心と考えられる病棟および外来に設置されている端末においては、リターンキーを押下することによりメニュー画面(図3)が出され、

そのメニューに含まれている処理しか実行できないような、tied-terminal としてある。通常必要と思われる業務については、ほぼそのメニューに含まれているが、各ユーザーが独自に開発したルーチンについては 9番の特定業務の中に組み込んで実行できるようにしている。また診療研究サポートという意味で14番の研究データ管理 (CUPIDS) を提供し、病棟および外来の端末よりドクターが独自のデータを登録・管理しさらに統計解析を行えるようにしている。端末の操作方法については、メニュー方法を中心としているが、一応システムマニュアルを準備し便宜を企っている。システムマニュアルの利用法として通常検索処理の端末操作ガイドとしての役割も含まれるが、先のCUPIDSの利用法の説明にほぼ半分のページをさいている。(図4) それだけのページを削いているという理由は、CUPIDSのシステムの細かい操作方法のみならずデータ構造の概念やアウトプットの意味も含めた解説書としての役割も担っているからである。またマニュアルの最後にはデータ略称名(表1)も記載されており、データ項目を直接指定する検索方法のための辞書的役割も果たしている。

教育に関してはCUPIDSおよびCUPIDSに関連する統計解析手法の解説を不定期に講習会を開き行っている程度である。

\*\*\*\*\* 業務の選択 \*\*\*\*\*

次の表から選択してその番号を入力して下さい。

- |                           |                          |
|---------------------------|--------------------------|
| 1 -- 患者データ検索              | 10 -- ゲーム各種              |
| 2 -- 検査データ印刷 ( デイリー 報告 )  | 11 -- 各種患者リスト            |
| 3 -- 検査データ速報              | 12 -- 在院患者リスト            |
| 4 -- 患者属性問い合わせ ( ID, 名前 ) | 13 -- 統計計算 ( 電卓 )        |
| 5 -- 入退院, 転科報告            | 14 -- 研究データ管理 ( CUPIDS ) |
| 6 -- 給食オーダー               | END -- 終了                |
| 7 -- 看護関係                 |                          |
| 8 -- 薬剤情報 ( D1 )          | 番号 =                     |
| 9 -- 特定業務                 |                          |

Fig.3 Sample display of first menu in retrieval system

### 退院サマリーについて

各種病歴・診療データの中で標準化しやすいものと考えられるものに、退院サマ

リーが上げられる。本院では退院サマリーの様式を標準化し全科共通様式を作成しているが、各診療科により必要な情報が異なるので、各科独自の様式も設定している。現在21診療科のうち7診療科で各科様式として登録されており、いずれもコンピュータシステムに入力されている。検索については患者単位についても、データ項目指定によるトランスパースな検索も可能になっており、データ項目としては項目名の先頭がSになっている。(表2)

目 次	
① 情報化社会と病院	4
② システムの歴史と未来	6
③ 端末機器の使い方	10
④ 病歴データの検索	20
⑤ 検査システムの概要	34
⑥ 検査報告の受け方	36
⑦ その他の管理業務システム	37
⑧ 外来端末での業務	40
⑨ 個人別データ管理システム(CUIPDS)	42
附録1 端末番号表	69
附録2 ICD コードについて	71
データ略称名	72

Fig.4 Table of contents in system manual

### まとめ

個々の病院内での標準化が企画され推進されていると思われる。大学病院のように大所帯でしかも人の流れが多い病院にあってはとくに院内における標準化も思うようにいかない部分も多いと思われるが、そろそろ病院間、施設間の標準化に目を向けて行く必要があろう。例えば退院サマリーの標準化については日本医療情報学会の「病歴データのコーディングと標準化に関する研究会」の方で議論されており試案が出る段階にまで来ている。本学会においてもマンプス言語仕様以外の部分の標準化についてスタートする時期にきているのではなかろうか。

### 文献

- 1) 里村洋一(1983): 病歴データと医事システム, 病院情報システム. 医典社: 9-15
- 2) 本多正幸・里村洋一(1985): 診療録データベースの検索にかかわる標準化.



医療情報学 5 ( 1 ) : 66-74

3) 西三郎 (1985) : 病歴データの標準化とプライバシー保護. 医療情報学 5 ( 1 ) : 75-80

Table 1 Part of the data dictionary in a general format

1. 患者の基本的識別情報

ID	患者番号 (7ケタ)
NAME	患者氏名 (カタカナ)
SEX	性別 ("F" 又は "M")
BIRTH	生年月日 (例 5 19380630)
	↓
ADDRESS	住所コード (JIS 自治体コード)
	(千葉県内 6ケタ, 県外 3ケタ)

2. 検査項目略称

1. 血液学検査

WBC	白血球数	ATY-LY	異型リンパ球	PRO	蛋白定性
RBC	赤血球数	W-OTHER	白血球異状	BIL	ビリルビン
HGB	血色素量	ERYTHR.B	赤芽球異状	BD	潜血
HCT	ヘマトクリット値	ANISO	大小不同	UG	ウロヒリノーゲン
MCV	平均赤血球容積	POIKILO	奇型赤血球	SG	比重
MCH	平均赤血球血色素量	POLYCHRO	多染性赤血球	ニョウチンサ	尿沈澱
		R-AB-C	赤血球異状	RBC/F	赤血球/各視野
		TOXI-O	中毒性顆粒	RBC/SF	赤血球/各視野

Table 2 Part of the data dictionary in a specific format

6. 各科入院要約データ略称名

01 第一内科用

S01-A1, S01-A2	疾患分類	S01-F1.9	腎盂造影	S01-F4.7	その他のRIシンチグラム
S01-B1~S01-B9	入院時所見	S01-F1.10	リンパ管造影	S01-F5.1	食道の内視鏡検査
S01-C1	輸血の既往	S01-F1.11	その他の造影	S01-F5.2	胃十二指腸の
S01-C2	手術の既往	S01-F2.1	肝の超音波検査	S01-F5.3	大腸の
S01-C3	薬剤過敏症	S01-F2.2	胆道の	S01-F5.4	腹腔鏡の
S01-C4	常用薬	S01-F2.3	脾の	S01-F6.1	眼底所見
S01-C5	異常体重	S01-F2.4	脾の	S01-O1	心電図
S01-C6~S01-C10	既往疾患	S01-F2.5	腎の	S01-O2.1	体重50g
S01-D1	悪性腫瘍の家族歴	S01-F2.6	消化管の	S01-O2.2	体重75g
S01-D2	肝疾患の家族歴	S01-F2.6	その他の	S01-O3	消化吸収試験
S01-D3	糖尿病の家族歴	S01-F3.1	胸部のX線CT	S01-O4	尿機能検査
S01-D4	高血圧症の家族歴	S01-F3.2	肺結核の	S01-O5.1	血液の細菌検査
S01-E1	タバコ本数	S01-F3.3	胸部の	S01-O5.2	尿の
S01-E2	喫煙年数	S01-F3.4	肝の	S01-O5.3	鼻の
S01-E3	アルコール量	S01-F3.5	胆道の	S01-O5.4	唾液の
		S01-F3.6	脾の	S01-O5.5	胆汁の
		S01-F3.7	脾の	S01-O5.6	唾液の

# On Standardization of Hospital Information System in Chiba University Hospital

Masayuki Honda, Yoichi Satomura

Division of Medical Informatics, Chiba University Hospital\*

The standardization is one of the most important tactics in order to realize the hospital information system, not only for inter-hospital communication but also for effective data processing in a single hospital. This paper presents the real cases of standardization in Chiba University Hospital.

First, real aspects of standardization of, (1) its range and scale, (2) relation between special and common events, and (3) data and database protection, are introduced. Second, the principals of standardization in Chiba University Hospital are shown the following items, global design, programming design and coding, user education and system management, and discharge summary.

---

Information authors. Masayuki Honda Yoichi Satomura

\*1-8-1 Inohana, Chiba-shi, Chiba 280, Japan

Phone: 0472-22-7171 (ex. 3291)



# 臨床医学を発展させる フィードバックシステム

野瀬 善明

九州大学医学部附属病院 医療情報部\*

Key words: Medical information system, feed back system

## はじめに

医療情報システムの役割りのひとつは、臨床データの集約と再編成を行い、研究成果を高め、この研究成果を臨床の現場に還元して医療水準を向上させることにあると思われる。この研究サイクルのフィードバックシステムになりえると思われる本院の事例に計画診療支援システム、疾病登録システム、ハイリスクグループ追跡システムがある。このようなフィードバックシステムは臨床医学の発展に寄与できると思われる。

臨床医学は基礎医学と異なり、人為的に均一条件下での実験を繰り返すことができない。日々の診療で取り扱う、多因子の絡む、不均一な、再現できない患者のデータ解析から新しい学問を創らねばならない。さらに困ったことに、臨床データは分散しているし、標準化されてもいないために、研究材料とすることが困難である。臨床データを集約・再編成して研究対象としうるデータに質を高める必要がある。医療情報システムの役割りのひとつは、臨床データの集約と再編成を行い、研究成果を高め、この研究成果を臨床の現場に還元して、医療水準を向上させることであろう。この臨床医学の科学としての研究サイクルのフィードバックシステムを創ることが、臨床医学の発展に寄与するものと期待されている。

臨床医学を発展させるフィードバックシステムを創る試みとして、九州大学のこれまでの14年間の活動のうち、このような観点に合致し、その観点から実践すればさらに発展させうると考えられる例には以下のものがある。

## 計画診療支援システム

新しい治療法の優秀さを科学的に立証するために、プロスペクティブな比較対照臨床試験が義務付けられている。我が国には、これを行なうシステム化さ

---

Information for authours: Yoshiaki Nose.

\*〒812 福岡市東区馬出3丁目1-1 (電話: 092-641-1151)

れた研究体制が見当たらない。そのため国際的に通用するPhase IIIの研究が極めて少ない。計画診療支援システムをつくり、基礎医学研究の成果を実用化させる研究体制を本学につくることを目指している。現在、8診療科で約1500名の胃癌、白血病、悪性リンパ腫、子宮頸癌、肝癌、肝硬変、肝炎を対象にした、プロスペクティブ研究を支援している。計画診療支援システムは、臨床医学を発展させるフィードバックシステムになりうると思われる。

#### 疾病登録システム

現在、7500名の癌患者の病歴学術データの集約と再編成を行っている。毎年3000名を追加登録している。WHOの規準に従って、病名、病理所見、検査成績、治療歴、死因、死亡日などを初診日から最終来院日まで、入院も外来も整然と集約している。治療の延命効果や、癌の自然経過の研究をしやすい場ができた。また、癌研究の対象患者を研究目的に応じて抽出することができるので、研究効率を高めることができると期待される。この病歴学術データの集約システムは癌以外のあらゆる疾病についても、情報の集約化と再編成を同じ手法で実行できる。研究対象疾患を拡大して、多数のレトロスペクティブな臨床研究を促進できると思われる。

#### ハイリスクグループ追跡システム

本院では現在、約6000名の輸血後患者の病歴を集約・再編成している。このシステムを使って、輸血部ではATL抗体陽性の血液を輸血された患者は高い確率で抗体が陽性化することを、世界に先駆けて立証した。この研究成果は、ATL抗体陽性血は輸血に供給しないという対策が全国一斉にとられるようになったことで、臨床の現場に還元された。また、輸血後肝炎の発症危険因子を探索して、予防対策に利用しようとの研究や、AIDS発症の危険のある血液製剤を過去に投与された患者の監視にも利用されている。薬物治療、外科手術、放射線治療などの特定の治療を受けた患者に起こりうる遅発性副作用および医源性疾患の監視と追跡のシステムなども医療の質の向上に役立つフィードバックシステムとなり得ると思われる。

#### む す び

このように、臨床データの集約と再編成を通して、臨床研究がしやすい場を提供し、臨床研究の成果を高め、この成果を再び臨床の現場に還元することによって、医療情報システムは臨床医学の発展をもたらすことが着実にできると考えられる。

# A Feedback System Contributing for the Progress of Medicine

Yoshiaki Nose

Information Science Laboratory for Biomedicine,  
Kyushu University Hospital \*

One of roles in medical information system is a feedback system between clinical practice and medical research. Clinical data should be gathered and prepared for research to obtain excellent results. These results are brought-back to clinical practice and get good expectations. This feedback system could contribute to progress of clinical medicine. Clinical medicine is different from fundamental medicine. It can not repeat experiments under an artificial uniform condition. Then the well planned gathering system and the sophisticated preparation system for medical information are essential in the scientific meaning. The author hopes that the above opinion is discussed among the people who work in the field of medical information system, and blushed-up with the criticism. Samples are a computed support system for protocol practice, a specific disease registration system such as a hospital cancer registration, and a follow-up system for a high-risk patient group. These feedback systems are in the use at Kyushu University Hospital.

---

\*3-1-1, Maidashi Higashi - ku Fukuoka - shi 812, Japan Phone: 092 - 641 - 1151



# Small Concept\*

Kensuke Baba

Department of Pathology and Surgical Pathology, School of Medicine,  
University of Occupational and Environmental Health, Japan\*\*

Key word: Small Concept

"Small concept" is a subroutine-like programming unit designed to be utilized by general programmers. The author described two examples of previously reported "small concepts", i.e. the execute  $\wedge$ e and the do  $\wedge$ c, in order to show what is the small concept. The author calls the authorized small concept "concept". The concepts should be standardized in extrinsic specification and in their capabilities by an authority such as MUG-Japan. When every MUMPS vendor submits himself to provide every concepts, a new co-operative community will be created, where concepts can be mutually implanted in different individual systems and used in any application program at any time. In the new community, a concept which contains Z-commands, Z-functions, system-dependent control characters etc., is also transferable among different systems, because the extrinsic features of concepts are thoroughly standardized. The author emphasizes MUG-Japan should go into action now for the establishment of some authorized body and appeal it strong to foreign MUGs and MDCCs.

## INTRODUCTION

MUMPS is a 'nicely self-synthetic' language. MUGs are 'unions' to cultivate nice environments in which we can efficiently use it. To standardize the language is the most fundamental requirement for creating such an environment. But that is not all and is only the first

---

\* Major part of this study was presented at 14th Scientific Conference on MUMPS,  
Annual Meeting of MUG-Japan (*in Japanese*)

\*\* 1-1 Iseigaoka, Yahatanishi, Kitakyushu, Fukuoka 807 Japan



step for bringing out the self-synthetic capability and distributing the brought out know-how to general MUMPS users. A common concept should be established to indicate the standardized programming unit containing the brought-out know-how. The term "concept" may be one of the best term for this standardized unit.

When a standardization authority for concepts successfully submits all MUMPS vendors to deliver the concepts that adhere to the extrinsic specification which has been prescribed by the authority for each concept, it will realize a highly advanced 'cooperative community'. In that community, the concept can be mutually implanted in different individual systems and used in any application program at any time, admitting that Z-commands, Z-functions, system-dependent control characters etc. are contained in the concept.

The term "small concept" is now a common term in MUG-Japan. The small concept is identical with the concept but has not been standardized. The author spends the major part in describing examples of small concepts in order to show what is the small concept and how useful it is. And then he presents a rational extrinsic specification of the concept.

#### DEFINITION OF SMALL CONCEPT AND CONCEPT

"Small concept"<sup>1)</sup> is defined as a subroutine-like programming unit designed to be used commonly in any program as a utility upon different systems within the standard MUMPS system. To be designed as a programming unit is *sine qua non* of the small concept. The small concepts are stored up in the routine directory (routine form) or in the global file (global form). The former should be designed to be inserted into and called from any program by the DO command with or without parameter passing or a \$\$function. The latter should be designed to be inserted into a current routine or work by the XECUTE command.

One of the best examples of the small concept may be %red^%CRT which sets the escape sequence text to write characters red on the local %red. It is one of the smallest of all small concepts. The examples of the largest case are the global, ^%e, and the routine, %c, which are explained later. However, the definition of the small concept does not lay down a restriction on its size, insisting that the notion of the small concept should be simple. A program unit which is coded in a

current routine and defined by a local variable is not a small concept as a rule.

Only the small concept which has reached the standard is permitted by the standardization authority to be called "a concept". To be designed as a program unit, and to be standardized in extrinsic feature and in action capabilities are the second *sine qua non* of the concept. However, not only the intrinsic coding style is freed if rational, but also it may contain system dependent language elements such as Z-commands, Z-functions, not standardized escape sequence, etc.

### SCOPE OF SMALL CONCEPT

The small concept encompasses: mathematical functions, character transforming, cursor position reporting, age pattern match logic supplying, escape sequence loading, inverse file managing, string editing, intra-MUMPS front processing etc. (Table 1).

The DO ^%RR is excluded from the small concept according to the definition of the small concept. Namely, because, to use it in a program is not the main purpose of the DO ^%RR utility and it may never be used in any application program.

Table 1. CLASSIFIED SCOPE OF SMALL CONCEPT

---

#### SUBFUNCTIONS<sup>[1]</sup>:

Mathematical functions.

String functions including advanced \$TR and \$FN functions.

#### SUBCONSTANTS<sup>[2]</sup>:

Device status reporters including the cursor position reporter.

Check logic getter including the age pattern match logic getter.

#### SUBLOCALS<sup>[3]</sup>:

Advanced device control setters including some CRT escape sequence setters.

#### SUBCOMMANDS<sup>[4]</sup>:

Non-interactive processors including the main-inverse file manager.

Interactive processors including the string editors and the intra-MUMPS front processors.

---

[1]: is called by \$\$...{^...}(...) and transforms passed parameter to returning value as a function.

[2]: is called by \$\$...{^...}.

[3]: is called by do...{^...} or do...{^...}(...) and sets returning value on local(s).

[4]: is called by do command or xecute command and executes intra-routine or intraglobal command line(s) with or without interactive operations.

## EXAMPLES OF SMALL CONCEPTS

XECUTE ^%e:

The XECUTE ^%e small concept<sup>2)</sup> is one of the most useful small concepts. This type of small concept or other similar ones are called "string editor"<sup>3)</sup>, "one line editor"<sup>1)</sup> or "edit-able READ"<sup>4)</sup> in Japanese. The XECUTE ^%e small concept which was previously reported in Japanese will be introduced in an international language. The small concept works within SP-MUMPS on PC-9800 and VAX DSM.

The XECUTE ^%e edits the string which is designated by the variable, %, according to the interactive key operation, and returns the edited value to %. The initial position of the cursor on the string to be edited can be designated by a parameter (%P). Users may select the initial status of insert/overwrite mode by a parameter (%O). The help level is also selected by a parameter (%H).

The small concept is constructed by following two globals and the globals which will be mentioned later using examples.

```
^%e =
  w *27,"[s" f %i=0:0 r *%a:1 x ^%e($s(%a<32:%a,1:32)) q:%i
^%e(-1) =
  w *27,"[u",%,*27,"[u",$(%,1,%P-1)
```

Where % is an editing string, \*27,"[s" is an escape sequence to set the home position of the cursor at the current position of the cursor. \*27,"[u" is an escape sequence to move the cursor to the home position (set previously by \*27,"[s"). %P is the current cursor position. When no keys are operated in more than one second, ^%e(-1) works to display the value of % completely and locates the cursor at the position of %P. Otherwise, the MUMPS control moves to ^%e(ASCII code of the pushed key) or ^%e(32), when a control key or any printing key is pushed respectively.

```
^%e(8) = (when the ← key is pushed)
  w *8 s %P=%P-1
...
^%e(12) = (when the → key is pushed)
  w $(%,%P) s %P=%P+1
...
^%e(24) = (when the DEL key is pushed)
  w " " s %=$(%,1,%P-1)_$(%,%P+1,255)
```

...

```

^%e(32) =      (when the printing character key is pushed)
  x ^%e(%O/10+32.1)
^%e(32.1) =      (overwriting mode)
  w $c(%a) s %=$e(%,1,%P-1)_ $c(%a)_ $e(%,%P+1,255) s %P=%P+1
^%e(32.2) =      (inserting mode)
  w $c(%a) s %=$e(%,1,%P-1)_ $c(%a)_ $e(%,%P,255) s %P=%P+1

```

Here, %O=0 when the insert/overwrite mode is in the insert status. Otherwise, %O=1. The value of %O is reciprocally turned by INS key operation.

For instance, when the DEL key is continuously pushed, spaces are continuously overwritten onto the displayed string, namely, the edited string is not displayed simultaneously, but the string (%) is updated correctly and simultaneously, and the edited string (%) correctly appears when the key operation pauses for one second. Likewise, for other key operation, ^%e cleverly leaves out the concurrent update display of the updated string (%) until the key operation pauses for one second. There are no differences in cursor movement between this small concept and ordinary word processors, but one line.

Table 2. KEY INSTRUCTION OF ^%e

KEY	CODE	ACTION
TAB	9	changes mode of cursor movement (skip_8_characters/no_skip).
→	12	moves cursor forward.
←	8	moves cursor backward.
HOME	26	moves cursor reciprocally to the top or end of string.
INS	18	changes insert/overwrite mode.
DEL	24	deletes the active character.
ctrl^	30	defines starting point of the sub-string to be stored in global with a name.
ctrl¥	28	calls the stored sub-string with a name.
ctrl_	31	enables Kanji input.
RETURN	13	exits the ^%e.

As you may understand, the response to key operation is quick and smart. The capability of XECUTE ^%e is incomparably superior to the \$ZEDIT function which is provided by some MUMPS. In order to sketch the capability of the ^% small concept, the key instruction of the small concept is summarized in Table 2. When we

set %= "" and issue XECUTE ^%e in a routine, it works as an 'editable READ command'. The XECUTE ^%e small concept has been fully utilized in a full screen editor named "%ee"<sup>3)</sup>.

DO ^%c and DO link^%c:

The MUMPS language allows the \$PIECE function to construct easily a structural record with the aid of delimiters. In other words, a record of MUMPS can contain many variable length items, which are structurally defined by delimiters. MUMPS has certainly been advanced in language specification. However, does the MUMPS environment in which it is used provide us with a way to extend the fixed CRT screen space to receive variable length string or data in the limited area of CRT? As long as MUMPS is a "self-synthetic" language, this environment can be developed by itself as this small concept. The author hopes this small concept will be elevated to a concept and distributed to general users as a public domain in the near future.

The concept called by DO ^%c is a small concept to provide an appropriate environment to input length undetermined data (variable length item value) *via* a screen. This small concept works on the SP-MUMPS. The small concept, %c, provides a cell of any length but single line at any location on a screen. When the editing or inputting item string overruns the length of the cell size, the item string scrolls horizontally within the cell. The full string is displayed on an optionally available monitor with a concurrently running pseudo-cursor. The current cursor position on the item string and the current length of the item string may be numerically shown if necessary. The string monitor and the numerical display are closed when the task for the next item input starts. A user may set the length of his cell to be longer than the full length of the item string if necessary. Another user may design a horizontally scrolling word processor, the MUMPS editor, etc., very easily, when he sets the length of the cell to be as long as the horizontal size of his screen.

Both capabilities of the concepts called by DO link^%c and DO ^%c are essentially the same, but the former is linked with 'item array'<sup>5)</sup> which is connected to intelligent globals which have information about item check logic, item name, next item accessing pointer, etc., as shown in Table 3.

Table 3 COOPERATIVE GLOBALS / DO link^%c

GLOBAL NAME	EXPLANATION
<code>^%c(taskname, ..., "X")</code>	X-position of top of the cell (for each item)
<code>^%c(taskname, ..., "Y")</code>	Y-position of top of the cell (for each item)
<code>^%c(taskname, ..., "L")</code>	Length of the cell (bite) (for each item)
<code>^%c(taskname, ..., "N")</code>	Next cell's node (for each item)
<code>^%c(taskname, ..., "B")</code>	Previous cell's node (for each item)
<code>^%c(taskname, ..., "C")</code>	Check logic for each item string (for each item)
<code>^%c(taskname, ..., "I")</code>	Heading of the item (for each item)
<code>^%c(taskname, ..., "IX")</code>	X-position of item top (for each item)
<code>^%c(taskname, ..., "IY")</code>	Y-position of each item top (for each item)
<code>^%c(taskname, ..., "MX")</code>	X-position of monitor top (for each item)
<code>^%c(taskname, ..., "MY")</code>	Y-position of monitor top (for each item)

## REMARKS AND SUGGESTIONS

How to enhance the through-put capability of the small concept should be mentioned generally. The following two sidesteps are suggested: (1) to design, for instance, the red small concept which is called by DO `%red^%CRT` to set the red escape sequence on a local (`%red`), which may be issued here and there in routines, instead of duplicative issue of `$$%red^%CRTs`, and (2) to provide an intraglobal tool to insert the assigned small concepts into a current routine. For the purpose of the second sidestep, to start the entrance reference name of the small concept with `%` is suggested for distinguishing it from the programmer's routine label.

For the purpose of house keeping of the small concept, similar small concepts would be put together in a routine as `%red^%CRT, %cyan^%CRT, ...; %forward^%CURSOR, %back^%CURSOR, ... etc.`

In order to distinguish the concept which is standardized by an authority from not standardized one, the following syntax is most suggested to a routine as well as a global from, respectively:

`{%}^%lowercasecharacters`

and

`^%lowercasecharacters`

If the MDC endorses the name syntax for specified routines for concepts such as %%... or \$... for this purpose, it will be the best.

The establishment of a subcommittee which is specified to standardize \$\$functions has been proposed<sup>6)</sup> in the MDC which elevated it to type A. There are no rational reasons to limit it within \$\$functions and subroutines which will be called by \$\$function. The small concept which is designed to be called by a parameter passing DO, by passing-parameter-less DO and by XECUTE also should be standardized as well as those which are called by \$\$functions.

Essentially the concept can absorb non-standard device control into the current system. The author believes that the concept absorbs well the European proposal for MDC to handle devices including not standardized CRT, page printer, *etc.*, within the current standard. Because, (1) the concept is standardized when the general programmers use it, (2) the concept for a special device, therefore, can be used in a different system which is connected to the same special device, (3) the concept has sidesteps for through-put problems, (4) if it has through-put problems in a very special case, the vendor will enhance the concept containing system dependent language element such as \$ZCALL function up-keeping its standard extrinsic specification, (5) the concept is easier to code, and (6) the concepts for device handling, herefore, will spread rapidly and widely.

The classification of the small concept shown in table 1 is done according to the following two major philosophies: (1) The small concept which is called by a \$\$function should never execute, but return the value(s) to the \$\$function. (2) The four categories should be distinguished by the entrance syntax of the small concept.

The first philosophy is originated from the most fundamental philosophy for designing language architecture of MUMPS. According to the first philosophy, for example, \$\$put^%CURSOR is replaced by DO %put^%CURSOR. The basic architecture of MUMPS language distinguishes between the roles of commands and those of functions very clearly. MUMPS is not the c language. MUMPSist should exclude the incorrect usage of function such as \$ZCALL("KANJIENABLE") function from the standard.

According to the second philosophy, the following rules for entrance reference syntax are advisable for a subfunction, a subconstant, a sublocal and a subcommand (see Table 1), respectively:

```

%lowercasecharacters,
%oneuppercasecharacter_lowercasecharacters
%uppercasecharacters

```

and

*none.*

### FUTURE VIEW OF THE SMALL CONCEPT

To code commonly useful small concepts should be the main interest of senior MUMPSist. To establish a general rule for extrinsic feature of "concepts", to show the extrinsic feature and capability to validate it, and to encourage vendors to implant it should be the matter of the highest priority for the MUGs. Are not the MUGs such a party?

The authority should make efforts to produce new useful concepts without any system dependent language elements. A senior MUMPS user may show a new idea on a concept with its extrinsic features. The authority will accept his specification as the standard specification of the new concept with or without modification and edition if it is rational. Another senior user may apply for the authorization of his own product as a concept. The accumulated resources in the form of concept under the administration of the authority may be rationally maintained and equally as well as efficiently distributed as a public domain. The authority should also validate the individual concept which has been provided by vendors.

When each vendor provides every concepts which the authority entacted, a new world is created. In the new world, users may use the concepts on different MUMPS systems. In the new world, programs may be transplanted among different MUMPS systems much more easily than now. In the new world, implementer may use system dependent elements such as \$ZCALL functions in the concept without any interference in program transferability, because the extrinsic feature of each concept which is used in the program is thoroughly standardized. Moreover, in the new world, program resources are mutually utilized, even they conatin system dependet elements, as long as the not standardized elements are thoroughly cabined in the concepts.



## REFERENCE

- 1) Baba, K. & Takyu, H. (1984): Small concept. Proceedings of MUG-Japan Meeting. 11: 45 - 50. (*in Japanese*)
- 2) Baba, K. & Takyu, H. (1985): Common intrinsic structure of small concept. Proceedings of MUG-Japan Meeting 12: 71 - 74 (*in Japanese*)
- 3) Baba, K. & Takyu, H. (1986): String edit small concept. Proceedings of MUG-Japan Meeting 13: 138 - 146 (*in Japanese*)
- 4) Kemuriyama, T. (1986): Personal communication.
- 5) Baba, K. & Takyu, H. (1986): An application of the string edit small concept and item array technique —— data-input/editing on screen. Proceedings of MUG-Japan Meeting 13: 147 - 156 (*in Japanese*)
- 6) Brown, D.B. (1986): Suggestion for future of MUMPS. Proceeding of MDC Meeting 34: 10 -11.

Letter to the Editor:           The 7th MDCC-Japan held at Osaka resolved to include "concept" into the main job of MDCC-Japan. The committee will have a function as *de facto* Bureau of Standard to standardize the extrinsic features and action specification of the concepts, *de facto* Police Office to validate individual versions, *de facto* Publisher to edit, enhance and distribute the concepts. The committee started discussions on how to generalize the rules of extrinsic specification of the concept and the collection of outstanding utilities which can be incorporated into one of the concepts.

# スモールコンセプト\*

馬場 謙介

産業医科大学第二病理学教室\*\*

MUMPS 優れた自己生成 (Self-synthetic)能力のある言語であり、従って、自身で、自身をより高級な環境に置き得ることをMUMPS人はよく承知している。MUG の存在価値の一つは、この優れた自己生成機能を共同でうまく使うための'共同組合'としての価値であることも認識している。言語の標準化は、良い環境を実現するための第一の段階ではあるが、それが全てではないことに目を向けたい。

優れた自己生成機能を使って、優れた MUMPS 利用技術を共同して使える具体的な方策の一つとして、small concept が、MUG-Japan では知られている。Small concept は、一般ユーザがいつでもどのプログラムでも使えるように設計された(ライブラリタイプの)プログラミングユニットと定義できる。見方を変えると、small concept は、命令や関数などの言語要素からなる(従って、これらより大きい、プログラムより小さくまとめられた)「言語利用技術単位」である。本論文で著者は、既に日本語で発表した `xecute ^%e` と `do ^%c` の二つの small concept を例示して、small concept の理解を促した。この二つは、いずれも、UCD MUMPS の \$ZEDIT 関数に似ているが、その機能は比較にならないほど多彩で、応用範囲は極めて広い(前者を用いたフルスクリーンエディタがすでに公開されている)。

著者は、Small concept が、特定の機関で規格化すべきであることを強調した。その特定の機関が規格化された small concept を仮に concept と呼ぶことにした上で、concept の外部仕様を具体的に提案した。全てのベンダーが全ての concept を提供してくれる状況になったとき、新しい「MUMPS 利用共同社会」ができることは見逃せない。この状況下では、concept に、装置依存性の言語要素が含まれていても、移送性になんら支障を与えない。この新しい共同社会では、利用技術のエッセンスが集約された concept が、共同の財産として利用できる。更に、実際には装置依存性の言語要素を含む資源でさえ共用することができる。著者は、この目的のために、MUG-JAPAN が、率先して、規格制定のための機関をつくることを提言した。

---

\* この論文の主要部分は、第14回日本MUMPS学会で発表した。

\*\* 〒807 北九州市八幡西区医生か丘 1-1 TEL: 093-603-1611 (EXT: 2323) FAX: 093-603-1611



# MUMPSによるトロトラスト被注入者 剖検例のデータベース

石川 雄一\* 日吉 徹\* 畠山 茂\* 田久 浩志\*\* 馬場 謙介\*\*\* 森 武三郎\*\*\*\*

\*東京医科歯科大学 医学部第一病理学教室 \*\*産業医科大学 振動研究室

\*\*\*同 第二病理学教室 \*\*\*\*放射線医学総合研究所 生理病理研究部

Key words: Thorotrast injection, date base

トロトラストは主として第二次大戦中に使われたX線造影剤で、40年以上たった今日、その晩発傷害が問題となっている。主成分は、自然放射性核種であるトリウムであり、物理学的半減期140億年・生物学的半減期200-400年であるから、非常にゆっくりとではあるがその顆粒の沈着した臓器（主として肝、脾、骨髄）を長期にわたって $\alpha$ 線で照射するので、肝がん・肝脾の線維症・白血病などをひきおこす。 $\alpha$ 線放出核種であるウラン・プルトニウム・トリウムなどの商業的利用が世界的規模でおこなわれているにもかかわらず、 $\alpha$ 線についての人類の知識はいまだに極めて不十分である。私たちは、ドイツ、アメリカ、デンマーク、ポルトガルなどの各国と歩調をあわせてトロトラストの生物影響を研究してきたが、このたびそれらをデータベースとしてまとめる試みを行ったので、簡単に紹介する。

## はじめに

私達は病気になったときに、胸のX線撮影を受けることがあるが、このようないわゆる単純撮影のほかに、血管などをレントゲン撮影する目的で、造影剤を管腔に注入して撮影することがある。トロトラストは、こうした場合に用いられるX線造影剤の一種であり、日本では主として第二次大戦中に軍の病院で血管造影に用いられた。ところが、このトロトラストの主成分が、自然放射性核種であるトリウム（ $\alpha$ 線を放出する）であったために、その後40年以上たった今日でも、その後遺症に悩む人々を生み出すことになった。いや、単に、後遺症と言うにとどまらない癌や肝硬変などの命に関わる重い疾患を、次々と発生させることになったのである。もちろん、使われた当時は、半減期が140億年もある極く微量のトリウムが、人体に癌を引き起こすなどとは、多くの人は考えもしなかった。トロトラストは、ドイツ直輸入の高級医薬品として、「お国のために」戦争で傷ついた将兵に優先的に使われたのである。

トロトラストは、いったん血管に注入されると肝臓、脾臓、骨髄といった網内系と呼ばれる諸臓器に沈着し、なかなか排泄されない。そのため、生涯にわたって

---

Information for authors, Yuichi Ishikawa

\*〒113 東京都文京区湯島1丁目5-45(03-813-6111)

それらの臓器を $\alpha$ 線で照射し続け、肝ガンや白血病などの悪性腫瘍を発生させたり、肝や脾の線維症をひきおこしたりする。

私達は、トロトラストによる $\alpha$ 線の内部被曝について、文部省科学研究費の補助のもとで研究を進めており、その結果の保存や検索を容易にするためデータベースの作製を試みている。本稿ではその試作例を紹介するのが目的であるが、まず、この研究のもつ意味と背景について述べてみよう。

### トロトラストによる疾患を研究する目的

ひろく放射線と呼ばれるものには、 $\alpha$ 線、 $\beta$ 線、それに電磁波（X線や $\gamma$ 線もこれに含まれる）などがある（細かく言うと、中性子線などもあるが、ここでは省略する）。私たちの身の回りには、微量であるが常にこれらの放射線が存在していて、私たちの体や、広くは生物の進化にまで様々な影響を及ぼしている。人間に自然に起こる癌の多くも、これらの自然放射線によるのではないかと考えている人もいる。さらに、現在人類は種々の人工放射線源に囲まれて生活しているし、将来もそうであろう。というのは、現代人の生活にとって放射能は必要欠くべからざるものになっているうえに、長期の保管を必要とする大量の放射性物質を我々は既に地上に創り出してしまったからである。例を挙げると、医療の場でのX線や人工アイソトープを用いた検査、工場での非破壊検査、夜光時計、研究用の種々の放射線発生装置やアイソトープ、原子力発電、核兵器、など、枚挙に暇がない。それなのに、人類の放射線についての知識は完全ではない。否、完全どころか $\alpha$ 線についてはほんの僅かしか判っていないと言っている。

自然放射線ひとつをとってみても、これまでよく年間100ミリレムと言われてきたが、研究が進むにつれて実際には200ミリレムの被曝をうけていることが分かってきた。これは、次のような事情による。環境中には、先ほど述べた何種類かの放射線があるが、 $\beta$ 線の寄与は少ないので無視することができる。それで、 $\alpha$ 線と電磁波の寄与分を算出することになるが、これまで、 $\alpha$ 線についてはその環境中における量も人体に対する影響もほとんど判らなかつたため全部無視して、電磁波（X線と $\gamma$ 線）だけで被曝量を算出していた。こうして求めた値が、年間100ミリレムなのである。 $\alpha$ 線による肺や骨の被曝を考慮にいった場合には、約2倍の200ミリレムになるというのだ。このような計算ができるようになったのも、 $\alpha$ 線についての知識がある程度蓄積してきたからであるが、それでも未だに仮定に頼っているところが少なくない。

人体に対する影響があまり判っていないのに、 $\alpha$ 線放出核種であるウラン・プルトニウム・トリウムを用いた原子炉が造られたり計画されたりしている。将来が原子力社会になるかどうか（あるいは、なるべきであるかどうか）は意見の分かれるところであろうが、いずれにせよ早急に $\alpha$ 線の人体に対する影響を十分に調べるべきであろう。

私達の行ってきた研究は、こうした $\alpha$ 線の人体に対する影響を評価するのに幾ばくかの寄与をなしようと考えている。

### トトロラストの人体に及ぼす影響

トトロラストは造影効果だけをとれば優れた医薬品であり、急性傷害はほとんどなかった。それに対して、数年以上たってから発生するいわゆる晩発傷害は、重篤な上にきわめて特徴的である。

たとえば、ふつう肝臓に発生する悪性腫瘍は肝細胞癌が大部分であるが、トトロラストによって生じる肝悪性腫瘍は肝血管肉腫や肝内胆管癌などのやや珍しいものも多く、肝細胞癌は比較的少ない。白血病も起こるが、赤白血病の占める比率が高かったり、前白血球状態（完全な白血球とは言えないが、異型のある白血球が増殖している状態）とすべき症例が多かったりしている。また、こうした悪性腫瘍の増えかたであるが、白血病が注入直後から増加に転ずるのに対し、肝の悪性腫瘍は、12年程度の潜伏期をおいてから増加しはじめる。悪性腫瘍の発生とは別に、肝臓には線維化が生じるが、ふつうの肝硬変（線維化と肝細胞の再生とが両方みられる）とは異なり肝細胞の再生を伴わない肝線維症である。トトロラストは、肝臓・骨髄以外に脾臓やリンパ節にも沈着するが、脾臓の悪性腫瘍は極めて少なく、悪性リンパ腫やリンパ性白血病も増加していない。

ところで、トトロラストによる傷害は、放射線によるものであるから悪性腫瘍がその本体であり、それ以外の死因はほとんど無視できると考えられてきたが、最近、トトロラスト注入者では全ての悪性腫瘍を除いても、対照群に比べて有意に生存率が低下していることが明らかになってきた。このことは、放射線による寿命短縮は発ガンがその本態であって、一般的な意味での加齢は起こらないとする現在の常識を覆すものである。

このほかにも、線量効果関係が直線よりむしろ2次曲線に近いことなど、特徴的なことが幾つかあるが、ここでは割愛する。詳しくは、森（1987）<sup>21</sup>を参照されたい。

### トトロラスト剖検例のデータベース

現在の研究班の構成となったのは1977年からであるが、日本におけるトトロラスト傷害の研究は既に約30年の歴史を有し、収集した解剖例は300例にのぼっている。集められたデータの中には、疫学的なデータはもとより病理解剖学的な所見や長期間にわたる臨床データ、および症例報告や論文に関する記録、さらには、線量測定にかかわるデータが含まれている。これまでは、こうしたデータはカードで保存されていたり、各々の分担者がそれぞれの方法で記録していたが、疫学的な検討を容易にしたり保管のことを考えた場合にはあまり適切な方法とはいえない。そこでコ

ンピュータによるデータベースを用いた整理が考えられるわけであるが、なにしろ種々さまざまなデータがあり、その量は症例によって大きく異なり、項目になりにくいものも少なくない。

例を挙げてみよう。症例Aは、第二次大戦中に中国でトロトラストを注入され、その後ずっと某国立病院で経過観察されていた。そのため、長期間にわたる臨床データが多量に保存されており、最近になって導入された腫瘍マーカーやホールボデー・カウンターによる全身のトリウム量の計測に関する資料も得られている。さらに、詳細な病理解剖所見と学会誌上での症例報告もある。そして、その後トリウム量の推定に用いた標準線源の校正に変更が生じたため、トリウム量の測定に関するデータには鉛筆で書き込みがなされている。

一方、症例Bは、日本病理学会の剖検輯報により初めて解剖が行われたことが知られ、解剖の後、約2年たってから収集されたため、簡単な臨床データと剖検所見のみしか得られていない。

このように症例によってデータの量および質に非常に大きな差のある、いわゆるきかないデータは、コンピュータによるデータベースには馴染みにくいと考えられがちであるが、ここでMUMPSの登場となるのである<sup>3)</sup>。一般のリレーショナル・データベースと違い、MUMPSでは上記のようなデータでも処理することができる。トロトラストのデータベースはいまだ試作段階であるが、その一例を次に示そう。

X1 氏名	X2 改姓	X3 性別	X4 生年月日	X5 グループ
X6 原籍	X7 転籍	X8 現住所	X9 生年	X10 職業
X11 転入	X12 受領	X13 年月	X14 注年	X15 影方
X16 注部	X17 注理	X18 注回	X19 注入	X20 注注
X21 注院	X22 カル	X23 死月	X24 分年	X25 創創
X26 T出	X27 線	X28 臨月	X29 子備	X30 子備
X31 最調	X32 標	X33 剖年	X34 施	X35 原
X36 剖主	X37 剖	X38 剖	X39 コメント	MEMO

対象とするデータベース名を入れて下さい  
 ?:(DIR) .:(終了)  
 表示先を指定して下さい  
 1:(CRT) 2:(PRINTER) 3:(SEQ FILE) 4:(出力せず) 1  
 書式の形式とレベルを指定して下さい ex. 1,1  
 1:(加工) 2:(べたうち) 3:(一部) 1,3  
 検索結果をしまうデータベース名を指定して下さい  
 ?:(DIR)

! :論理和 & :論理積 = :等しい < :以下 > :以上 [ :包含 ' :否定 ( ) :演算順序  
 検索条件を上の変数とMUMPS論理式で記述して下さい X35[15599123

Fig.1 The first picture on CRT when running the program ^KENSAKU.

上の図は、^KENSAKU というプログラムの最初の画面である。X27などの変数を自分の検索したい条件に合わせてMUMPS論理式であらわすと、それにあてはまる症例が表示される（または、ファイルに入れられる）。例えば、X27 '=' とすると、線量番号のある症例がすべて表示され、X35]15599123とすれば、原発は肝で組織型が血管肉腫の症例が出てくる。（1559は、剖検された症例を日本剖検輯報に報告する際の肝臓を表わす番号であり、9123は同じく血管原性肉腫を表わす番号である。）

この条件で検索した場合の結果を下図に示す。

このデータベースでは、条件にあった症例を一度小データベースに移し、それから更にシーケンシャル・ファイルに移して統計計算などを行うことになっている。

実際のデータは次のようにグローバル変数として記録されている。

私のようなMUMPS初心者、一症例あたりのデータが可変長であるということだけでも大変な驚きであって、そのことばかり強調しているが、このほかにもデータの書換え・追加・消去などがとても容易にかつ効率的にできるようである。

以上、データベースについて簡単に説明してきたが、残念ながらこのデータベースを利用して統計計算を行なった結果は、今回は間に合わず紹介できなかった。後の機会に譲りたい。

----- ID: 1087	氏名 ●村 実	-----
改姓	性別 M	生年月日 19191101 グループ T
原籍	転籍	現住所 000036 住所変更
職業	転職	受傷年月 1941 注入年月 1941
注入年齢 30	造影方法 血管	注入理由 下肢
注入回数 22	注入量 015.00	複数注入 E
カルテ	死亡年月 19721025	分類NO
ThNO 1217	検量NO	臨床NO
子備 2	最終調査 1981	標本NO 000022
剖検年次 072	施設 13601	抽出病院 013601
原発・組織	肝・肝内胆管/血管原性肉腫/造血・骨髄/造血異形成・前白血病/	
剖検主病変	肝硬変症NOS/トトロラスト傷害/	
剖検副病変	肋膜炎(非結核性)/腹膜炎NOS/腹膜癒着/腹水貯留/	
死因	肝悪性血管内皮腫	
*** 死因	●島大学剖検No 1351	
*** 剖検	下肢血管造影	
*** 注入法	下肢靭傷	
*** 理由	S-(A)-188	
*** 分類NO	●島県三●郡山●町末貞777-3	
*** 現住所	1351	
*** 施設剖検番号	肝 1400G, 脾 90G 期間 31Y P-1	
*** コメント		

Fig. 2 A case searched by ^KENSAKU with the condition of X35]15599123.

```

^TH("1","1001")="/91/3*23"
^TH("1","1001","1")="小● 稔**M*19230714*T*****10 ***1932 *09*01*07*55**015
.00*E*012301**19581031*S- *1001*****1981 ***"
^TH("1","1001","1","1")="91 トトロラスト注入後26年目に死亡"
^TH("1","1001","2")="05812301 351010400010003000000000$15598163/04:529569579629/
00:$99073$7852@7853@5817$333"
^TH("1","1001","2","1")="3 3255*23 肝 G,脾 G,期間 26Y .P-0"
^THOROT("1001","1")="小● 稔*1*35*19230714*1932 *1141*257* * *C
* *
^THOROT("1001","2")="肝胆管癌*19581031*名古屋大学剖検No 3255*"
^THOROT("1001","3")="*015*下肢血管造影**右大腿骨骨髄炎*26*09*1190*080*"
^THOROT("1001","4")="*****"
^THOROT("1001","5")="**1190*****0080*****"

```

Fig. 3 Global variables recording the Thorotrast raw data.



## おわりに

トロトラストは、ドイツや日本などで使われて少なからず傷害を引き起こしたが、その症例を収集してヨーロッパ各国と比較することで、 $\alpha$ 線の人体影響に関する貴重なデータを世界的な規模で効率よく検討することができる。トロトラスト症のような医原性の疾患は、決して繰り返されてはならないが、しかし、人間の行為は常に不完全であり、ふたたび多くの人が $\alpha$ 線の被曝を受けることが全くないとはいえない。現代のわれわれが為すべきことは、二度とこうした悲劇を繰り返させないことと、「禍を転じて福となす」べく、この不幸な事例を研究して放射能についての人類の知識を増していくことであろう。ただ、そうした知識が十分に蓄積した暁に、人類は放射性物質を安全に利用することができるようになるのか、それとも、ある種の放射性物質の使用はやめることになるのか、今の我々には判らないが。

## 参考文献

1. United Nations Scientific Committee on the Effects of Atomic Radiation (UNSCEAR committee): "Ionizing Radiation: Sources and Biological Effects, 1982 Report to the General Assembly, with annexes", United Nations, New York, 1982.
2. 森 武三郎 : 「トロトラスト」晩発傷害研究の現状 『放射線科学』 VOL. 30, 289-297 (1987).
3. 馬場謙介、田久浩志 : 「トロトラスト」被注入例のデータベースの作成 『文部省科学研究費エネルギー特別研究 トリウム燃料に関する総合的研究 昭和61年度研究成果報告書』 pp283-294.

# Computer Database on Thorotrast-administered Patients Using MUMPS System

Yuichi Ishikawa,\* Toru Hiyoshi,\* Shigeru Hatakeyama,\*  
Hiroshi Takyu,\*\* Kensuke Baba,\*\*\* and Takesaburo Mori\*\*\*\*

- \* Department of Pathology, Tokyo Medical and Dental University School of Medicine
- \*\* Vibration Laboratory, University of Occupational and Environmental Health
- \*\*\* Department of pathology and Surgical Pathology, University of Occupational and Environmental Health
- \*\*\*\* Division of physiology and Pathology, National Institute of Radiological Sciences

Thorotrast is a tradename of X-ray contrast medium composed of 25% thorium dioxide, used mainly in the period of 2nd World War in Japan. Although the medium hardly showed acute damages at the injection, the chronic damages (or late effects) by long-term internal alpha irradiation are in question even now. After injection Thorotrast granules accumulate in liver, spleen, bone marrow and lymph nodes, and cause liver malignancies, liver fibrosis, leukemias etc. as late effects of internal alpha irradiation. The knowledge of effects of alpha particles is very much insufficient although nuclear power with alpha emitting nuclides (e.g. U, Pu, Th) is commercially utilized or planned in the world.

We have studied the biological effects of Thorotrast in association with German, American, Danish and Portuguese groups. In this paper we present a computer database using MUMPS system which is produced by way of trial. The database will be improved from now on.

---

Information for authors.Yuichi Ishikawa  
\*5-45 1-choume Yushima Bunkyou-ku Tokyo Japan 113



# MUMPSにおける YAC II とデバッグツール

藤本 浩司 今泉 幸雄

アップジョン ファーマシューティカルズ リミテッド 総合研究所\*

Key words : YAC II, debugging tools, MUMPS, standardization

YAC II は設計支援の構造化表記法のひとつとして米国で考案されたYAC Chart をもとに富士通が開張したもので、1984年よりソフトウェア開発のための方法論として採用されている。我々は、診断仕様書を開発するにあたって基本設計、論理構造設計、詳細設計について YAC II を MUMPS言語用に改造した。また、日本語表現を統一すると同時に、システムの保守性を高め、開発時の効率化をはかるために独自のDebugging toolを開発し、システムに組み入れた。

## はじめに

演者らは昨年の本学会で「時系列システムにおける保守時の診断仕様書(2)」という報告を行った(第13回日本MUG学会論文集所収)。今回は、この診断仕様書の開発時におけるソフトウェアエンジニアリングの手法を、下記の2点について報告する。

- ①設計Phase からプログラム表記法までにおいては、従来のHIPOにかえて YAC II (Yet Another Control chart II) を導入した。
- ②独自のDebbuging toolを組み込んだ。

## 基本設計と構造化設計におけるYAC IIの導入

基本設計時はジョブの分割とジョブ間におけるデータの移動や機能を、ジョブ構造図、ジョブ概要定義、ジョブ関連図の各用紙を用いて表記し、ジョブ間における共通の処理をまとめ(⇒共通化ルーチン)、内部・外部共通ルーチンとして内部・外部モジュール定義用紙に記述した。これに並行してDataBase設計も行っている。

論理構造設計においては、各ジョブに対しプログラム分割・結合を行って、プログラム構造図、プログラム概要定義、プログラム関連図に YAC II を用いて表現し、DataBase設計とのvalidation checkを行った。

論理構造設計において、各プログラム設計用紙に記述されたのをもとに詳細処理手順用紙に、プログラムステートメントと YAC II 表記法(日本語表現)が対応するように記述した(図1)。共通ルーチン設計に記述されたルーチンも同様である。

---

Information for authours. Kouji Fujimoto.

\*〒370 群馬県高崎市大八木町168 (電話:0273-63-2211)

日本語表現の統一

YAC II の表記法を MUMPS 言語用に改造した時に、次の 2 点について日本語表現を統一した。

- ① 表記記号に対応した日本語の意味と表記法
- ② 記述する日本語文章の動詞の種類と主語・述語の用い方

機能と特徴としてまず第一に、実行中の trace 情報、Global file dump 情報、local-variable dump (全変数あるいは指定変数) 情報が指定されたプログラムあるいは Global file 単位で出力されることがあげられる。第二に、出力装置として指定端末 (会話型) や file (実行後参照) が任意に選択可能であるということである。また、Debug の指定方法としては、プログラム実行前に指定する静的指定方法とプログラム実行中に会話形式に指示する動的指定方法の 2 種類を用意している (図 2)。

おわりに

YAC II を実際に導入した結果、修正の容易性、制御構造の明確さなどにおいて HIPO よりすぐれていたことがわかった。同時に、日本語表現方法の統一により第三者作成のプログラムも誤解なく容易に修正できるようになった。そして、将来への自動プログラミング化への基礎づくりがで



図 1 YACUにおける MUMPSコーディング例

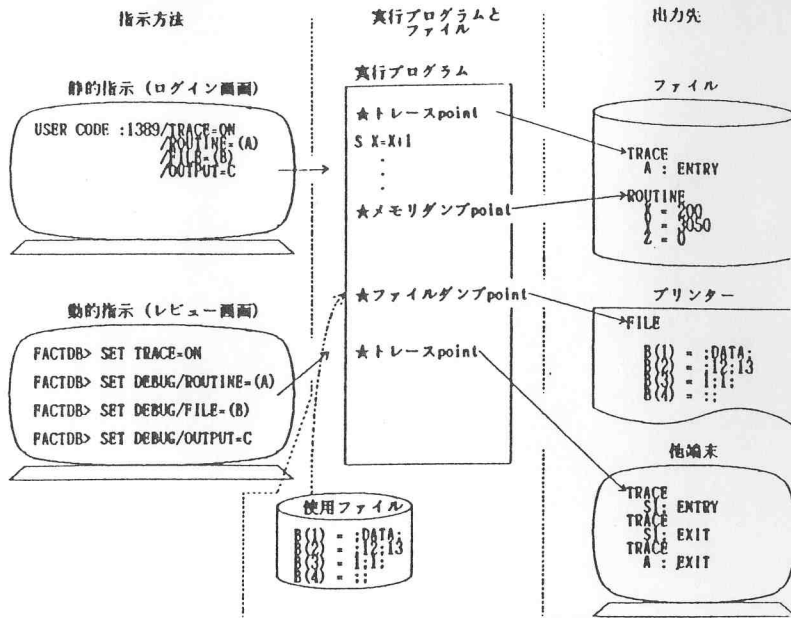


図2 デバッグ・ツールの機能

きたというメリットも得られた。

debugging toolの導入においては、テストングの効率化が促進され、保守性を高めることができた。

### 文献

今泉幸雄, 藤本浩司 梶哲弥 「時系列システムにおける保守時の診断仕様書(2)」  
第13回日本MUG学会論文集 pp 230 - 236 (1986)

2) "FACOM SDEM (V-2)" SE Library IK838, JF827, 富士通(株) (1984)

# YACII & Debugging Tools for MUMPS

Kouji Fujimoto, Yukio Imaizumi

Upjohn Pharmaceuticals Limited, Research Laboratories.\*

We report following two items about the method of software engineering in development of this diagnostic specification.

① YACII (Yet Another Control cart II) was introduced instead of HIPO from design phase to program description method.

② Original debugging tools was installed.

YACII was redesigned by FUJITSU LTD based on YAC Chart which was designed as structured description method for design support in U. S. A. YACII has been adopted for the method for software development since 1984. We transformed YACII into that adapted for MUMPS language for basic design, logic structure design and detail design in the development of diagnostic specification. Also we standardized Japanese description, developed original debugging tools and installed it into the system to upgrade system maintenance and make effective of development.

First thing as function and characteristics is that executed trace information, Global file dump information, local-variable dump (all variable or indicated variable) information can be outputted in each program indicated or in global file. Next is that indicated terminal (conversational type) or file (refer after execution) can be selected independently.

---

Information for authours. Kouji Fujimoto ,et al.

\*168 Oyagi-machi Takasaki-city Gunma-Prefecture Japan 370

# プログラム開発・保守の標準化

古林榮次郎 大榎 陽一 野口 弘 寺村 昌文  
大阪府立羽曳野病院 情報企画室\*

Key words : Standardization, programming, maintenance of program

## はじめに

MUMPSを使用したシステムが、日本で本格的に稼働して10年以上になる。MUMPS導入の最大のメリットは、簡単に修得でき、ユーザ自らがプログラムを作成できるところにある。当初、「Do it yourself」の合言葉がさかんに唱えられこの点が強調された。この事自体は先進的であり、現在でもMUMPSの持つ魅力である。しかしながら、この自給自足体制の適用は、小規模な個人的システムか、「趣味」あるいは「遊び」のシステム以外では、十分な注意が必要である。単に、見掛け上の生産性や効率をねらいとした開発ではなく、真に個々のシステムおよびサブシステム群全体の完全性、一貫性、信頼性、効率性、安全性を達成するものでなければ、システムの成功はあり得ない。MUMPSの世界のみならず、これらを実現する一つの重要なポイントは、システムのライフサイクルの各過程にわたる標準化手法の導入である。

ここでは、プログラム開発・保守に関する標準化に限定して羽曳野病院での標準化の実例を紹介する。

## 標準化の方針

羽曳野病院ソフトウェア開発規程作成に際しての大きな方針は以下のようなものである。

1. MUMPSの特徴を生かした標準化であること。

MUMPSは文字形式の取扱、ファイル構成などに特徴があるが、この長所を生かすことが重要である。受け渡し変数の定義、コマンドおよび特殊な関数の標準的使用方法、パターン・ベリフィケーションの記述形式などを、ファイル仕様書規定、プログラム・コーディング規定などに定めておくことが必要

---

Information for authours. Eijirou Kobayashi  
\*〒583 大阪府羽曳野市はびきの3-7-1



である。

2. 開発過程を体系的なものにすること。

システムの設計、プログラムの設計、仕様書の作成の各フェイズにおいて、標準化された帳票を使用し、手順にそった開発を行う。

3. ソフトウェアの可視性を向上させること。

単純明快に重点をおいたプログラミングを行う。

MUMPSで動くプログラムをつくることは簡単であるが、他人がみてわかるプログラムを書くことは難しいし、価値がある。

プログラム・コーディング規定においてコーディング形式の標準化を定める。

4. 保守を念頭においたシステムの設計およびプログラミングを行うこと。

最初にきちんとした仕事を行う。これは標準化された手順、手法を守ることによって実現する。標準化はMUMPSの自由度を制限するものではなく、むしろMUMPSのコーディングの容易さ、保守性のよさをひきだすものであることを認識することが重要である。

5. トップダウン・ボトムアップ設計をおこなうこと。

6. 構造化プログラミングの手法を導入すること。

システム設計、プログラム設計においては、HIPO手法を利用する。

ジョブ構成図(図式目次)、HIPO総括ダイアグラム(総括ダイアグラム、詳細ダイアグラム)を使用する。プログラムの構造化・モジュール化については、プログラム・コーディング規定に定める。

7. ファイルを標準化すること。

ファイル名の付け方を標準化する。ファイル設計については、ファイル設計用紙を使用し、ノードの定義、レコード形式の規定を行う。

8. 変数名を統一すること。

メタデータの管理について規定をもうける。

変数名、プログラム名の付けかたの標準化を行う。

MUMPSにおいては、共通変数の管理が特に重要である。

9. 文書を標準化し、これを使用すること。

開発・保守の統制、承認のオーソライズを行い、作業者に対する指示、方針を明確に提示する。

10. プログラム・ライブラリを作成し、使用すること。

ユーティリティを作成し、使用することは重複したプログラムの開発を避け、またプログラムの管理を容易にする。

11. 自動化されたツールを作成し、使用すること。

標準化の有効な手段である。

仕様書作成、変数表作成、JOB解析ツールなどを使用し効率的運用を行う。

## 羽曳野病院ソフトウェア開発規程

羽曳野病院ソフトウェア開発規程の各項目を表にまとめて示す。

システムの規定	プログラムの構造 プログラム名のつけ方 プログラム・コーディング要領 スピードの出るプログラム例
システムの定義 U C I の定義と運用管理 システム・テーブルの運用 システム・オープン <small>の</small> 運用管理 アップデートの運用管理 共通メニュー 共通メニューの運用 共通メニューにおけるチェック および受け渡し変数 DDPの使用 コンピュータ間ネットワークの管理 ユーティリティ（%ZDDP）の仕様	仕様書作成規定
ファイル仕様規定	プログラム開発工程の手順と そのドキュメンテーション システム基本設計： サブシステム構成図、ジョブ条件書、 画面・帳票設計用紙 ファイル仕様書、共通変数表 プログラム設計 ジョブ構成図、Hi p o 総括ダイヤグ ラム 補足資料（フローチャート、 ディシジョン・テーブル） 変数表 その他
ファイル名の定義 オンライン・ファイル、バッチ・ ファイル テーブル、カウンタ・ファイル、 ワーク・ファイル キーの表し方 デリミタ キーのソート データ形式の表記法 ファイル仕様書フォーマット L o c K の定義と管理	取扱説明書作成規定
プログラム・コーディング規定	操作手順説明書 入力画面マンマシン 入力手順説明書
用語の定義 システム、サブシステム、ジョブ プログラム、ルーチン、セグメント	入力設計標準
	画面エリアの定義 表示フォーマット 入力フォーマット 画面制御コマンド 画面制御とチェック
	出力設計標準

タイトルの印字 印字コントロール 漢字の印字標準
開発規定資料
ファイル仕様書例 プログラム仕様書例 プログラム・コーディング例 ユーティリティ 仕様書作成ツール 端末属性一覧 共通変数一覧

### おわりに

羽曳野病院のソフトウェア開発規程の概要を述べたが、もちろんこれらの規定は守られることが前提である。

作成された規定が有効性を発揮するには、

1. 明白なリーダシップ
2. モラルの向上
3. 責任と実績の評価

などが、日常業務の中でチェックされることが必要である。

具体的には、例えば仕様書のないプログラムは排除すること、エゴのないプログラミングを行うことなどの徹底である。

標準化は、MUMPSの自由度を制限するものではなく、むしろMUMPSのよさをひきだすための手段であることを認識することが重要である。

MUMPSは大規模システムの構築に耐えられる言語である。今後この言語を生かすための環境の充実が更に必要と思われる。

# Standardization of Programming Development and Maintenance

Eijiro Kobayashi, Yoichi Ogushi, Hiroshi Noguchi, Masafumi Teramura  
Osaka Prefectural Habikino Hospital\*

Standardization of programming development and maintenance is a critical point to realize the integrity, security, efficiency of a system. There are basic strategies that a system designer and programmer can employ in standardization. This paper concentrates on one aspect of the methods developed by Habikino hospital information system. This includes the following components:

1. standardization in advantage of MUMPS
2. systematization of the development process
3. understandable programming
4. programming in consideration of maintenance
6. top down and bottom up programming
7. standardization of globals
8. standardization of variable names
9. using of standardized documents
10. libraries and utilities
11. software tools and technology

---

Information for authors: Eijiro Kobayashi  
\*3-7-1, Habikino, Habikino City, Osaka 583 Japan



# グローバル設計の標準化 VA FileManの場合

嶋 芳成

岐阜県厚生連久美愛病院\*

Key words : VA FileMan, global, standardization

## はじめに

MUMPSのグローバルファイルは文法的には散在配列の変数であり、機能的には木構造のデータベースである。その特徴は、予め宣言することなく、きわめて自由な構造のデータベースを作ることができることにある。

しかし宣言なしに自由な構造をとり得ることは、正確なデータベース構造はプログラムを解析しなくては分からないという欠点でもある。また、データベース構造が異なれば、個々のファイルに専用の登録・検索・出力プログラムを用意しなくてはならないことにもなる。

一方、MUMPSグローバルの構造を意識的に標準化することは可能である。標準化すれば、グローバル設計の自由度が少なくなり、個々の場合に応じた最適化はし難くなる。しかし、データベースの保守や、汎用的なプログラムの開発も容易となり、アプリケーション間のデータの互換性も高まるといった利点が得られる。

以下に、グローバルの構造を決定する要因を挙げ、次にMUMPS上で開発された汎用データベース管理システムであるVA FileManのデータ構造を紹介する。

## グローバルの構造を決定する要因

グローバルの構造を決定する要因としては、つぎのようなものがある。

1. 主キー、すなわちグローバル変数の添字を何にするか
2. 登録するデータの値は1つか、あるいは複数か
3. どのような索引構造をとるか

これらについて若干の考察を加える。

---

Information for authors. Yoshinari Sima.

\*〒506 岐阜県高山市大新町5丁目68番地(電話:0577-32-1115)

## 主キーの値

主キーというのは、ひとつのファイルの中で、レコードを識別するための最も基本となる情報である。例えば患者ファイルでは患者のIDであり、食品ファイルでは食品コードであろう。グローバル変数の場合は、添字である。

主キーの付け方は次のように分類できる。

1. 患者の名前や生年月日などの意味を持った値をキーとする。
2. ファイルに登録された順番に連番を発生させてこれをキーとする。この場合には、キーにはレコードの識別以外に意味がない。
3. 例えば商品コードのような特別に用意されたキーを用いる。

1.の利点は、キーとなったデータによるレコードへのアクセスが単純になることである。しかし同名のレコードが存在する場合には、これらを識別する構造をキーの中に取り込んでおく必要がある。例えば、

```
^PATIENT("山田 太郎",1) ... 1人目の山田 太郎氏
```

```
^PATIENT("山田 太郎",2) ... 2人目の山田 太郎氏
```

といった構造にする必要がある。また、そのキーに利用したデータが変更されたときには複雑な処理をしなくてはならない。まずそのキーの下に登録されているデータを変更後のキーの位置へ移動しなくてはならない。次に、そのデータファイルを参照しているファイルがあった場合には、そのファイルのデータも修正する必要がある。このことは、複雑にファイルが絡み合った大規模なデータベースでは重大な問題となる。

2.の登録順の連番にする利点は、同名レコードが存在してもデータベース構造が単純になることと、データが変更されてもレコードを移動する必要がないということである。他のファイルから参照されている場合も、参照しているファイルへの影響が最小限になるというメリットがある。しかし、ランダムなデータ検索をするには、索引ファイルを準備し、それを通してアクセスしなくてはならないという欠点を生じる。

3.はもし、商品コードのように、1対1対応して重複しないキーを特別に付けることができ、そのキーで検索することが多い場合には有効な構造である。キーの値は、数値である必要はなく文字列でも良い。ある範囲のレコードをアクセスする場合や、連続したキーを発生する場合には数値キーの方が操作しやすい。

## 複数値データ

患者の氏名、性別、生年月日のように、ひとつのレコードにひとつしかないデータもあるが、患者の病名データのように複数の値をもつものも多い。複数値データを別ファイルにして、リレーショナルデータベースのような構造を作ることもできる。しかしグローバルの特徴を生かして木構造に展開する方法が一般的である。木構造に展開するには次の3種類の方法がある。

1. ひとつのグローバル節に境界子を挟んで連結して格納する

```
^GLOBAL(KEY, ITEM)=DATA1^DATA2^DATA3^...^DATAn
```

例: ^PATIENT(2345, "病名")="胃潰瘍^高血圧症^糖尿病"

2. データ項目を区別する添字の範囲を区切って、別々の節にデータを格納する

```
^GLOBAL(KEY, ITEM*COEFF+n)=DATAn
```

例: ^PATIENT(2345, 1001)="胃潰瘍"

^PATIENT(2345, 1002)="高血圧症"

^PATIENT(2345, 1003)="糖尿病"

3. 下位節を設けて、各データを別々に格納する

```
^GLOBAL(KEY, ITEM, n)=DATAn
```

例: ^PATIENT(2345, "病名", 1)="胃潰瘍"

^PATIENT(2345, "病名", 2)="高血圧症"

^PATIENT(2345, "病名", 3)="糖尿病"

1. はグローバル構造は単純になるが、1つのデータが255文字(バイト)以内という制限があり、オーバーフローした場合の対策が必要となる。

2. は添字が深くない代わりに、設計上も、実際のデータ操作上も添字の範囲の管理が必要である。

3. は添字が深くなるが、管理上は最も簡単な構造である。

構造を複雑にする要因として、複数個登録される項目のその中がいくつかのデータに分けられることがある。例えば、ある患者に複数の病名が登録され、その中には病名自身他に、発症年月日、治癒年月日などを持つ場合である。

上記の1.の構造を拡張すると、

```
^PATIENT(2345, "病名")="胃潰瘍;870601;870910^高血圧症;870601;^..."
```

という形になる。データが長くなるので、オーバーフロー対策は是非必要である。2.や3.の構造では、例えば

```
^PATIENT(2345, "病名", 1)="胃潰瘍^870601^870910"
```

```
^PATIENT(2345, "病名", 2)="高血圧症^870601^"
```

という形に拡張できるので、より柔軟であるといえる。

もし、3.の形でさらに

```
^PATIENT(2345, "病名", 1, "名称")="胃潰瘍"
```

```
^PATIENT(2345, "病名", 1, "開始")=870601
```

```
^PATIENT(2345, "病名", 1, "終了")=870910
```

という形にすれば、各データ毎の長ささえ制限範囲内なら、データの個数やデータ項目数についてはオーバーフローを全く考慮する必要がなくなる。



さらに複雑な例として、ある種の疾患は再発、寛解を繰り返す。すなわち複数ある病名の内の、再発年月日はそれぞれの病名毎に複数存在する可能性があるということである。これをグローバルに登録するにも上記のような構造が考えられ得る。

1. ひとつのグローバル節に連結する方法

```
^PATIENT(2345,"病名",5,"再発")="800520^820601^841011^..."
```

2. 添字を別にして登録する方法

```
^PATIENT(2345,"病名",5,"再発",1,"日付")="800520"
```

```
^PATIENT(2345,"病名",5,"再発",2,"日付")="820601"
```

```
^PATIENT(2345,"病名",5,"再発",2,"日付")="841011"
```

この場合でも、1.ではオーバーフロー対策が必要であり、2.の方が自由度が大きい。

### 索引構造

あるデータからそのデータに関連するレコードのキーを得たり、あるデータでレコードを並べ替えるには、索引ファイルを作るのが一般的である。

この索引ファイルの構造にもいくつかの形が考えられる。しかし形式の単純さ、索引作成、更新、削除の容易さから考えて、次の形が最もよいと考えられる。

```
^INDEX(DATA,KEY)=""
```

この索引を作成するには、次のSET命令だけでよい。

```
SET ^INDEX(DATA,KEY)=""
```

削除するには、

```
KILL ^INDEX(DATA,KEY)
```

更新は、

```
KILL ^INDEX(OLDDATA,KEY) SET ^INDEX(NEWDATA,KEY)=""
```

で済む。また、グローバル変数の添字に圧縮機能があれば、似たデータが増えるほど索引グローバルの効率が良くなるというメリットも期待できる。

他にも、

```
^INDEX(DATA)=KEY1^KEY2^KEY3^...^KEYn
```

```
^INDEX(DATA,n)=KEYn
```

といった索引もあり得るが、オーバーフロー対策が必要なこと、登録、削除、更新などが複雑になることなどの欠点がある。

VA FileMan は米国の Veterans Administration で、George Timson らによって開発されたデータベース管理システムである。このシステムはそれだけで閉じたシステムとして用いることもでき、他のアプリケーションからそのモジュールを呼び出してユーティリティとして用いることもできる。

このシステムは、かなりオーバーヘッドが大きく、実行速度が遅いので実用的ではないという批判がある。しかし、用途を誤らなければ十分実用的なプログラムパッケージである。例えば、実際のデータ処理は専用開発したプログラムを用い、データベース構造の管理にのみこのパッケージを使うことができる。あるいは各種のテーブルのような、更新が頻繁ではないファイルのメンテナンスに用いることも悪くはない。これによってテーブルメンテナンス用のプログラム開発を省略できるからである。さらに VA FileMan はメニュー起動型で扱いやすいので、ノンプログラマでもアプリケーションのプロトタイプ開発に利用できよう。

いずれにせよ、VA FileMan のデータ構造に合わせてプログラム開発すれば、いつでも FileMan をユーティリティとして利用できる上、データ構造の標準化という点でも有利である。FileMan のデータ構造は、後述するような構造で、添字が深くなってしまいが、柔軟な形に正規化されているので、グローバル設計の参考にもなろう。FileMan のファイル定義のリストの例を図 1 に挙げる。

以下に FileMan のグローバル構造と、データ問い合わせ、データ編集のモジュールの呼び出し方について概略を述べる。

### 登録番号

VA FileMan のレコードの主キーは常に 0 より大きい数値である。基本的には登録順の連番であるが、数値に意味を持たせて、不連続な値にすることも、また小数を用いることも可能である。登録番号以外のデータでレコードを参照するには、原則として索引（後述）を経由することになっている。

次に示すのは、コンパクトディスクのファイルの例である。

```
^CD(1,0)="Jungle Jane Tour Live"
^CD(2,0)="カマクラ"
^CD(5,0)="モナリザ"
^CD(23,0)="わがまま"
```

↑

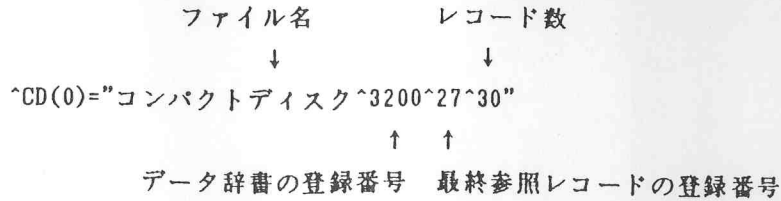
登録番号

### 登録番号の管理節

レコードの登録番号のレベルの添字が 0 である節には、そのレベルに登録されているレコードの数、最後に参照された添字の値、およびそのファイルについて定義しているデータ辞書の登録番号が入っている。

データ 要素	名前	グローバル の位置	説明
データ辞書 #3000 -- 腫瘍患者 ファイル 久美愛科院 1987年9月30日 ページ 1			
格納しているグローバル ^TUMOR( (エントリー 1 件)			
索引領域: 患者氏名(漢字)(B), 患者氏名(カナ)(C) DD アクセスコード: @ RD アクセスコード: @ WR アクセスコード: @ DEL アクセスコード: @ LAYGO アクセスコード: @ 作成 1987年9月15日			
3000,.01	患者氏名(漢字) 文字数 2-30 文字以下で回答してください (必須データ)	0;1	自由文章 入力変形: K:\$L(X)>30!(\$L(X)<2)!'(X?1全.全1" "1全.全) X 削除権: クロス参照: 3000^B 1)= S ^TUMOR("B", \$E(X,1,30),DA)="" 2)= K ^TUMOR("B", \$E(X,1,30),DA)
3000,.02	患者氏名(カナ) 文字数 3-30 文字以下で回答してください (必須データ)	0;5	自由文章 入力変形: K:X[""]!(\$A(X)=45) X I \$D(X) K:\$L(X)>30!(\$L(X)<3)!'(X?1カ.カ1" "1カ.カ) X クロス参照: 3000^C 1)= S ^TUMOR("C", \$E(X,1,30),DA)="" 2)= K ^TUMOR("C", \$E(X,1,30),DA)
3000,1	性別	0;2	コードのセット ; 'M' は 男 ; 'F' は 女
3000,2	生年月日	0;3	日付 (必須データ) 入力変形: S %DT="EX" D ^%DT S X=Y K:Y<1 X
3000,2.5	ID番号 文字数 8 文字以下で回答してください	0;4	自由文章 入力変形: K:\$L(X)>8!(\$L(X)<8)!'(X?8N) X
3000,10	診断名	5;0	3000.01 複数値
3000.01,.01	診断名 診断名 ファイル(#3001)へのポイント (繰返して質問)	0;1	クロス参照: 3000.01^B 1)= S ^TUMOR(DA(1),6,"B", \$E(X,1,30),DA)="" 2)= K ^TUMOR(DA(1),6,"B", \$E(X,1,30),DA)
3000.01,1	左右の別	0;2	コードのセット ; 'R' は 右 ; 'L' は 左 ; 'B' は 両 ; 'N' は なし ; 'U' は 不明
3000.01,2	臓器 臓器名 ファイル(#3002)へのポイント	0;3	
3000.01,3	届け出年月日	1;0	3000.03日付 複数値
3000.03,.01	届け出年月日	0;1	日付 入力変形: S %DT="E" D ^%DT S X=Y K:Y<1 X 最終編集: SBP 22, 1987

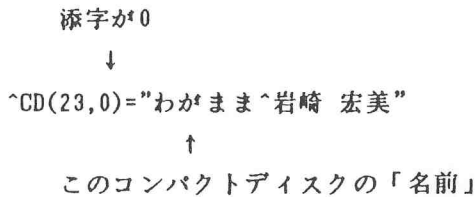
図-1 VA FileMan による  
データ定義リストの例



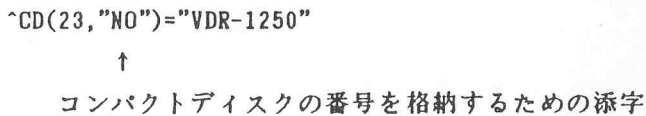
### データを格納する添字

各レコードのデータは登録番号の節には直接格納されない。常にその下の添字に格納される。

すべてのレコードにはレコードを識別する「名前」という項目がある。患者ファイルの患者氏名や、文献ファイルの文献名である。これは、常に添字0の「^」記号を元に\$PIECEで分けた1番目の位置に格納されている。



「名前」以外の項目を格納する添字は任意である。文字列でも構わない。



### 境界子

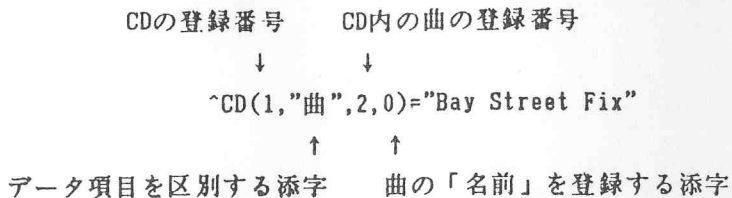
上記のように、ひとつの節に複数のデータを格納するための境界子には「^」記号を用いる。

プログラマモードでは、境界子を使用せずに、固定長のデータを格納するように定義できる。しかしひとつの節に可変長のデータと固定長のデータを混在させることは許されていない。

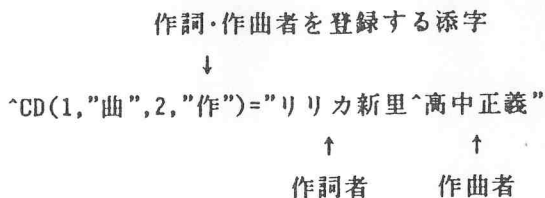
### 複数值領域

患者の病名のようにひとつのレコードに複数のデータを登録することもできる。その場合にはそのデータをまとめて1つの添字が与えられる。そしてその添字の下にそれぞれのデータを区別する登録番号（0より大きい数値）、さらにその下に添字0があり、データの

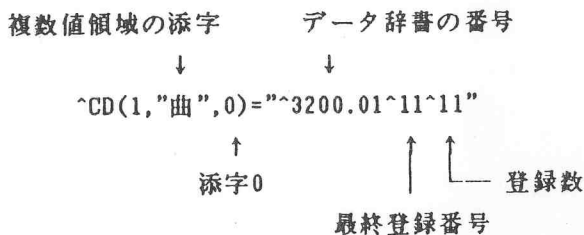
「名前」が格納される。



曲名の他に、各曲の作詞者、作曲者を登録することもできる。



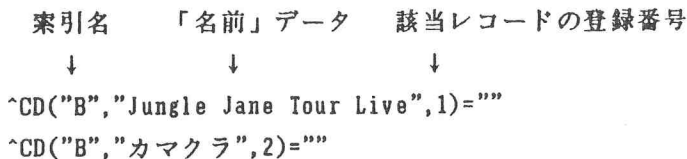
複数値領域を管理するために、登録番号のレベルの添字0の節に、登録されているデータ件数や、データ辞書の登録番号が格納される。



これは、ファイル全体のレコード数を管理する ^CD(0) とほぼ同じ形式になっている。従って複数値領域のことをサブファイルと呼ぶことがある。

### 索引

索引は、メインファイルと同じグローバル内に作られる。レコードの登録番号と同じレベルに索引を区別するアルファベットを置き、次に索引データ、レコードの登録番号を添字とし、索引グローバル節の値は空文字列である。デフォルトで、「名前」によるつぎのような索引が自動的に作られる。



索引はこの他にも任意に、作成することができる。例えばコンパクトディスクの「歌手」で索引したとする。その索引に「C」という名前を付けると、

索引名    索引データ    該当レコードの登録番号

↓           ↓           ↓  
 ^CD("C","サザンオールスターズ",2)=""  
 ^CD("C","高中 正義",1)=""

複数値領域で索引を作る場合には、ファイル全体を索引する場合と、同一レコード内のその複数領域についてのみ索引する場合とがある。

ファイル全体を索引する場合は次の通り。

該当レコードの登録番号    該当レコード内の複数値領域内の登録番号

↓           ↓  
 ^CD("D","Shake it",1,10)=""  
 ^CD("D","ちゅちゅちゅ",5,3)=""

各レコードの複数値領域内だけの索引の場合は次の通りになる。

索引の名前

複数値領域の添字	↓	索引データ		複数値領域での登録番号
レコード番号	↓	↓	↓	↓
	↓			

^CD(23,"曲","B","もういちど",4)=""  
 ^CD(23,"曲","B","カサノバL",8)=""

FileMan の問い合わせモジュールの呼び出し

以上のような構造のグローバル変数を定義しておくこと、FileMan のレコードの問い合わせ、新規登録モジュール(^DIC)を呼び出すことができる。呼び出し手順は、次の通りである。

K DIE,DA,DR	←	パラメータのリセット
S DIE="^CD("	←	目的ファイル名
S DA=23	←	編集したいレコードの登録番号
S DR="2,3,4"	←	編集したいデータ項目の番号のリスト
D ^DIE	←	編集モジュールの呼び出し

DIC を呼び出して新規登録すると、名前による索引("B"索引)も自動的に更新される。

FileManのデータ編集モジュールの呼び出し

登録されたレコードについて、指定されたデータ項目の編集をするモジュールも提供されている。このモジュールの呼び出し方法は次の通りである。

K DIC	← パラメータのリセット
S DIC="^CD("	← 目的ファイル名
S DIC(0)="AEL"	← キーボード入力、エコーバック、新規登録許可
D ^DIC	← FileMan の問い合わせモジュール呼び出し
S NO="+Y	← 登録番号
S NAME=\$P(Y,"^",2)	← レコードの「名前」

このモジュールも、定義されている索引を自動的に生成、更新、削除してくれる。

これらのモジュールを組み合わせることにより、手軽に専用のデータ入力編集プログラムが作成可能である。ここでは、説明しなかったが、FileManにはあるデータが入力されると、それを引金にして別のデータを更新するといった機能もある。

このような機能を利用すると、かなり複雑なアプリケーションに対応可能である。

## 文 献

- 1). VA FILEMAN Programmer's Manual, Version 17, March 1986, MUMPS Users' Group
- 2). File Manager Module Programmer's Guide, Version 16.3, May 13, 1985, Information Mapping, Inc., Waltham, Mass, U.S.A.

# MUMPSを考えるにあたっての要件\*

馬場 謙介\*\*

産業医科大学第二病理学教室\*\*\*

Key word: Future view, 5th generation

著者は、過去と現在の MUMPS の根底に流れる MUMPS の基本精神にたち戻り、次世代に MUMPS を正統に継承しようとする若い技術者に伝えておくべき要件を4項目にまとめた。即ち、「言語仕様の拡張に先立つ小利用技術の規格化の検討に基づき濃縮・洗練した言語仕様の拡張を心掛けること」、「聡明な発想に基づいて均衡のとれた言語仕様を心掛けること」、「機に敏で MUMPS 理念に根ざし、周辺技術の進歩に敏速に対応すること」、「国際感覚で内外の懸案にあたること」の四点である。

## はじめに

ここで、将来のMUMPSが備えるべき新しい機能や仕様を提言するつもりはない。それらは、純粹に技術者の英知に負うところであり、著者はそれを語るにふさわしくないからである。著者はここで、MUMPS の基本精神にたち戻り、その姿を、正統な若い技術に伝えておきたいのである。私見をできるだけ排除する目的で、著者は、過去と現在のMUMPSを改めて見直し、その中から、将来のMUMPSの姿と将来のMUMPSに対するMUMPS人の取るべき姿を模索する方法をとった。この方法で示唆された次世代のMUMPSを考えるにあたっての正統であろう要件を書き残す。

## 利用技術の規格化部品が言語仕様を助ける

過去のMUMPSの最大の事業は、言語仕様の標準化の事業であろう。電子計算機利用技術の歴史の中で、特筆すべきことである。しかし、MUMPS 人は、一つだけ思い違いをしていた。それは、「言語が標準化されていれば、一度何処かで開発されたプログラムがどの装備でも使える」と考えたことである。実際にはどうであったか。答は否

---

\* Requisites to Design MUMPS for the Next Generation

この論文の一部は、第14回日本 MUMPS 学会(1987.7.29-31, 北九州)で講演した。

\*\* Kensuke Baba

\*\*\* 〒807 北九州市八幡西区医生が丘 1-1 (電話: 093-603-1611 (2323), FAX: 093-603-8518)



である。プログラムと云う単位、システムと云う単位は、大きすぎて、環境がおいそれと受け入れるとは限らないことを、今の我々は痛い程知っている。また、たとえ装備が受け入れても、相手先のニーズは多様で、この大きな単位をそのまま適用できる相手先は、まず望めないことも知っている。不確定多数の多様な相手先のニーズに対応できる汎用性を保持しようすると、パッケージは途方もなく膨れ上ってしまい、現実性が無くなるか簡易言語や市販ソフトと大差なくなって、言語であることの存在価値が脅かされる。

このジレンマから抜け出すにはどうすればよいか。答は簡単である。単位を小さくすることである。単位を小さくすると、汎用性能は飛躍的に向上するからである。具体的で実用的な単位として、現行 MUMPS で実現できることを前提に考案されたのが、スモールコンセプトである。スモールコンセプトは、ユーティリティとはその発想からして違い、むしろライブラリープログラムの発想に近いことは、既に機会ある毎に説明してきた。スモールコンセプト<sup>1-2)</sup>の外部仕様とその機能が規格化され、各装備者がこの規格に沿って、スモールコンセプトを提供してくれる状況になると、新しい形の「資源の相互利用」道が開けることも既に力説してきた。この発想の最も根幹を成す思想は、外部仕様が規格化された云わば'純正部品'で、装備間の互換性を維持しようとするところにある。逆に云うと、内部に装備固有の言語性能(Z-命令、\$Z-関数、VIEW 命令、\$VIEW 関数)を使っていながら、外部仕様が規格化されているので、標準的に移送ができることである。MUG のシニアメンバーは、このことをとくと銘記すべきである。

蛇足ながら\$\$関数について論及する。確かに、\$\$関数の提案は、'部品'を関数として持てることは魅力的であり、スモールコンセプトの普及にとっても効果がある。しかし、肝腎な互換性の観点からみると、\$\$関数それ自体は必須の条件ではない。必須なのは、しっかりした小さい利用技術の開発の蓄積と、規格化の努力である。\$\$関数が制定されても、中身が貧弱であったり、'部品'が規格に合っていないければ、'部品'による互換はあり得ないか意味が薄くなってしまう。

以上「言語の標準化のみに固着せず、MUMPS 人が、MUMPS利用技術に関心を持ち続け、その共同利用のための規格の制定と開発・普及に、意欲を持ち続けることが、MUMPSの発展を約束してくれる最も確実な道であり、MUG の真の存在価値がここにある」ことを述べたが、これは、著者が伝えたい真意の前奏曲にすぎない。著者が伝えたい真意は、こうすることによって、MUMPS 言語自身が、言語本来の機能としては無駄な機能を捨てざる効果があることを強調したいのである。このことは、次世代のMUMPS を考えるに先立ち、誰もが銘記すべき必須の要件である。

### 聡明な発想に基づき首尾一貫した言語体系の追及

MUMPSは、データ構造の記述に革命をもたらした。それを懐古してみよう。MUMPSは、その創設期にファイルの概念を捨てて、階層構造でデータ(レコード)を持つことにした。この発想を言語的には、グローバル変数と \$ORDER 関数のペアで実現した。また、レコード(文字列データ)に可変長を許すだけでなく、レコードの下位構造であるアイテムとの間に階層構造の関係を与えた。この発想を言語的には、\$PIECE関数の導入とデリミッタの概念の導入のペアで実現した。MUMPS は、ここまで、首尾一貫して階層構造に執着してきた。この聡明で大胆な発想がMUMPS を特徴づけ、MUMPS の思想の基本理念を成していると云って過言ではない。アイテムの下位の構造には、多岐階層構造を採用しなかったが、「文字列-数値-整数値-真偽値」なる'包含的分類構造'を採用することによって、演算操作を簡便にすることに成功した。言語的には、この直列階層構造を「値の解釈」なる概念の導入と「文字列演算子」なる概念の導入ペアでスマートに実現した。ところが、(その下位構造である'文字'となると、MUMPSは、種々の文字列関数、文字列演算子を用意したものの)、文字の構造には、何の策もこうしなかった。言語的に、16進記述を排除したために、「A」と「a」のビットシフトの単純な関係さえ保持できなくなっていることは、文字～文字以下のデータ構造となるとMUMPS の発想は貧困を極めていることを象徴していると云える。現にこのしわ寄せが\$TR関数に現れたと見ることもできる。

最近までは、文字に対する配慮を欠いていてもそれで結構通用した。ところが、今日では、創設期には扱えなかった色や変形文字は当時のディスクより身近になり、更に画像情報処理も至近距離に位置するようになって、「文字の構造化」の問題の言語的解決は急務の課題となっている。「文字の構造化」とは、文字にもデータを持たせることである。この場合、文字列の中の個々の文字が持つデータは、1文字のデータとは限らない。文字に色や変形文字のための属性を持たせようとする発想は、文字にデータを持たせる発想の最もシンプルな場合にあたる。それに対し、文字にデータを持たせる発想は、データをマトリックス単位で扱う発想であるから、画像処理をも包含した大きな飛躍を約束してくれる。単に色や変形文字のために属性を持たせるなどの姑息的方法に翻弄されず、MUMPS の本来の思想を継承して文字にも(論理的に)階層構造を持たせる方向での言語体系の思い切った拡張<sup>2)</sup>を、次世代の MUMPS を担う技術者の英知と勇気に待望する。この英知によって、所謂「2バイト文字の問題」も奇麗に吸収できてしまであらう。2バイト文字の問題を文字セットの問題や印字の問題として取り組むのではなく、「データ構造はいかにあるべきかの命題」として受け止めてもらいたい。

「データ構造はいかにあるべきかの命題」に解答を与えるだけでは、不十分である。文法体系への波及効果も同時に配慮されていなければならない。波及効果の一つは、上記の「文字列-数値-整数値-真偽値」に及ぶこと明白である。恐らく、データ構造は、「文字行列-文字列-数値-整数値-真偽値」の形で捉えられることになるだろう。

MUMPSのデータ構造の革命を別の視点から眺めよう。グローバル変数による階層構造データの記述は、画期的な‘発明’である。もしかしたら、木構造でデータを持つと云う‘発明’よりすばらしい‘発明’であるかもしれない。であるのに、MUMPS人の多くは、木構造を採用したことが、すばらしいことであると錯覚するか、逆に網構造になっていないことを嘆き、ビューが一方であることにグチさえこぼしている。網構造を装備しないのは、装備者が怠けているからではない。グローバル変数のような、言語の中で一般性をもって使える記述方法が、発明されていないことの方が本質的な問題なのである。網構造のデータを、一般的で、整合性があり、記述の中で実用的に簡便に使える記述方法の‘発明’が次世代のMUMPSに求められている真の課題ではなからうか。

思想の貧困は恐い。データの構造に限らず、MUMPSの基本思想を継承して、新しい思想を形成する正統で妥当で大胆な思索こそ、MUMPSの確かな発展を約束してくれる。MUMPSの思想を継承した大胆で周到な発想の転換を第二の要件として提示する。

#### 機に敏で理念に根ざした勇氣ある行動

次に、MUMPS言語の環境への対応を眺めてみる。マウス、マルチウィンドー、アイコンに対する現行MUMPSのしらけた対応は、それらが一時のはやりで、言語とは無関係であるからであろうか。それとも、そもそも、MUMPSの「S」が‘System’であることを忘れてしまったからであろうか。これらは新しい課題であるからまだしもとしても、現行MUMPSは、折角の\$PIECE関数によって実用性の高い可変長で構造化できる文字列変数を持ちながら、画面制御対策の貧弱さのために、可変長階層構造を意識したコンパクトなデータエントリー<sup>4.5</sup>さえ難しいのである。LINKED CELLなど改善の方向に向かう動きが第14回日本MUMPS学会で見られたことは、注目に値する。

しかし、信じられない程、恐るべきことが、根強く現行のMUMPSに残っていることも忘れてはならない。CRT画面では、WRITE #, \$X, \$Yが、装備によってまちまちであることに、無関心であってはならない。このことを問いかけると、‘文法学者’のいわく、「それはプリンターやTTY型コンソールの規定であって、CRTでは‘望ましい’となっている」である。のっけから是認しているのである。思想の貧困は、やむをえないとしても、この言い訳と怠惰は許してはなりませんまい。

MUMPS の現行標準規定のように、CRT に対して責任を回避するが如きは、これからはあってはならない。MUMPS は、過去のしがらみを脱ぎ去り、しかし、しっかりと、MUMPS の基本理念を踏まえ、周辺技術の動向を敏に真摯に受け止め、迅速且つ適切に、勇気をもって対処する知恵と勇気が次世代を支えるであろう。MUMPS の基本理念を踏まえて、いち早く周辺技術を吸収する態度を第三の要件として提示する。

### 国際感覚で内外の対処する

編集者の一方的な都合でMUMPSニュースに掲載されなかったが、 $\$ \$$ 関数と等価の関数( $\$USER$ 関数)の試案が、1982年に投稿されている。米国で $\$ \$$ 関数が話題になるか前のことである。この他にも日本の発案が国内で埋もれてしまった例は少なくない。著者は、日本人の知恵を高く評価している。自身を持って、外国に向かって、正統に発言することを望む。雑誌Mumpsなど、そのためのお膳立てができています。

一方反省すべきこともある。日本MUGが持つ2バイト系文字に対する知識と経験は外国のそれを、確かに凌駕している。しかし、変形文字などに関する知識と経験は、決して充分ではないことに思いをいたさなければならない。2バイト系文字の問題と変形文字の問題はしばしば混同されて、この識別の貧困の故に、議論が結実しない場面にしばしば遭遇するのが、認識の深いはずのわが国での現状である。外国が、2バイト系文字に慣れている日本を評価して、日本に変形文字についても期待していると伝えられている。変形文字についても期待してくれることは有難い。しかし、これを伝えられているようにまともに受け止めてはならない。何故なら、もともと両者は別の問題であるからである。外国の真意は、「(基本文字情報と属性情報から成る2バイト体系もあり得るので)、この体系と日本の2バイト体系と不整合があってはならないので、日本に伺いを立てている」と考えるのが妥当である。この観点で、冷静に2バイト系文字の問題に対処してもらいたい。

国際感覚を持つことは、国内問題の処理も早めることになる。日本MUGは、日本語処理用の言語要素を(米国から見て外国の仕様に米国仕様がリザーブしてくれている文字である)「 $\{$ 」,「 $\{$ 」,「 $\{$ 」)を使って定義して、了承を得る形で提案するのが筋である。これらの文字の日本語標準MUMPSでの使用は、日本の裁量に委ねられている案件であるからである。従って、例えば、日本語標準  $\$ZPOSITION$ 関数 を止めて、 $\$iPOSITION$ 関数(または、 $\$iiPOSITION$ 関数)として名実ともに日本語標準MUMPSに組み込むべきである。そうすれば、米国の承認を待たずに、( $\$iZPOSITION$ 関数の右詰め拡張など)日本の裁量でどんどん改良できる。

2バイト系文字の処理の国内体系がスマートにできれば、2バイト系文字をかかえる他国にも模範になり、他国も 大筋でこれに従うことであろう。云うまでもなく、2バイト系文字に関する日本の思索は深い。経験も積んでいる。このことを世界に向かって自負するとともに、 $\$iZPOSITION$ 関数の右詰め拡張などの未解決の問題を抱えていることも謙虚に認めて、世界から信用される国際MUMPS人になって欲しい。

最後に、米国MDCの受け入れ体制にかかわらず、日本が世界に向かって発言する場合、「現行言語に磨きを掛ける提案」と「次世代に向かっての提案」を区別<sup>2)</sup>して発言すべきであろうことを付け加える。日本での議論でも、この二つは、明確に区別すべきが当然である。例えば、前述の2バイト系文字の問題は、「現行言語に磨きを掛ける」問題として、しかも内政問題として議論すべきである。変形文字の問題は、限りなく「次世代に向かって」の問題に限りなく近い問題として、区別して議論するか、スモールコンセプトで解決できる問題として議論すべきである。更に、前者が内政問題であることも外国に明確し、且つ国内でも、内政問題と自覚することを忘れてはならない。

前述の文字に下位構造を持たせる提案を含め、世界に向かって、もの申すマンブス人に育って欲しいと同時に、国内問題も国際感覚で対処することが、次世代のMUMPSを正しく方向づけることをであろう。国際感覚で対処することを第四の要件として提示する。

### む す び

以上の四点を、次世代のMUMPSを考えるにあたっての要件として提示した。これを、次世MUMPS人へのメッセージとして受け止め、正統に継承する技術者の多からんことを切望する。正統に継承しようとする技術者は、目先の対応に走らず、MUMPSの基本精神を汲み取ることからはじめて欲しい。そのためには、過去・現在のMUMPSをその目で見返すことを薦める。かくて正統な技術者は、MUMPSの基本精神を身につけ、それを発展させる英知をもち、勇気を持って新しい仕様の確立を断行してくれることと信じる。

### 文 献

- 1) 馬場謙介他 (1984): Small concept. 第11回日本MUG学術大会論文集(大谷元彦編, pp114) 45 - 50
- 2) 馬場謙介他 (1986): 文字列編集のためのスモールコンセプト. *Mumps* 13: 138 - 146
- 3) 藤江 昭 (1987): 次世代のMUMPSを考える. *Mumps* 14 Suppl: 143 - 148.
- 4) 馬場謙介他 (1986): 文字列編集スモールコンセプトと'配列化文字列'を活用したデータの画面入力. *Mumps* 13: 147 - 156
- 5) 馬場謙介他 (1987): Concerning Small Concept. *Mumps* 14, *in press*

# MRPによる生産管理システムの実現

山田 勝 岡部 孝久 佐々木一郎  
株式会社 高岳製作所 システム事業部\*

Key words : Material Requerememts Planning

## は じ め に

当社株式会社高岳製作所は、長年電機メーカーとして主に電力関連機器の製造販売を実施している。また、当社の1工場である小山事業所北工場は、電力関連機器である配変電用柱上変圧器等の中小型変圧器を製造している。

昨今、日本の産業界を取り巻く環境は、高度成長の時代から低成長の時代へ移りかわり非常に厳しい状況下であり、各企業は、不確定な時代を反映し、より柔軟に変化に対応していく体制が必要とされ、当社も決してその例外ではない。また、急速なエレクトロニクスの発展、顧客ニーズの多様化等により電力関連機器といえども製品のライフサイクルは短縮化され、さらに多品種化、高機能化し、生産活動を従来とは、比較にならないほど複雑化している。

このような背景のもとに小山北工場において従来の生産管理システムを見直し、よりダイナミックで順応性に富んだシステムに再構築する必要がでてきた。そこで、生産活動をより円滑でより有機的に実現することを目的とし、最近生産管理の分野で多品種少量生産にも向く同期化システムとして脚光を浴びているMRP (Material Requirements Planning) の考え方をベースにMUMPSを使用し総合生産管理システムを開発構築したのでここに報告する。

## MRPとは

MRPとは、一般に資材所要量計画といわれ、1970年 [アメリカ生産、在庫管理協会] 略称APICS (American Production and Inventory Control Society) の提唱により、急速にその活用が広まった資材計画の1手法である。MRPでは、図1に示すように立案された製品レベルの生産計画を基にして、組立品、部品、原材料などの品目について

---

Information for authours: Masaru Yamada, Takahisa Okabe, Ichirou Sasaki

\*〒452 愛知県西春日井郡西枇杷島町芳野町 (電話: 052-501-4118)

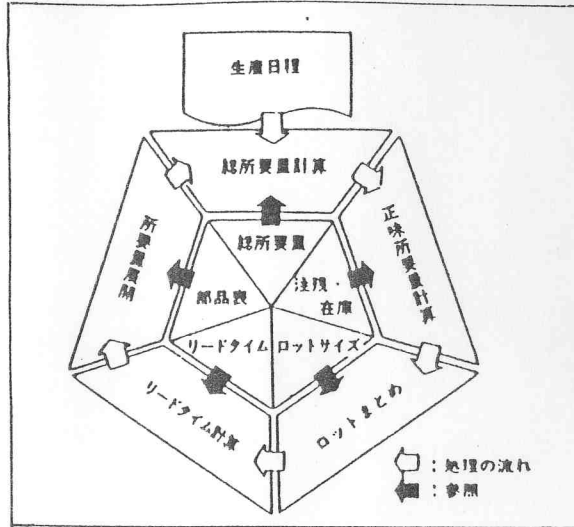


図2 MRPの処理

表1 MRPの計画例

X-A 部品X、Yは、部品Aを1個使い  
 Y-A

部品Aの  
 ・リードタイム 2バケツト  
 ・仕掛率 5%  
 ・発注方式 F/Q=200

	現在量	1	2	3	4	5	6	7	8	9	10
X		100		100		100		100			
Y				100		100		100			
A		100		200		200		200			
割当済		100									
仕掛率		10		10		10		10			
安全数				150							
組込済		210		380		210		210			
OH, OO	160	100									
純所要量				310		210		210			
ロット繰上				400		200		200			
着手日		400		200		200					

JIT (Just In Time) の発想で”必要なものを(品目)、必要なときに(納期)、必要なだけ(所要量)”製造したり、購入したりするための手配計画を実施する。よって、以下の効果があげられる。

- (1) 作業の優先順位が明白になる。
- (2) 在庫(仕掛)がミニマムとなる。
- (3) 納期、設計変更への対処が敏速になる。
- (4) 人員、設備計画に役立つ。

図2にMRPの処理、表1にMRPの計画例を示す。

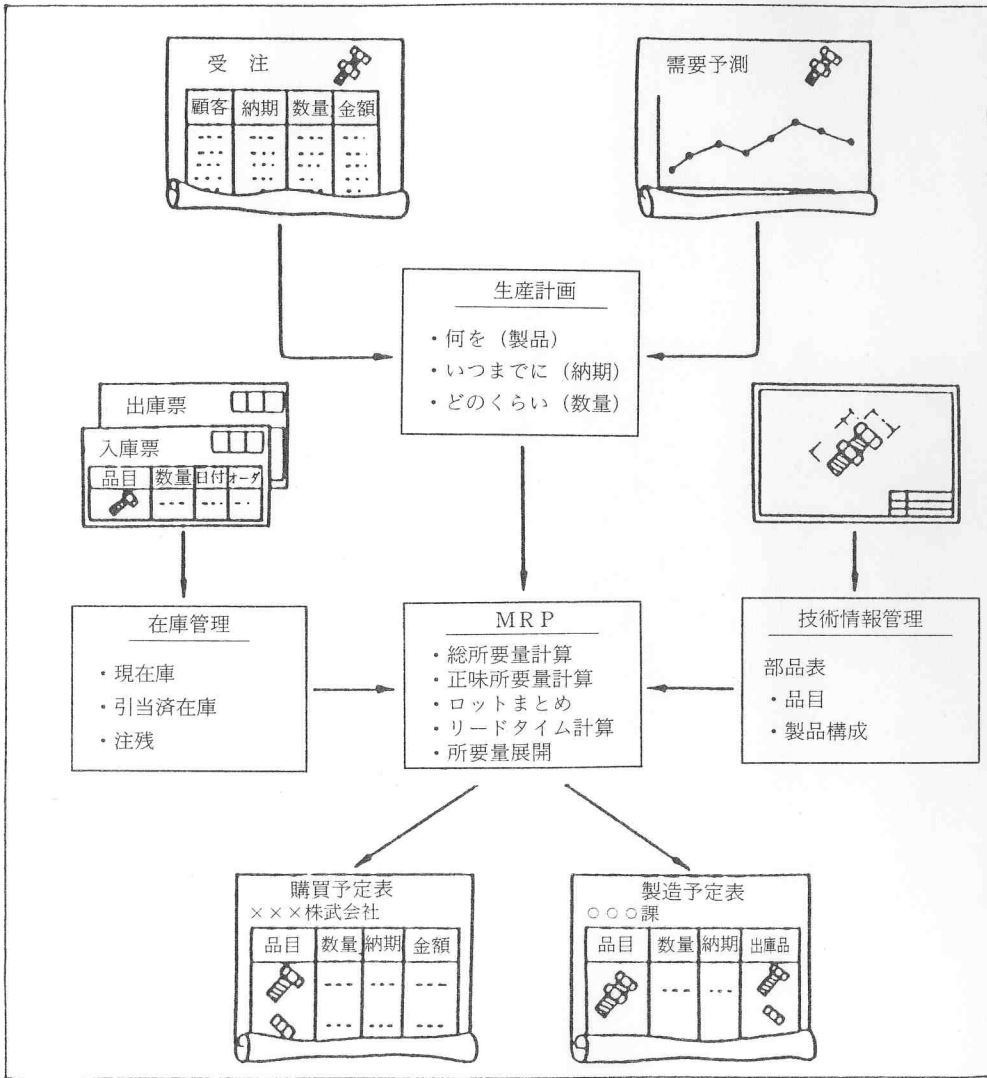


図1 MRPの概要



システムの概要

システム構成

システムは、管理系と実績収集系の2システムに分割されており、ホストマシンとして当社製TAKAC-11/73、Micro TAKAC-11/73を使用しOSは、InterSystems社製M/11+を採用した。両システムは、LAN(Local Area Network - Ethernet)で接続されており相互にデータ転送を実施している。図3にシステム構成を示す。

管理系システムは、生産計画、作業計画および作業差立を実施しており、実績収集系システムは、作業指示された情報をもとに作業進捗、操業実績等の実績収集機能を処理している。管理系システムは、主にパソコンを接続し各業務部署に設置しており、実績収集系システムは、光モデムを用いて工場内に光ケーブルを敷設し、実績収集端末としてハンドヘルドコンピュータを設置し、実績情報をバーコードによりオンライン収集している。

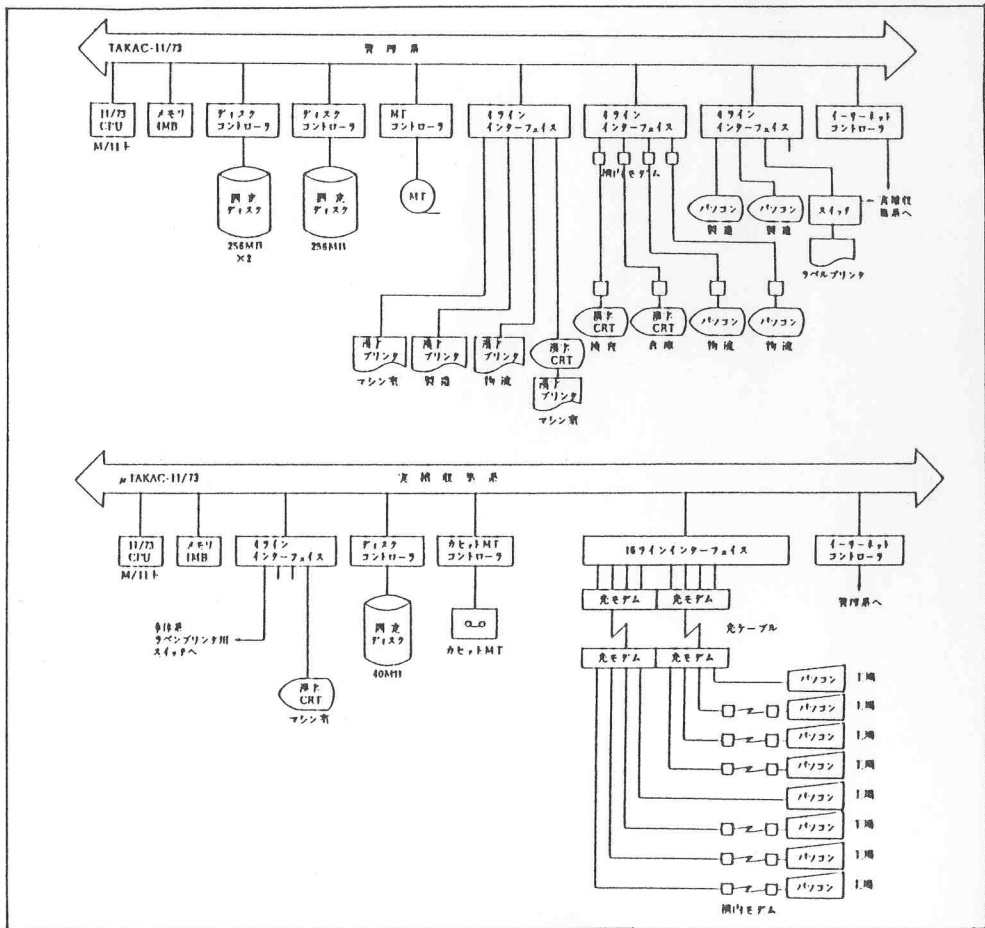


図3 システム構成図

## システムの機能

本システムは、生産計画、負荷計画、在庫管理、購買管理、外注管理、仕掛管理、データベース管理の7サブシステムから構成される。各サブシステムのモジュール関連図を図4に示す。データベース関連図を図5に、各サブシステムの機能一覧を表2に示す。

次に各サブシステムの概要を記し、

生産計画サブシステム：基準生産計画を作成しMRP計算により各部品の実生産計画を作成する。また、資源計画、出図指示等を実施する。

負荷計画サブシステム：生産計画から操業負荷計画を算出し、残業計画、応援計画を立て負荷調整を図る。

在庫管理サブシステム：入出庫管理を実施しリアルタイムで在庫状況を把握する。また、棚卸を実施する。

購買管理サブシステム：生産計画により購入品に関する業者折衝、発注、納期管理を実施する。

外注管理サブシステム：生産計画により外注品に関する業者折衝、発注、納期管理、支給品管理を実施する。

仕掛管理サブシステム：生産計画と工程等の実績収集により作業指示、進捗管理等を実施する。

データベース管理：基礎情報である品目情報、部品構成情報、工程資材情報等を管理し一元化されたデータベースを構成する。

## MUMPSの評価

今回の生産管理システムの開発構築にあたりMUMPSを採用したことにより次の利点を得た。

- (1) 生産管理システムの開発時にありがちな頻繁な仕様変更やデータベースの変更にも柔軟にしかも敏速に対応することができた。
- (2) 生産管理システムの膨大なデータベースを効率良く構成できた。
- (3) プログラマー教育期間が短時間で実施できた。
- (4) プログラミングの開発時間が短時間で実施できた。

次に問題点をあげる。

- (1) プログラムの実行速度が遅い。特にバッチプログラムは問題である。
- (2) ドキュメントの整備状況が悪い。

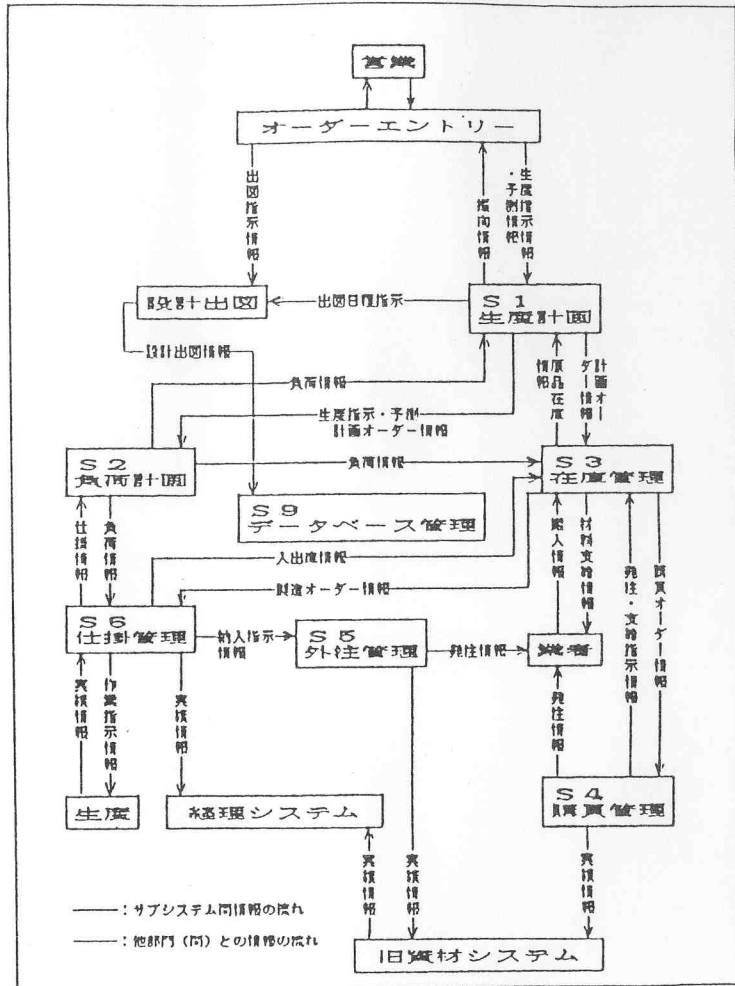


図4 モジュール関連図

おわりに

MUMPS (M/11+) を使用しMRPをベースとした生産管理システムについて述べた。

本システムは、80年7月より第一次稼働に入ったが当初は、データベースの精度の悪さ、ユーザの不慣れによるオペレートミス等のため若干の混乱がみられ、システムとして安定するには数カ月を要した。しかし、それ以後はこれといったトラブルは無く順調に稼働している。

生産管理システムは、常に成長しており、仕様の見直しと維持が継続的に必要である。また、データ精度、ユーザーフレンドシップの維持と向上も重要な課題である。今後もMUMPSを使用し、これらに対処していく予定である。

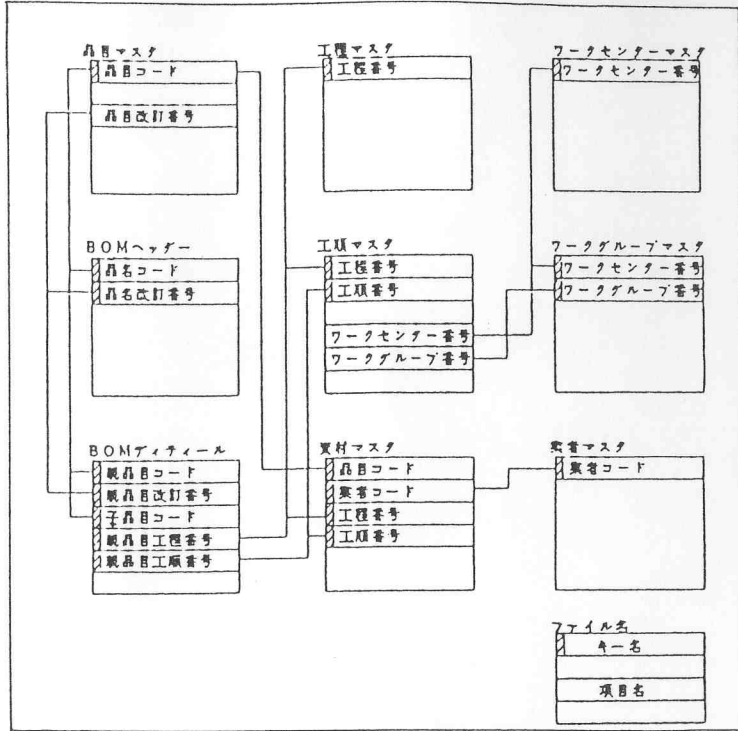


図5 データベース関連図

表2 機能一覧表

サブシステム	コンポーネント	主なプログラム
S 1 生産計画	生産指示管理 生産日程計画 資源所要量計画 所要量計算 緊急処置 出図指示管理	製番登録、生産指示登録 基準生産日程決定、生産計画表 資源所要量計算、有効資源問い合わせ 所要量計算、計画オーダー問い合わせ レベル表示部品構成問い合わせ 出図状況表
S 2 負荷計画	負荷計画	総所要工数計算、W/G別負荷山積 負荷明細表作成、保有工数算出
S 3 在庫管理	オーダーオープン 引当管理 出庫管理 入庫 棚卸	一括オーダーオープン 部品個別引当、引当状況問い合わせ W/C別まとめ出庫指示票 購買・外注発注品入庫、製造出庫 棚卸票、棚卸報告
S 4 納付管理	予告、折衝 発注 納期管理 申込	予告通知 自動発注登録、注文書発行 オーダー状況問い合わせ 申込書発行、申込状況問い合わせ
S 5 外注管理	予告、折衝 発注 納期管理 支給品管理	自動発注登録、発注メンテナンス 外注品納入指示票 有償支給品問い合わせ、有償支給相殺
S 6 仕出管理	日程計画 作業指示 進捗管理 実績収集 採集管理	日程計画メンテナンス 作業指示書 仕出明細問い合わせ、日程計画表作成 採集実績収集、採集実績問い合わせ 採集実績表作成、出来高実績表作成
S 7 データベース 管理	標準備付登録	品目マスター、原価マスター、 工場カレンダー、従業員マスター、 その他各種マスターメンテナンス

文 献

- 1) 高岳レビュー 1987 VOL34 NO1 高岳製作所
- 2) 高岳レビュー 1986 VOL33 NO2 高岳製作所
- 3) オリヴァW. ワイト著 吉谷龍一訳 (MRPによる生産管理)  
日刊工業新聞社
- 4) 平野裕之、上野直紀 (MRP用語500選) 日刊工業新聞社
- 5) 藤本邦明 (アメリカ式かんばんシステム) ダイヤモンド社
- 6) 日本電気情報処理製造・装置システム事業部編  
(実践MRP方式による生産管理システム) 日本能率協会

# The Execution of Production Control by MRP

Masaru Yamada  
Takaoka Electric MFG. Co.\*

The OYAMA North Factory of our company makes various small and medium size transformers. Now we report that in this factory we developed the production control systems by MRP using MUMPS language.

MRP(Material Requirements Planning) is one technique of material planning. The effect of MRP is reduction of items in inventory and in process, flexible management of change of due date and engineering change, and so on.

System organization is that TAKAC-11/73, Micro TAKAC-11/73 (made in TAKAOKA CORPORATION) link by LAN (Ethernet), operating system is M/11+(made in INTERSYSTEMS CORPORATION). System contains seven subsystems which are production planning, load planning, inventory control, purchasing control, subcontract control, process control and database control, serves users the useful information timely in CRT and personal computer.

Using of MUMPS gives us advantage which is flexible management of specification change and database layout change, effective making database and short period of programming development and programmer education.

System runs normally now. But system grows usually and needs continuous update and improvement of data. We suspect that MUMPS is effective for them.

---

Information for authours. Masaru Yamada

\*Yoshino-machi Nishibiwatou-machi Nishikasugai-gun Aichi-Prefecture  
Japan 452



# DSM-11 V3.3J —— 日本語標準MUMPS イ草案の採用に向けて

佐藤 真美

日本デジタル イクイップメント 株式会社\*

Key words : DSM-11 V3.3J, Standardization, Japanese MUMPS, 'I'-draft

## はじめに

8月1日リリースしたDSM-11 V3.3Jでは、新しいハードウェアのサポート、高速データベースインテグリティチェック、複数のJIS型漢字デバイスのサポート等、V3.1Jを更に機能の向上をはかった。また、標準MUMPSイ草案も今回のバージョンよりサポートしている。V3.3Jの機能、イ草案で新たに採用した関数とDSM-11上で既存の関数の使い分けについて、解説する。

## 日本語標準 MUMPS イ草案のサポート

日本語標準MUMPS イ草案で扱う漢字コードは、JIS C6226を採用しているが、DSM-11で扱う漢字コードは、JIS C6226に準拠した。DSM-11で扱っている漢字コードと、JIS漢字コードとは、下記のような単純な変換で対応している。

DSM-11漢字コード(16進)=JIS漢字コード + 8080  
(16進) (16進)

## イ草案サポートモード

DSM-11では、イ草案で掲げられた項目中、DSM-11 V3.3J以前のバージョンと互換性のとれない部分がある。DSM-11と互換性のとれない部分に対しては、DSMの実行モードを切り換えることによって対応している。モードの切り換えは、\$ZCALL関数を使用して行う。

• \$ZCALL (IDON) ... イ草案サポートモードへ

---

Information for authours. Mami Sato

\*〒550 大阪市西区京町堀1丁目3-13 辰巳ビル (電話:06-445-7941)



・ \$ZCALL (IDOFF) ... DSM-11 モードへ

関数値は、\$ZCALL関数を実行する以前のモードを表す。

関数値= 0 ... DSM-11モード

関数値= 1 ... イ草案サポートモード

DSM実行中、現在のモードは

\$ZCALL(IDSTS): 関数値= 0 ... DSM-11モード

関数値= 1 ... イ草案サポートモード

又、LOGIN 時のデフォルトのモードは、DSM-11モード、イ草案サポートモードのどちらにも設定することができる。

#### イ草案とDSM-11との非互換部分

以下の項目については、DSM実行中のモードがイ草案サポートモードになっている場合のみ、サポートされる。

**\$ASCII関数:** イ草案サポートモードでは、全角文字の任意の1つを1文字と数える。DSM-11では、\$ASCII関数の日本語処理は行っていませんのでDSM-11モードでは、全角文字に対しては、半角文字と同様1バイトずつ10進化したものを関数値として返す。又、全角文字列の前後には、漢字シフトインコード(1)とシフトコード(2)がある。

**\$CHAR関数:** イ草案サポートモードでは、\$CHAR関数で全角文字を関数値として得るには、\$ASCII関数と同様に、全角文字の第1バイト、第2バイトをそれぞれ10進化したものをC1,C2とすると、 $C1*256+C2$ を引数として使う。DSM-11では、\$CHAR関数の日本語処理は行わないので、DSM-11モードでは、\$CHAR関数で全角文字を関数値として得るには、漢字シフトインコード124(10進)全角文字の第1バイト(DSM-11漢字コード)、全角文字の第2バイト(DSM-11漢字コード)、漢字シフトアウトコード126(10進)を引数として使う。

**パターンマッチ:** DSM-11では、半角カタカナ文字を調べるには、パターン文字として'K'、全角文字を調べるには'Z'を使う。イ草案モードでは、上記のパターン文字以外に、さらに半角カタカナ文字を調べるには、パターン文字として'k'、全角文字を調べるには、'全'を使うことができる。

**\$JUSTIFY関数:** イ草案モードでは、ASCII文字と全角文字の任意の1つを1文字と数える。DSM-11モードでは、\$JUSTIFY関数の日本語処理は行わないので、全角文字列の前後には、漢字シフトイン、シフトアウトコードがついた内部コードで処理を行っている。

**\$ZPOSITION関数:** イ草案サポートモードでのみ、\$ZPOSITION関数をサポートし、DSM-11 モードでは、指定された表示上の長さに文字列が何文字まで入るかを知る関数としては、従来よりある\$ZPOINT関数をサポートしている。

**\$ZWIDTH 関数:** イ草案サポートでのみ、\$ZWIDTH 関数をサポートし、DSM-11モードでは、文字列の表示幅を知る関数としては従来よりある\$ZLENGTHをサポートしている。

### 新しいハードウェアのサポート

DSM-11 V3. 3Jでは、新たに以下のハードウェアをサポートしている。

**CPU:** Micro PDP-11/53

**ディスク装置:** RX33(1.2MB 5インチフロッピーディスク), RD54(159MB 固定ディスク), RD31(20MB 固定ディスク)

**磁気テープ装置:** TU81E(1600/6250BPIテープドライブ), TQK50(Q-BUS)/TUK50(UNIBUS) (カートリッジテープドライブシステム)

**コミュニケーションデバイス:** DELUA(UNIBUS)(イサernet・コミュニケーション・デバイス, DPV11(Q-BUS)(シンクロナス・コミュニケーション・デバイス(バイシク))

### 高速データベースインテグリティチェック(↑FASTIC)

DSM-11は、マルチウェイ・ツリー構造を用いて、グローバル変数を実現している。これらのマルチウェイ・ツリーの各要素は、DSM-11ボリュームセット内の1024バイトのディスクブロックにストアされている。↑FASTICルーチンは、マルチウェイ・ツリーを形成するブロックを継いでいるポインタが正常かどうかを判断する。また、↑FASTIC は、ブロック内部の整合性をもチェックする。データベース内に矛盾が生じている場合、↑FASTICは↑FIXユーティリティを用いてエラーを修復するために必要となる情報を出力する。↑FASTIC は、20メガバイトのデータを約1分でチェックする。

### JIS系漢字タイプデバイス

漢字DSM-11では、DEC型漢字コード体系の他に、JIS型漢字コード体系を使用するハードウェア(端末、プリンタ等)もサポートしている。DSM-11では、↑JISDEF ユーティリティを用いてJIS 型漢字コード体系のハードウェアに対する漢字および ASCIIキャラクタセットを呼び出すシーケンスを定義するが、DSM-11 V3.1J以前のバージョンでは、

呼び出しシーケンスをコンフィギュレーションごとに1組だけしか定義することができなかった。

DSM-11 V3. 3Jでは、コンフィギュレーションごとに3組の呼び出しシーケンスを定義することができる。また、各シーケンスは、第一バイト目(通常<ESC>キャラクタ)も含めて任意に定義することができる。これにより、システム中に同時に異なる3タイプのJIS型漢字コード体系のハードウェアを接続することができる。これにともない、端末属性設定ユーティリティ(↑MUX)で設定する項目に、JIS型漢字コード体系デバイスの指定をした場合のシーケンス番号の指定が追加された。

### グローバル・リプリケーション

グローバル・リプリケーションとは、UCI内のグローバルへのアクセス(Set, Kill)を自動的に他のUCI に対しても行ない、それにより、グローバルのコピーを作成する機能である。ここでいう、他のUCIは、異なるボリュームセット中に存在するものでも、他のDSM-11システム内に存在するものでも可能です。ただし、他のシステム内のUCI の場合は、DDP ネットワークにより接続されている必要がある。

それぞれのUCI で稼働しているルーチンは、グローバルの読み込みに関しては、マスタのグローバルをアクセスするが、書き込み(Setコマンド)および削除(Kill コマンド)に関しては、そのグローバルの全てのコピーに対してオペレーションする。Lock コマンドについては、そのグローバルのコピー(マスタも含む)の中で "ロック・マスタ"に指定されたものに対してオペレーションする。(注)この機能はオプションである。

### 最 後 に

本稿では、イ草案サポートを中心に、本学会中、時間の関係上説明できなかったDSM-11 V3.3Jの機能面についても述べた。今後とも、DSM-11/VAX DSMは、よりいっそう使いやすいシステムをめざして、機能、能力面等の向上をはかって参る。

# DSM—11 V3.3J

## Standardization of Japanese MUMPS

### '1' — Draft Support

Nami Sato  
Nihon Digital Equipment Corporation\*

DSM11 V3.3J has been released on August 1st. DSM11 V3.3J supports Standardization of Japanese MUMPS '1'draft. In this article, and I will introduce the following DSM11 V3.3J enhancements and the differences between new functions of '1'draft and the functions that have been used from previous versions.

- Standardization of Japanese MUMPS '1'-draft support  
change mode ..... \$ZC(IDON), \$ZC(IDOFF), \$ZC(IDSTS)  
incompatible with ...\$A, \$C, \$J, pattern match, \$ZP, \$ZW
- New Hardware Support
- Fast Data Base Integrity Checker (^FASTIC)
- Multiple Kanji Invoke escape sequences of JIS Kanji  
Device support
- Global Data Base Replication

---

Information for authors. Nami Sato

\*Tatami B.1 1-3-13 Kyoumachi Hori Nishi-ku Osaka-city Japan 550

## 目 次

- 1-6 KIOKU — 実験動物データ解析のエキスパートシステム  
今泉 幸雄、藤本 浩司、小形 稔、金子 泰久、小池 敏
- 7-12 U — MUMPS の機能強化(II)  
小林 勝、煙山 孝、上戸 隆
- 13-19 MUMPS による日本語処理とそれをめぐる諸問題について  
嶋 芳成
- 21-27 千葉大学病院における標準化について  
本多 正幸、里村 洋一
- 29-31 臨床医学を発展させるフィードバックシステム  
野瀬 善明
- 33-43 スモールコンセプト  
馬場 謙介
- 45-51 MUMPS によるトロトラスト被注入者剖検例のデータベース  
石川 雄一、日吉 徹、畠山 茂、田久 浩志、馬場 謙介、森 武三郎
- 53-56 MUMPS における YAC II とデバッグツール  
藤本 浩司、今泉 幸雄
- 57-61 プログラム開発・保守の標準化  
古林 榮次郎、大櫛 陽一、野口 弘、寺村 昌文
- 63-72 グローバル設計の標準化 — VA FileMan の場合  
嶋 芳成
- 73-78 MUMPS を考えるにあたっての要件  
馬場 謙介
- 79-87 MRP による生産管理システムの実現  
山田 勝、岡部 孝久、佐々木 一郎
- 89-93 DSM — 11 V 3.3 J ——— 日本語標準 MUMPS イ草案の採用に向けて  
佐藤 真美

## CONTENTS

- 1 – 6 KIOKU – Expert system of Experimental Animal Data Analysis  
*Y. Imaizumi, K. Fujimoto, M. Ogata, Y. Kaneko, S. Koike*
- 7 – 12 Enhancement of U–MUMPS(II)  
*M. Kobayashi, T. Kemuriyama, T. Kamido*
- 13 – 19 Japanese Standard  
*H. Shima*
- 21 – 27 On Standardization of Hospital Information System in Chiba University Hospital  
*M. Honda, Y. Satomura*
- 29 – 31 A Feedback System Contributing for the Progress of Medicine  
*Y. Nose*
- 33 – 43 Small Concept  
*K. Baba*
- 45 – 51 Computer Database on Thorotrast–administered Patients Using MUMPS System  
*Y. Ishikawa, T. Hiyoshi, S. Hatakeyama, H. Takyu, K. Baba, T. Mori*
- 53 – 56 YACII & Debugging Tools for MUMPS  
*K. Fujimoto, Y. Imaizumi*
- 57 – 61 Standardization of Programming Development and Maintenance  
*E. Kobayashi, Y. Ogushi, H. Noguchi, M. Teramura*
- 63 – 72 VA File Man – Global Standardization  
*H. Shima*
- 73 – 78 Future view of MUMPS  
*K. Baba*
- 79 – 87 The Execution of Production Control by MRP  
*M. Yamada*
- 89 – 93 DSM – 11 V3.3J – Standardization of Japanese MUMPS '1' – Draft Support  
*N. Satou*