

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。

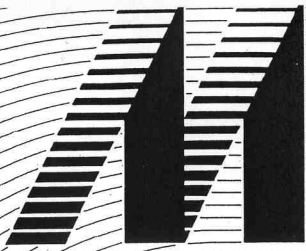
Copy right : M Technology Association - Japan

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



*Technology
Association
Japan*

Mumps

Vol. 20, 1995

Journal of MTA-Japan

真のダウンサイジング!!

一台のパソコンによるマルチユーザー、マルチタスクシステム、これは既に常識ですね!!

DataTree M と DT Network を使えば、1000 台以上のパソコンをネットワークで結び、分散処理、分散データベースシステムを構築することができます。

100 台のデータベースサーバーに、100 台のバックアップサーバー、
合計100 Gbyte 以上のデータベース、4000 台ものクライアント.....
米国で現実に進行中の大プロジェクトです。

もちろんこれらはすべてパソコンです。

◆ パソコン用高性能MUMPS処理系

DataTree M (DTM-PC,DT-MAX)

◆ GUI 環境

Visual M

◆ 対応機種

DOS/V対応パソコン

NEC PC9801 / 9821シリーズ

日本ダイナシステム株式会社

〒460 名古屋市中区新栄2-1-9

雲竜ビル東館504

TEL 052-242-5441

FAX 052-242-5984



InterSystems



目 次

	頁
■ 巻頭言	
J I S化を終えたM言語のこれから	嶋 芳成 1
■ 総 説	
M言語 (MUMPS) のJ I S化	大楠 陽一, 本多 正幸, 藤江 昭, 高橋 隆, 里村 洋一, 若井 一朗, 山本 和子, 河村 徹郎, 木村 一元, 山下 芳範, 嶋 芳成, 小林 泰道, 松本 重雄, 永井 肇, 岩田 稔, 丸尾 誠史, 加山 英男, 内山 誠作 3
The USA MUMPS Activities	John Dumas 13
Background and Technical Features of DTM and DTM-J	Eric K. Pape 17
The Future of M	Robert P. Mappes 23
■ 原 著	
A New Platform for Brigham and Women's Hospital	Robert Beckley 31 Pasha Roberts Michael Monroe Jim Marra Frederick L. Hiltz, Ph.D. P.A. McManus
新開発環境E a s y - T o o lの開発	大楠 陽一, 三澤 哲夫, 古田土義裕, 小西千恵子, 高橋 政志, 平野 健一 37
M言語におけるオブジェクト指向環境	今泉 幸雄 41
■ 技術ノート	
M環境と外部世界との融合の試み	佐藤比呂志 51
■ 第21回 日本エム・テクノロジー学会大会 報告	
第21回日本エム・テクノロジー学会大会を振り返って	岡田 好一 55
■ 日本MUG事務局からのお知らせ	
「日本Mテクノロジー学会」ご入会のご案内	59
「日本Mテクノロジー学会」規約	63
■ 資 料	
投稿規定	69
編集後記	73
表紙装丁	岡田 好一

JIS化を終えたM言語のこれから Future of JIS M Language

日本エム・テクノロジー学会幹事 嶋 芳成

M言語システム(MUMPS)が誕生してから四半世紀が経過しました。その間のコンピュータ技術の進歩は、まさに驚異的なものです。コンピュータの性能はおおよそ2年に2倍になるという経験則があるそうです(二の法則)が、これから単純計算すると、現在50万円のパソコンは、M言語誕生のころの10億円のコンピュータの、さらに3倍近くの性能をもつことになります。

この強大なパワーを利用するためのソフトウェアも、急速な勢いで変化を遂げてきました。MacintoshやMS-WindowsなどのGUI(グラフィカルユーザーインターフェース)も、数年前には賛否両論が雑誌を賑わしていましたが、現在では当たり前になっています。オブジェクト指向という言葉も最近では珍しくもなく、かえって反省期に入っているようにも見えます。

そのような中で、大楠先生を始めとするMTAを中心として諸先生の努力が実り、M言語が念願のJIS化を果たした訳です。これを梃子にして、M言語の一層の普及を図りたいというのは、M言語に関わる多くの関係者の思いであると推察します。

ただし、状況は楽観的ではありません。例えばGUI技術とM言語を融合しなくてはなりません。オープンシステムの掛け声の中で、SQLがデータベース間の共通言語になっています。オブジェクト指向の開発環境や、CASEツールなど開発手法までがセットになって整備された競合製品の中で、M言語の長所だけでは受け入れられにくいのが実情です。当然、雑誌の広告や提灯記事の通りには、現実のシステム開発は進みません。あるユーザーが新システムの開発ツールを選択するために、代表的なRDBとM言語を比較する為のプロトタイプシステムを作成したところ、RDBでは期限内にそれが作成できなかったためにM言語との比較ができなかったとか、あるいは、やはり代表的なRDBを使っていて、5年分、50万人の健診データを扱えると発表されていたシステムが、2年分しかデータ検索できなくて、現場では困っているといった裏話が伝わってきます。

これらの例が単に技術者が未熟なだけなのか、あるいは彼らが選択したツールが課題に合わなかったという問題なのか、それともユーザーの要求仕様が固まらないといったプロジェクト管理運営上の問題なのか、わかりかねます。少なくとも、広告に書いてあるほどには、すぐにでも何でもできるというわけではないようです。

「何故M言語なのか」「他のRDB製品と比較してどうなのか」という議論を重ねていく必要があると思いますし、当然M言語に欠けている点を補っていかなくてはなりません。そして、その作業の成果を広く、社会にアピールする必要があります。残念ながら、このアピールということについては、今までのM言語関係者の不得意とする分野であったように見受けられます。

様々な広告宣伝活動のほか、各種セミナーの開催、教科書/技術書などの充実を図り、集積したノウハウを積極的に公開することで、M言語利用への門戸を広くすることができるはずです。さらに、開発環境や

日本ダイナシステム株式会社

〒460 名古屋市中区新栄2-1-9雲竜ビル東館504

TEL 052-242-5441 FAX 052-242-5984

2 JIS化を終えたM言語のこれから

ユーティリティの整備は、今後ますます重要になります。M言語ユーザーが増加すれば、その分、切磋琢磨し、技術向上へつなげていけることでしょう。さらに世界規模で見ると、日本においてM言語ユーザーやMTA会員が増えることで、M言語の開発事業の中での日本の発言力強化につながり、漢字を含む多国語処理機能の標準化などで有利になります。M言語はユーザーが作っていく言語であることを再確認したいと思います。

日本におけるM言語の普及、その利用技術の向上について、MTA-Jはリーダー的役割を果たすべく、積極的な活動を行わなければなりません。会員一人一人の皆様には、さらなるご協力をお願いしたいと思います。そして、その第一歩としての雑誌MUMPSには、質・量ともに、今後ますますの発展を期待しております。

M言語 (MUMPS) の J I S 化 Japan Industrial Standard of M

大櫛陽一、本多正幸、藤江昭、高橋隆、
里村洋一、若井一朗、山本和子、河村徹郎、
木村一元、山下芳範、嶋芳成、小林泰道、
松本重雄、永井肇、岩田稔、丸尾誠史、
加山英男、内山誠作

Yoichi Ogushi, Masayuki Honda, Akira Fujie, Takashi Takahashi,
Yoichi Satomura, Ichirou Wakai, Kazuko Yamamoto, Tetsuro Kawamura,
Kazumoto Kimura, Yoshinori Yamashita, Yoshinari Shima, Yasumichi Kobayashi,
Shigeo Matsumoto, Hajime Nagai, Minoru Iwata, Masashi Maruo,
Hideo Kayama, Seisaku Uchiyama

長年の課題であったJIS制定が実現した。1969年にアメリカMassachusetts General Hospital(MGH)のDr.G.O.Banettにより発表されて以来、25年が過ぎた。この間のコンピュータテクノロジーの発展は急激であり、多くのコンピュータ言語が発表されてきた。Mは、こうした中で、実用システムとして着実に発達を続けてきた。JIS制定を機会に、今までの報告をも統合し、ISO/IEC制定からJIS制定までの経過を詳細な記録として残し、MUMPS-JIS化の概要を述べる。

(キーワード：日本工業規格、J I S、国際標準規格、I S O、I E C)

Japan Industrial Standard of MUMPS has been established at 1995/02/01. Dr.G.O.Banett of Massachusetts General Hospital(MGH) proposed MUMPS system in 1969. It passed about 25 years. Computer technologies have been developed extremely in this period. Many computer languages have been proposed. M has steadily grown as tools for practical systems. We have summarized the course until the establishment of MUMPS-JIS and described the outline.

1 はじめに

長年の課題であったJIS制定が実現した。1995年2月1日は、日本のMTAにとって記念すべき日となった。日本規格協会から、「日本工業規格 プログラミング言語MUMPS」が、6月中には出版されるとのことであ

MUMPS言語原案作成専門委員会（通商産業省工業技術
院、日本電子工業振興協会）
（社）日本電子工業振興協会 基盤技術部内
〒105 東京都港区芝公園3-5-8 機会振興会館

Special Committee for MUMPS Language JIS-Draft
Department of Basic Technology, Japan Electronic Industry
Development Association
Kikaishinkou-kaikan, 3-5-8 Shiba-kouen, Minato, Tokyo 105,
Japan

4 M言語 (MUMPS) の J I S 化

る。1969年にアメリカMassachusetts General Hospital(MGH)のDr.G.O.Banettにより発表されて以来、25年が過ぎた。この間のコンピュータテクノロジーの発展は急激であり、多くのコンピュータ言語が発表されてきた。Mは、こうした中で、実用システムとして着実に発達を続けてきた。1976年にはロシア語MUMPSが登場し、1977年にはAmerican National Standard Institute(ANSI) 標準言語となり、1979年にはマイクロMUMPSが登場し、1984年にはANSIの第2版が制定され、1986年には米連邦情報処理標準言語 (Federation Information Processing Standard:FIPS) となり、日本でも日本語MUMPS標準 (イ草案) が本学会で認められ、医療情報学会誌にも発表され [1]、1990年にはANSIの第3版が制定された。これが、Joint Technical Committee of the International Organization for Standardization(ISO) and the International Electrotechnical Commission(IEC) JTC1, Information technology に提出され、国際標準規格となった。

JIS制定を機会に、今までの報告をも統括し [2 - 4]、ISO/IEC制定からJIS制定までの経過を詳細な記録として残し、MUMPS-JIS化の概要を述べる。

2 経過

以下に、国際標準の決定から、JIS制定までについて、出来る限り詳細に、時間的経過に沿って述べる。

- 1992.07.06 情報規格調査会SC22専門委員会委員長の中田育男氏 (筑波大学・電子情報工学系) よりISO/IECのJTC1/SC22が1991-11-21から1992-05-21の間に実施したFast-track方式での投票結果において、MUMPSが議決基準を上まわったとの連絡を受けた。
- 1992.08.03 工業技術院に出向き、JIS化に対する依頼を行った。(MTA幹事多数参加)
――― その後、資料の送付や担当官に対して、メーカ (住友電工システムズ) でティーチインを実施した。
- 1992.09.22 中田育男氏よりフィンランドで開かれたJTC1/SC22本会議においてMUMPSが国際標準言語とすることが議決されたとの連絡を受けた。
- 1992.12.25 日本電子工業振興協会に伺い、JIS化の受皿となるべく依頼を行った。(MTA幹事、メーカ、ベンター多数参加)
- 1992.12.末 JTC1から日本工業標準調査会情報部会 (JISC) に正式報告書が届いた。
- 1993.01.19 JIS化に当たり、政府の予算化について、工業技術院担当官と作業方式の調整を行った。
その後、予算の確定の連絡を受けた。
- 1993.02.22 中田育男氏よりMUMPSの国際規格書が届いた。
ISO/IEC 11756 First edition 1992-12-15
INTERNATIONAL STANDARD Information technology-Programming languages- MUMPS
- 1993.04.13 日本電子工業振興協会と作業の詳細についての打合せを行った。
――― 以後、JIS化の作業行程、委員会の構成、作業方式などの調整を行った。その結果、平成5年度から2年間でJIS化することとなり、日本規格協会を經由して、日本電子工業振興協会が受託し、標準化マネージメントの元にある言語標準化調査研究委員会を親委員会とし、MUMPS言語原案作成専門委員会でJIS原案を作成することとなった。また、JIS化作業に関する参考資料をいただいた。
- 1993.04 文部省公文書用語の一太郎 (V4.3) 辞書を作成した。

- 1993.07.15 第一回原案作成専門委員会が開催された。この委員会の委員長は大櫛で、翻訳担当幹事に住友電工の藤江氏、公文書化担当幹事に千葉大学の本多氏をお願いすることになった。その他の委員としては、MTA、メーカ、ベンダー、工業技術院、日本規格協会、日本電子工業振興協会から17名と多数の協力を得ることとなった。
- 1993.08 JIS用語とM言語用語の日本語英語対比表やインデックスを作成するなどの作業効率化の準備も行った。(藤江幹事)
- 1993.10.01 幹事会(東京) 通商産業省工業技術院(工業技術院)と打合せ
- 1993.10.05 93年度第2回MUMPS言語原案作成委員会
 ---- 日本語処理追加
 ---- 日本工業標準調査会情報部規格調整専門委員会(規調委員会)の委員である筧捷彦氏(早稲田大・工)とコンタクト
- 1993.10.19 幹事会(住電:赤坂)
 ---- 一部修正
 ---- ISOからのJIS化希望調査と報告
- 1993.10.28 幹事会(住電:赤坂)
 ---- 一部修正
- 1993.11.24 93年度第3回MUMPS言語原案作成委員会
- 1993.12.07 幹事会(東海大) 工業技術院、日本電子工業振興協会の視察と打合せ
- 1993.12.18 MDCC-Jでの打合せ(名古屋)
- 1993.12.20 幹事会(住電:赤坂)
 ---- 一部修正
 ---- 筧捷彦氏に査読を依頼
- 1994.01.17 93年度第4回MUMPS言語原案作成委員会
 規調委員会委員 筧捷彦氏出席
 ---- 解説、英和对訳、索引追加
- 1994.02.14 93年度第1回言語標準化調査研究委員会(親委員会)
 ---- 一部修正
- 1994.03.04 93年度第2回言語標準化調査研究委員会(親委員会)
 ---- 一部修正
 ---- 年次報告書作成
- 1994.03.31 日本工業規格[案]中間報告:第11版
 日本工業規格[案]プログラム言語MUMPS JIS X 30xx-19xx
- 1994.06.09 94年度第1回MUMPS言語原案作成委員会
 工業技術院OBと打合せ
 ---- 一部修正
- 1994.06.23 幹事会 工業技術院、日本規格協会、日本電子工業振興協会と打合せ
 ---- 一部修正

6 M言語 (MUMPS) の J I S 化

- 1994.06.29 日本工業規格 [案] : 第13版
日本工業規格 [案] プログラム言語MUMPS JIS X 30xx-19xx
通商産業省工業技術院標準部情報規格課へ提出
――― 用語委員会提出用資料作成
- 1994.07.08 規格調整専門委員会 (INSTAC) 用語委員会へ用語申請用紙と理由書を提出
- 1994.08.05~08.07
通商産業省工業技術院標準部情報規格課より、MTA大会視察
- 1994.08.08 規格調整専門委員会 (INSTAC) 用語委員会
――― 付議 (起案) 工業技術院標準部長が作成
――― 大臣付議 通商産業大臣→日本工業標準調査会会長 (08.24)
――― 付託 日本工業標準調査会会長→情報部会会長 (08.24)
- 1994.09.01 日本工業標準調査会情報部会 (JISC) 用語委員会
- 1994.09.19 日本工業標準調査会情報部会 (JISC)
――― 一部修正
――― 上申 情報部会会長→日本工業標準調査会会長
――― 答申 日本工業標準調査会会長→通商産業大臣
――― JIS事前意図公告 標準化ジャーナル: 日本規格協会発行 3ヶ月間
- 1995.01.31 94年度第2回MUMPS言語原案作成委員会
- 1995.02.01 日本工業規格 プログラム言語MUMPS JIS X 3011-1995
- 1995.03.14 94年度第3回MUMPS言語原案作成委員会
- 1995.03.31 日本工業規格 [案] 最終報告

3 J I S の概要

J I Sには、要約J I S (国際規格をそのままJ I Sとし、日本語の要約を付けたもの)、国際一致規格 (内容を変更することなく日本語に翻訳したもの)、国際様式規格 (様式は国際規格に基づくが、内容の一部を日本の現状に合うように追加及び変更したもの)、日本独自規格、とあるが、今回は国際規格にイ草案を追加したため、国際様式規格を採用した。[5]

内容の詳細は、日本工業規格を参照とし[6]、ここでは代表的なMの用語を付録に示した。JIS化に当たっては、JIS用語の基本的な考え方や、他言語との統一性、カタカナ表記を出来る限り避けるなどの方針があり、従来の使いなれた用語から変更になっているものが多い。しかし、M言語のオープン化や今後の発展のためには、必要な関門と思われる。今後の使用で、どうしても定着しない用語については、次期バージョンで検討することが可能である。

4 おわりに

日本の国家規格となったので、これからは大いにアプリケーションの幅を広げ、教科書や解説書の出版、ビジネス用の拡販など、Mの普及に取り組むことが望まれる。

国際標準は、すでに次のバージョンの検討が進んでいる。次のバージョンでは、MS-WindowsやX-WindowなどのG U IをMから直接かつ同じプログラムで動かせ、トランザクション処理とエラー時のデータベースの自動復旧、オープンシステム対応による異機種M間やR D Bシステムとの相互アクセス、グラフィクスや端

末特性の標準化などの最新の機能が包含されている。これにより、C言語やACCESSなどと同等のウィンドウ・アプリケーションの開発が可能となり、COBOLなどと同等の信頼性を持つことができるようになり、総合機能として、他の言語やシステムを抜き出したテクノロジーとなる。また、日本語処理は国際標準規格として当初から包含され、マルチ言語システムとなる。これにともない、規格書は、"Programming Languages - M", "Communication Protocol - Open MUMPS Interconnect", "Computer Graphics - Graphical Kernel System(GS) MUMPS Language Binding", "Programming Languages M Windowing API"の4分冊となっている。そのボリュームも、現在の約4倍となり、最終的な調整に時間を要している。日本でも、国際規格のバージョンアップに伴う体制を準備しておく必要がある。

文献

1. 大櫛陽一、八日市谷隆、馬場謙介、大谷元彦、錦見尚道、山本和子、若井一朗、平手徹、平川顕名、里村洋一、鈴木利明、河村徹郎、石丸徑美、嶋芳成、野上稔、木村一元、前田卓雄、内田達弘、奥野英子、田久浩志、菊池洋二：日本語マンプスの標準化について、医療情報学、6(2)、207-215、1986。
2. 大櫛陽一：ISO通過の正式報告が届きました、新MUMPSニュース、1(1)、1-1、1992。
3. 大櫛陽一：MUMPSの世界標準化決定とそのドラマ、風人倶楽部PRESS、号外、1-1、1992。
4. 大櫛陽一、本多正幸、藤江昭、高橋隆、里村洋一、若井一朗、山本和子、河村徹郎、木村一元、山下芳範、嶋芳成、小林泰道、松本重雄、永井肇、岩田稔、丸尾誠史、加山英男、内山誠作：特別報告 J I S 原案作成について、第21回日本エム・テクノロジー学会大会予稿集、8-8、1994。
5. 日本工業標準調査会：J I S 規格票の様式 J I S Z 8301、日本規格協会、東京、1993。
6. 日本工業標準調査会：プログラム言語MUMPS J I S X 3011-1995、日本規格協会、東京、1995。

【付録】MUMPS-JIS用語（抜粋）

and operator (&)	論理積演算子(&)
argument	引数(ひきすう)
indirection	引数の間接指定
list	引数並び
less	引数なしの
arithmetic binary operator	算術2項演算子
atom	子(最小の単位)
attribute block	属性ブロック
binary operator	2項演算子
call-by-reference	参照呼出し
call-by-value	値呼出し
character pointer	文字指標
collating order	照合順序
command	命令
concatenation operator ()	連結演算子()

8 M言語 (MUMPS) の J I S 化

conditional command	条件付き命令
contains operator (I)	包含演算子(I)
data-cell	データセル
data type	データ型
data value	データ値
defined	確定
defining occurrence of label	実在のラベル
delimiter	区切り記号
difference operator (-)	減算演算子(-)
directory node	ディレクトリノード
division operator (/)	除算演算子(/)
division operator, integer (\)	整数除算演算子(\)
empty string	空文字列
empty value	空値
entry	項目, 見出し
entry reference	入り口参照
equals operator (=)	等価演算子(=)
exclusive	排他的
execution level	実行レベル
execution location	実行位置
exponential expression atom	指数式子
exponential value	指数値
expression	式
atom	式子
tail	式尾
extrinsic	外部の
function	外部関数
special variable	外部特殊変数
follows operator (I)	追従演算子(I)
FOR scope switch	FOR有効範囲スイッチ
formal	仮(の)
frame	枠
full (string)	正式(文字列)
full reference	正式参照
function	組込み関数
global variable	大域変数
global local variable name	変数名
greater than operator (>)	大号演算子(>)
global variable name	大域変数名
implicit	暗黙の
indirection	間接指定

integer division operator (\)	整数除算演算子(\)
integer	整数
interpretation	整数解釈
literal	定整数列
truth value	真偽値解釈
intrinsic function	組込み関数
intrinsic special variable	組込み特殊変数
label	行ラベル
offset	行ラベル相対位置
reference	行ラベル参照
less than operator (<)	小号演算子(<)
level	深さ
execution	実行レベル
indicator	レベル指標
line	レベル行
precedence	優先順位
line reference	行参照
line body	行体
list	並び
literal	定数
local variable	局所変数
local variable name	局所変数名
locktable	LOCK表
logical operator	論理演算子
minus operator (-)	減算演算子(-)
modulo operator (#)	剰余演算子(#)
naked indicator	略式指標
naked reference	略式参照
name	名前
name indirection	名前の間接指定
name-table	名前表
named partition storage	名前付きパーティション記憶域
named system storage	名前付きシステム記憶域
name value	名前値
nest	入れ子
nesting level	入れ子の深さ
node	ノード
designator	ノード指定子
not operator (!)	否定演算子(!)
name reference	名前参照

10 M言語 (MUMPS) の J I S 化

number	数
numeric	数値
data value	数値データ値
expression	数式
expression evaluation	数式評価
interpretation	数値解釈
literal	定数字列
relation	数値関係
value	数量値
operator	演算子
arithmetic	算術演算子
concatenation	連結演算子
logical	論理演算子
metalanguage	超言語演算子
pattern	パターン演算子
precedence of	演算子の優先順位
relational	論理演算子
truth-value	真偽値演算子
unary	単項演算子
or operator (!)	論理和演算子(!)
parameter frame	パラメタ枠
parameter passing	パラメタ渡し
partition	パーティション
marker	パーティションマーカ
space	パーティション域
stack	パーティションスタック
storage	パーティション記憶域
pattern	パターン
atom	パターン子
code	パターン符号
indirection	パターンの間接指定
match	パターン照合
match operator (?)	パターン照合演算子(?)
plus operator (+)	加算演算子(+)
post conditionals	後置条件並び
product operator (*)	乗算演算子(*)
quotient operator (/)	除算演算子(/)
integer (\)	整数除算演算子(\)
relational operator	関係演算子
routine	ルーチン
body	ルーチン体

execution	ルーチンの実行
head	ルーチン頭
name	ルーチン名
reference	ルーチン参照
scope	有効範囲
marker	有効範囲マーカ
scoping	有効範囲
FOR	FOR有効範囲
variable	変数有効範囲
separator	分離子
setpiece	左側\$P
single atomic operation	単元操作
special variable	特殊変数
string	文字列
function	文字列関数
literal	定文字列
operator	文字列演算子
relation	文字列関係
subscript	添字 (そえじ)
subscript indirection	添字の間接指定
subtraction operator (-)	減算演算子(-)
sum operator (+)	加算演算子(+)
truth operator	真偽演算子
truth value	真偽値
expression	真偽値式
expression atom	真偽値式子
interpretation	真偽値解釈
tuple	値組
type conversion	型変換
type declaration	型宣言
unary operator	単項演算子
undefined	不定
unlock	LOCK解放
value-table	値表
variable	変数
context	変数環境
name	変数名

The USA MUMPS Activities

John Dumas

1 M Innovations 1994

My name is John Dumas and I am from the U.S. Veterans Administration. We are the largest user of M in the United States. In the past year, many new ideas and concepts have evolved particularly in the methods used to display the data to the users.

I will present to you these ideas, especially those that will make M a more competitive tool in the marketplace.

2 Users Demand Changes

The users of M have become more sophisticated in computers and are becoming more familiar with personal computers and full screen displays. Users are requesting M database displays similar to what is available in the windows enthronement. The normal display for M with terminals is a line by line, or 'progressive display' screen. This is not acceptable to many users who now request full screen displays and 'point and click' database inquiries.

The demand also exists to interface with many 'off-the-shelf' products that have been developed and marketed by commercial companies. Most of these packages are not written in M and interfacing these systems to M create many complicated problems.

3 Multiple Problems Exist

The first problem we encountered is that most of users are accessing the database with RS-232(serial) terminals using ANSI codes. We have over 2,200 terminals in place. This works fine with progressive display, but serial communications is too slow for full screen graphical displays, so the network must be extended to allow workstations. Now we have to learn a new technology and install networks for workstations. Replacing the serial wiring with network wiring is a costly enterprise. Fortunately, 10BaseT Ethernet is able to utilize the existing telephone wiring and allow both voice and data communications.

A terminal is an easy machine to master for the user. Installing a workstation (personal computer) requires the user to be trained in the use of the machine and associated software. Our training costs have more than quadrupled in the past six months and we have only 200 workstations installed. Also, the cost

of a terminal is about \$300. Compared to the cost of a suitable workstation, \$2,400, our ADP budget is expanding considerably.

Designing programs that will run on the workstation and access the M database requires an inexpensive interface. This problem has been partially solved with two products.

4 Progressive Display: Background

Scrolling screen is the fastest and the easiest way to display data. But, as other languages evolve, the display becomes more user friendly and flashy. M lags behind. In order to stay acceptable, full screen displays must become available to the users.

VA Fileman v20 has taken a step in this direction by offering the programmer the ability to provide full screen displays. But, the screen and cursor controls remain archaic and are still not acceptable to the users.

Our goal is to provide the user with a full screen graphical user interface (GUI) that will allow the user to use the full capabilities of their workstations.

5 Progressive Display: Status

We have been using DEC (Digital) DDP DOS to interface the workstations with M systems. This Ethernet protocol allows the workstation to be a M node on the network, and to access the database exactly as if the workstation were a M server. We have encountered a problem with DDP DOS and Windows. Fortunately, DEC, hardware vendors and Microsoft are assisting in the development of this interface and aiding us to overcome these difficulties. This also requires the programmer to learn a language other than M. All workstation software is programmed in either Microsoft Visual Basic or Microsoft Visual C++. Programs are then royalty free and the only recurring cost is the license for the DEC DDP DOS.

Micronetics M for Windows (GUI) is an alternate method of interfacing with an M database. This requires M to be running on the workstation and becomes an actual node on the network. The programmer is able to create all screen displays and interfacing in M. This option requires the purchase of a MSM system for each workstation.

6 Language Interface: Background

As previously mentioned, DEC DDP DOS will interface directly with M servers providing the most direct link to a M database. DDP DOS allows calls to the protocol to fetch and to write data. We worked with DEC about two years ago to develop this protocol and originally used it to link PC based MUMPS systems to our main database. A license cost about \$75 per node.

To date, the only alternative to the DDP DOS protocol is MSM GUI workstation. This system is much simpler to integrate into a M facility, as the M programmers do not require additional language capabilities. The maintenance for such a system will be large, as each workstation maintains a M system and requires updates. The cost of each workstation could be as high as \$300 each.

7 Language Interface: Status

As seen in the demonstration, Microsoft Visual Basic for Windows performs very good and is the easiest language for a M programmer to learn. Development in Visual Basic is quite easy and the M programmers adjust to it quickly. Interfacing to other languages and database, especially graphic images is its strong point.

Once a base module (API) is created to make the calls to the DEC DDP DOS protocol, programming is easy. Writes, Reads and global lists are almost automatic. DDP DOS is limited in that it is a one way street. Only the workstation may query the M server, the server may not address the workstation.

Micronetics M workstation was readily accepted by our programmers. They have been able to rapidly create full GUI interfaces to the M database and are not concerned with learning a new language, just the M API and the related M syntax. Development is progressing quickly using MSM.

Whichever method is used, entirely new programming concepts are required. Now our programmers are faced with Object Oriented Programming (OOPS) and require a different mindset to accomplish design and implementation of software.

8 Key Issues

The cost of a general purpose workstation to run Windows is around \$2,400 in today's market. If the programs are created in a language other than M, such as Visual Basic or Visual C++, then these programs may be distributed royalty free. The only cost would be for the DEC DDP-DOS protocol. On the other hand, if each workstation requires a M system to be installed, workstation costs will rocket.

Another requirement is the number of programmers required to alter our current database systems and to design the new workstation software. M programmers that can make the switch to GUI programming will be in high demand. Currently we are suffering under budget cutbacks and positions are being eliminated. Consequently, we are contemplating using commercial contract programmers to design some of our software.

An effective and inexpensive interface to an M database is required in order for M to progress to this level. PC workstations are definitely in the future, but without an M interface, will only be another terminal on the system.

9 Next Steps

The need for an inexpensive interface between a PC workstation and the M database systems is paramount. Without this tool, M will remain a stagnant language. It is our job to discover this interface and make it available.

Develop intuitive and user friendly screens. A screen that has multiple functions and provide the users what they want and need will move M to the forefront of database managers. M is still the best in the world, but we have to provide data displays in a more advanced format.

A seamless integration to digital imagery is required. This will allow the user to view X-Rays, documents, lab results, photographs, etc. Providing a link between data transcriptions and the associated images is mandatory.

We are currently experimenting with CD-ROM for the archiving of data and for data storage. We now have over two years of medical data transcribed documents on line for the clinicians to view. Our goal is to provide a paperless medical chart.

BACKGROUND AND TECHNICAL FEATURES OF DTM AND DTM-J

Eric K. Pape

This paper will discuss the current state of the InterSystems and DataTree business relationship, the technical partnership between InterSystems and Japan DynaSystems, and the M products currently available and planned from InterSystems and Japan DynaSystems.

1 Business background

Eighteen months ago, DataTree combined with InterSystems. As of today, InterSystems and DataTree are now completely integrated companies. InterSystems continues to expand geographically, with staff in Brazil, Belgium and Australia. Revenue growth and profits continues to be strong.

InterSystems has developed three key technology alliances. IBM is a "Marketing Development Partner". InterSystems is a "Joint Marketing Partner" though a program provided by Dell Computer. Lastly, and most importantly, InterSystems has a strong technology partnership with Japan DynaSystems, Inc. (JDS).

The relationship between InterSystems and JDS is a long term partnership which began as a relationship between JDS and DataTree. JDS is the exclusive distributors in Japan of the DTM products from InterSystems. Also, JDS co-developed the two-byte version of DTM, called DTM-J.

2 InterSystems Products

InterSystem's version of the M Language currently runs on more than 24 hardware or operating system platforms. The list of compatible platforms includes Alpha OSF-1, Alpha VMS, Altos, Chicago, DEC RISC, DEC VAX, DG AViiON, DOS, HP, IBM RS6000, ICL DRS6000, MIPS, Motorola, NCR 3000, Novell Unixware, NT, Pyramid, Sequent, Siemens, Solaris, Sun, TI, Unisys, and Windows.

3 Overview of DTM-J a two-byte implementation of M

The DTM-J version of DTM is currently at release level 4.5. It is based on DTM, a version of the M language created by DataTree. The two-byte Kanji translation, DTM-J, runs on PC and PC compatible machines. Full "logical character" support is provided for Kanji Shift-JIS characters, and the implementation is compliant with proposed JIS M standard. JDS has established the goal that DTM-J

InterSystems Corporation,

One Memorial Drive, Cambridge MA 02142, USA

TEL 617-621-0600 FAX 617-494-1631

versions be available within two months of commercial release in USA of the corresponding DTM product.

4 DTM enhancements

The next version of DTM, currently being released in the United States, is version 6.0. This version is designed with the goal of high availability. It is intended to provide round-the-clock operation in enterprise networks. Additionally, it increases database and connectivity limits and it supports Visual M as well as M/WAPI

There are some language feature enhancements which have been added to DTM 6.0. Source and object compatibility with DTM 4.x versions has been maintained. DTM version 6.0 supports larger routines, with sizes up to 30K. A larger symbol table has been provided, with partition space up to 145K. DTM 6.0 is integrated with the M/SQL tools from InterSystems.

Some of the new elements added in DTM 6.0 are aimed at better supporting global databases. These include larger datasets, which can now be up to 32 Gigabytes. An important new feature is "global hardening". This added robustness provides "physical" integrity even if power is lost during a database update operation. Pointer and structural components of the database remain in-tact after such an unintentional shutdown. An important new feature is the ability to do "on-the-fly" integrity checking of databases. Also, some customers find the addition of DSM-compatible string collation to be a valuable feature.

5 DTM networking enhancements

Another area where DTM 6.0 has added substantial enhancements is in our networking software. InterSystems had added high-speed networking to other M/SQL systems running ISM (InterSystems M). Also added is a feature called network hardening, which means the ability of clients to recover from lost connections to servers without loss of LOCKs or data. In the event of the loss of connection clients wait gracefully for reconnection.

Networking in DTM 6.0 can support many more clients than in prior versions. The limit has been raised from 253 to a theoretical value approaching 64 thousand. This greatly expanded connectivity is provided by a new capability called server-router. This feature has been in use for a few years at Brigham and Women's hospital, where the DTM network currently is used to support a few thousand concurrent clients.

Larger networks usually require 24 hour uptime. This capability is supported in DTM 6.0 by the addition of "dynamic namespace mapping", "dynamic network configuration" and enhancements to our database "shadowing" feature.

DTM 6.0 has changed the mode of system configuration. It makes use of the configuration method used by Microsoft Windows, with a text readable text file named DTM.INI. Configuration elements which used to be separated into various DOS files are now centralized in the DTM.INI.

6 DTM enhancement plans

Plans for the future enhancements to DTM continue to be driven by InterSystems perception of customer needs and the desire to continue to provide the best client/server networking capabilities in the industry. Among future plans are drive-spanning datasets. Part of the "Visual M" product will be an enhanced editor and workplace. As with the rest of the client/server software industry, central network management utilities are planned, as well as stronger configuration utilities. Improved security will be a significant addition, as well as a real time dataset compression utility.

7 A development environment in M which uses Visual Basic

An entirely new user design and implementation model is provided in a visual development environment called Visual M. This product consists of a Visual Basic (VB) Design Environment which communicates M code and VB properties to DTM. DTM can be networked to M servers on a PC VAX or Unix environment. This combination makes M programming as convenient as Basic programming in the VB environment

M and VB are merged in a programmer environment which is a component of Visual M and which is called the "M/Desktop". In the M/Desktop the programmer can use any VB control and program M code to respond to it. To do this, the programmer must double click on the VB control and Enter M code for that control in the M/Desktop code editor.

In order to provide for code which links M and VB InterSystems has used a language syntax extension to indicate that M is interacting with Visual Basic. A leading underscore indicates a reference to a Visual basic property or method. For instance, the code to execute a method would look like:

```
DO _CustomerList.AddItem(argument)
```

A property can be referenced as in:

```
IF _CustomerName.Text=value
```

And modifying a property can be accomplished with the syntax:

```
SET _Age.Visible=0
```

8 DTM-J enhancement plans









Enhancements to the DTM product line can be expected from Japan DynaSystems. Release DTM-J 6.0 is due for release in the near future. In addition to working on a Kanji two-byte version of Visual M, Japan DynaSystems is also working to provide:

- NEC 9801 version of DTM-J

- New two-byte versions for Korea, Taiwan, and mainland China

- the InterSystems SQL toolset

表1 MTA Japan 1994講演内容

- 1 
- 2  **InterSystems Recap**
 - InterSystems and DataTree completely integrated
 - Expansion with staff in Brazil, Belgium, Australia
 - Strong revenue growth and profits
- 3  **InterSystems Technology Alliances**
 - IBM
 - Marketing Development Partner
 - Dell Computer
 - Joint Marketing Partner
 - Japan DynaSystems Inc.
 - Technology Partner in Japan
- 4  **InterSystems - JDS**
 - Exclusive distributors of DTM products in Japan
 - Co-developers of the Japanese 2-byte version: DTM-J
 - Technology Partner, first with DataTree , now with InterSystems
- 5  **M Computers (Fig.1)**
- 6  **What is DTM-J 4.5?**
 - Based on DTM: runs on PC and PC compatible machines
 - Full “logical character” support for Kanji characters
 - Compliant with proposed JIS M standard
 - DTM-J versions available within 2 months of commercial release in USA
- 7  **Overview:**
What is DTM 6.0?
 - Designed for high availability
 - Round-the-clock operation in enterprise networks
 - Increases database and connectivity limits
 - Supports M/VB as well as M/WAPI
- 8  **DTM 6.0 Language Features**
 - M source/object compatible with DTM 4.X
 - Large routines (up to 30K)
 - Large symbol table (up to 145K)
 - Integrated with M/SQL tools

- 9 **DTM 6.0 Global Database**
- Large datasets (up to 32Gb)
 - Global Hardening (“pull-the-plug”)
 - On-the-fly integrity check
 - DSM-compatible string collation
- 10 **DTM 6.0 Networking**
- High-speed networking to M/SQL
 - Network hardening
 - clients wait gracefully
 - On-the-fly network configuration
 - Server-Router model
 - like BWH
- 11 **DTM 6.0 Configuration**
- DTM.INI File - readable and centralized
 - Dynamic namespace mapping
 - Dynamic network configuration
 - Shadowing
- 12 **Future Releases of DTM**
- Drive-spanning datasets
 - Enhanced editor and workplace
 - Central network management utility
 - Stronger configuration utilities
 - Improved security
 - On-the-fly dataset compress utility
- 13 **M/VB Development Environment (Fig.2)**
- 14 **Merging M and VB (Fig.3)**
- 15 **Using VB Objects in M**
- Execute a method
 - DO _CustomerList.Additem(...)
 - Reference a property
 - IF _CustomerName.Text=““ ...
 - Modify a Property
 - S _Age.Visible=0
- 16 **Coming soon from JDS**
- DTM-J 6.0
 - M/VB
 - NEC 9801 support
 - New versions for Korea, Taiwan, and mainland China
 - SQL toolset

Fig.1

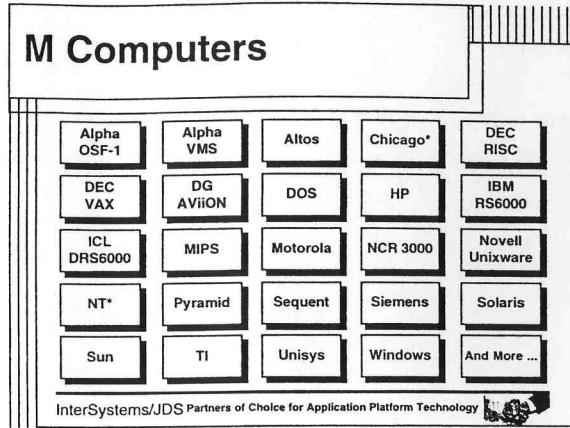


Fig.2

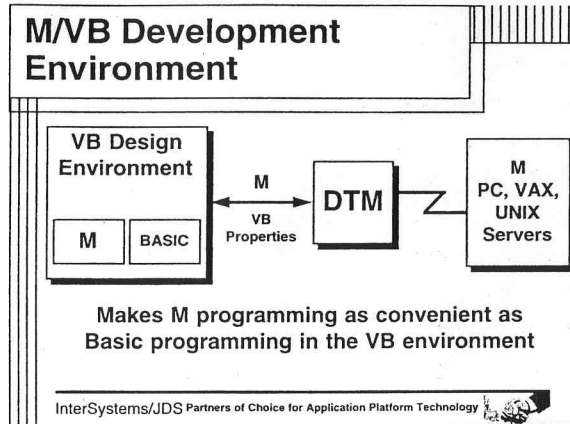
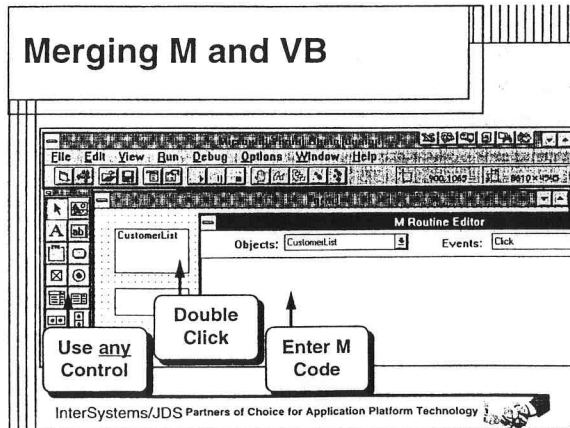


Fig.3



The Future of M

Robert P. Mappes

皆様の第21回年次大会で講演をさせて頂くことを会長、大会長に感謝します。MTA-Japanで講演するのはこれが3回目です。Micronetics Design Corporationを代表して我が社の製品を、それが住友電工からのM-, SP-MUMPSであろうと、Micronetics Standard Mを利用してくださる皆様に感謝申し上げます。

今日は、それが特に日本、韓国、中国、台湾での成長に関連するMの将来についてお話し致します。

先ずアメリカでのMの成長の私の経験から、私達がやってはいけないことと、やらねばならないことを中心に話します。

1 Background

- ∴ 1992 Gartner Group - Executive Summary
- Worldwide DP market
 - Double from \$350 billion to \$700 billion
 - Dominated by software and networked system solutions
 - Evolution of new app. dev. and user tools
 - GUI, CASE, SQL, client/server, open systems
 - M must modernize its technology to succeed

[背景] 1991年にベンダー群/ユーザ共同体はガートナーグループに「MUMPSの市場と将来性」という調査を依頼した。報告書のサマリーでは、世界のデータ処理市場は3500億ドルに始まって10年後に7000億ドルの市場となる成長があり、この成長はソフトとネットワークの解決をもつ全てのハードウェアプラットフォームを含むものである。この10年間に持続的な新しい応用面とユーザーツールの開発が進み、それらはGUI, CASE, SQL, クライアント/サーバー、オープンシステムが主流となる。成功するためにはM言語は技術を現代化しなければならない、と結論しました。

- ∴ 1991 M and M related products - \$1 billion
- Major growth from:
 - Expansion to Europe and South America
 - Movement beyond the medical community
 - Downsizing to a PC platform
 - Creative ideas for new applications

MICRONETICS DESIGN CORP.

1127 WAGON WHEEL DRIVE

SKANEATELES, NY 13152 U.S

TEL 001-1-315-685-2037 FAX 315-685-2041

1991年のガートナーの試算ではMとMに関連した生産物の世界中での利益は10億ドルとした。おもな成長の部分は、ヨーロッパと南アメリカ、医療の世界をのり越えた動き、PCプラットホームまでダウンサイジング、そして新しい創造的なアプリケーションによる、とした。

2 Technical Enhancements

- Provide Windows capability
- Offer a Graphical User Interface
- Enhance networked PC capability
- Provide client/server interconnect
- Provide open connectivity across application and database platform
- Provide a CASE development capability for larger more complex applications
- Add image processing and document management
- Agree on standards

ガートナー報告書は、Mが2つの点に集中しなければ成長はないと指摘した。第1の点は技術的な補強でありMを90年代の他の進歩したソフトウェアとユーザの要求に追い付かせるためには次のものを完成させよと説いた。

ウインドウ

GUI

PCネットワーク

クライアント/サーバ相互結合

アプリケーションとデータベースを通じてオープンな結合性をもたせる

複雑なパッケージでCASEに匹敵するものを作る

画像処理と文書処理を追加する

標準手法として以上を行うこと

3 Market Awareness

- Overcome views that M is old technology
- Overcome perception that M is only for the medical/hospital community
- Provide more education and training
- Expand the acceptance of M by computer and computer adjunct manufacturers
- Participate in industry conferences, exhibits, seminars
- Formulate a promotion program

第2の点は市場のM認識度に十分に注意し、Mの共同体はそれ以外の社会にMを知らせようとしなければならぬことである。それには、次のことが必要である。

Mが古い技術だというイメージを克服する

Mが医療/病院の世界のものだという認識を取り除く

教育とトレーニングの方法と資源を整える

コンピュータおよび周辺機器メーカーにMを受け入れさせる
 産業界の会議や展示会、セミナーに参画する
 プロモーションのためのプログラムを今作ること

4 Executive Summary

∴ Conclusion

- Given these actions, we believe that M can improve upon industry growth and double its size to over \$2 billion by 1996.

以上の2つの点を実行すればMは1996年迄には20億ドルの市場に倍増すると、ガートナー報告書は結論しました。

5 Status 1994

∴ Technical Enhancements

- Provided Windows capability
- Provided a Graphical User Interface
- Enhanced networked PC capability
- Provided client/server interconnect
- Provided open connectivity across application and database platform
- Provided a CASE development capability for larger more complex applications
- Added image processing and document management
- Agreed on standards

今年のANSI/ISO草案(4冊)ではこれらを全て完成した

ウインドウ

GUI も

PCネットワークの強化

クライアント/サーバ相互結合ができた

アプリケーションとデータベースを通じてオープンな結合性をもたせた

少なくとも1つのCASEに匹敵する複雑な開発ツールのパッケージがある

少なくとも1つの会社で画像処理システムが稼働を始めた

MDCは今回の拡張は標準的拡張を満足させたとおもわれる

6 Status 1994

∴ Market Awareness

- M is old technology
- Name change from MUMPS to M
- Addressed technical enhancements

- M is only for the Medical Community
- Generic booth at COMDEX
- Provide more education and training
- Expand the acceptance of M by computer and computer adjunct manufacturers
- Participate in industry conferences, exhibits, seminars
- Generic booth at COMDEX
- Formulate a promotion program
- M is Japan INDUSTRY Standard
- COMDEX activities

誠に残念なことはアメリカで市場の認識を変えることが出来なかったことです。その原因は医療にMを閉じ込めたことです。ガートナーが示したように、Mは古いテクノロジーであり、元来が医療/病院のものであると市場が認識していることは、1つの会社がいかなる犠牲を払っても払拭することは出来ないのではありません。私がIBMにいたときの経験からしても、このような市場認識があるところでそのイメージを変えるための企業的な計画を立てることは出来なかったと思います。

たった今、MTAグループが全体となって、この問題に取り組み、今までの体質とやり方を変えなかったならば、どれだけ技術革新がMに起こってもその結果を享受することは出来ません。

今Mの共通の目的を遂げるため、全ての産業がこれまでにMで完成させた結果を持ち寄ってどれだけ成果を挙げてきたかを市場に示さないでM市場は死んでしまうということを強調しなければなりません。

M装備会社の殆どと、システムツール開発会社の全てがこの目的のためには足並みを揃えるはずですが、しかし、年間2-3百万ドルから数千万ドルのソフトウェア開発会社の殆どは今すでにMテクノロジーの支持者として動くことは出来ません。それは、今後も安楽にやってゆけるという将来の保証があってこそ出来ることです。

MがMUMPSから名前を変えたことは、今までの古い体質から脱して新しい革新的な市場に適した技術をもみにつけていることを示すためでありました。しかし、これは古い物が衣装だけを変えて新しいと言おうとしているという逆効果をもたらしました。MTAの活動を続けている顔ぶれが旧態依然ときれい事を、伝統だけを重んじたアカデミックな閉鎖的な形態の中で続けているのです。これを腐敗の時代と言います。これがMを死に追いやるのです。

私達は医療のMという認識を払拭して、一般的なコンピュータ産業というイメージに作り変えるためにCOMDEXで一般ブースを作りました。ここでは医療外のMアプリケーションが企業コンサルタントと一緒に一般にアピールしました。ここでは、教育をM装備社に任せ、医療と関係のない大学や、会社から講師をだしました。

コンピュータメーカーとその周辺機器メーカーのMの受容のための活動は装備社にまかされています。Digital Equipment Corp.を除いてコンピュータメーカーがM市場拡張のために活動している所はありません。

すべての産業のコンファレンスをCOMDEXがスポンサーしました。840名以上がMブースに来て、200名が資料を請求し、その50%以上がソフト屋やコンサルタント企業で、その後の繋がりが出来ました。市場受容がどう変わるかを計量予測するにはまだ早すぎます。1995年度はMTAがどのようなプロモーションを考えているかも分かりません。

Mは1995年に日本の工業規格になります。日本のCOMDEX的な活動が期待できます。

7 Results

- ∴ United States - Growth (Gartner projected 20% in 1994)
- Vertical growth from application developers and existing customers
- No increase in the number of application developers
- M implementors' growth often at the expense of another M implementor

今年はアメリカで20%の成長率をガートナーは示しました。私は疑問を抱きます。医療用アプリケーションは年9%であり、Mの固定ユーザは施設内で拡張をしています。そのほかにMを開発ツールとして市場に出すところを入れると20%になるかもしれません。しかし、MTAの古い体質が変化しないことには、成長どころかMの死をまねくのも時間の問題です。

今、起こっていることは装備会社同志の共食い現象です。DSMとISMの400のソフトライセンスをもつVA内のPDP機種800台がPCベースのMSM製品に置き換えられました。結局、アメリカのMの成長は医療という固定ユーザーとソフト屋さんの仕事を作りだしているものであって、ソフト屋さんの数は増加していません。装備会社は他の装備会社を殺すことで生き延びているのです。装備会社の寡占現象が起こればM産業の発展はパーカー万年筆のように固定ユーザのためにのみあることになり、もう技術革新は起こらないでしょう。

8 Growth Opportunity

∴ Gartner: Over 50% growth outside the US

"However this opportunity will not be realized without the implementation of the technology enhancements and market promotions as specified."

ガートナー報告書は成長率の50%は外国に依存すると言います。それは外国での成長率を高く見ているからです。尤もこの成長率はMに技術革新と市場拡張努力がなされた場合のことを前提としたものです。

9 Observations

∴ Observations of M Market in Japan:

- Primarily medical
- Limited involvement by computer manufacturers
- Looks too much like US market

三回日本にやってきて観察したところでは、私は第一義的にMのマーケットは医療に限定されていると見えています。今日の会合であなたの周りを見てください。登録台帳を見てください。コンピュータメーカーが何人いるのでしょうか。Mで偉大な成果を挙げている会社から何人出てきているのでしょうか。

尊敬の念を込めて申し上げます。この姿がアメリカのMなのです。この姿を取り続けられるならばアメリカと同様に日本ではMの年50%の成長はありません。

1 0 Suggestions

∴ If M is to achieve its potential in Japan DO NOT do as has been done in the US.

-Make every effort to implement the Market Awareness actions as defined by Garter

日本ではMがJapan Industry Standardとして制定される時にあります。この時に当たって申し上げます。日本はアメリカの過ちをこれ以上繰り返して下さいますな。

市場意識を失わないで下さい。アメリカではMの装備は政府の応援で行われ、アプリケーションは政府の金で作られユーザーにとっては唯でした。ユーザーは政府によって作られその機構は閉鎖的にMTAに残っております。市場性を向上させる機構とは全く正反対の形です。

日本にはそれを模倣したアカデミックな学会権威主義が見られます。おそらく、医療と非医療という分類がなされるかも知れません。しかしそれも、白人と非白人という分類をした白人の思い上がりに等しい時代錯誤なのです。Mは医療のために有るものではありません。もしも、そのように思わせるマネジメントを行えば医療畑の人が医療の名においてMを殺し、延いては医療を殺すことになります。

確かに、医師たちは退き工学者に席をゆずりました。しかし工学者たちは以前に増して医療施設やアカデミーの中から、一層形骸化した学会権威主義の視野の狭い見通しのない方法でM産業を殺すことを続けているのです。これはMの自殺行為と言うものです。1964年にIBMが360を発表した時にCOBOL, FORTRAN, PL/I, RPGがプログラム言語の主流でした。1980年代にMicronetics, InterSystems, DataTreeなどの企業家が現れて将来の可能性を見たときに私はMの将来を見てIBMを説得しました。私は歴史家ではないがコンピュータ進化の時代を生き抜いてきました。アメリカで生まれた技術は欧州に渡り、さらに日本やラテンアメリカなど他の地域に広がってゆきました。Mの時代はヨーロッパに来ております。次は日本に花を咲かせることでしょう。言っておきますが、アメリカが行ったMTAの今の医療中心主義のやり方を変えないかぎり絶対にMの花が咲くことはありません。

日本でMがJISになることは、COBOL, FORTRAN, Cなどと肩を並べて今後Mが開発ツールとして脚光をあげ、客観的な評価を受ける時代が来たことを思い、私は今極度に興奮しております。しかし、一つ誤れば、アメリカでANSIになろうとFIPSになろうとMに市場性が見られないのと同様に日本でもJISの飾り棚に入ってしまう危険があります。それはたった今、MTAの意思決定によって、Mが世界レベルの実用的な技術になるのか、または壁龕の中の飾り物となってしまうかの世界の風潮が変わってしまうという大切な瞬間にあることを申し上げたいと思います。

PCのOSは只になり、WindowsもWindows-NTも日本には新しく応用パッケージはこれからという次期に当たります。Mのチャンスは今ここに来ているのです。それをどのようにプロモートするかという日本の企業戦略的な上手さを、今発揮すべき時にあります。私達の未来なのです。閉鎖的なMの進化と発展の過程をここで絶ち切らねば私達の未来はありません。

1 1 What Micronetics is doing

- Technically:

- Micronetics products available double-byte enabled and localized for Japanese characters

- PCs, UNIX, VMS and Windows-NT

- Marketing:

- Assigned dedicated resource to International Market Development

-Working with computer manufacturers

●Participation in industry conferences, etc.

Micronetics Standard M (MSM)は住友電工から1980年代からPCの為のSP-MUMPS, UNIXの為のU-MUMPSとして出されておりました。我々との関係は良く最近契約を更新しました。しかし、住電は他の開発屋に製品を与えなかったことを理解しながらも、他の開発屋さんとMicroneticsの間に問題を起こすことを知りましたので、自身で日本、韓国、中国、台湾その他の地域で必要となる2-バイト環境を整え、誰とでも直接取引できる様にしました。これは最新のMSM 4.0やその他の全ての製品を含みます。ガートナー報告書が示した技術拡張はすべて備えております。

私は国際部長になったのですが、Mをどうやってプロモートすればよいかについて、皆様にはアメリカ同様にその戦略はないだろうとおもいます。しかし、日本では応用ソフト生産会社を大切に扱うことだと思います。彼らがMフェアに出てこなければなりません。そして、これをハードウェア会社にバックアップさせることです。

日本のMの将来のために私は働きたいと思います。日本はアメリカが陥っている状況避けることは出来るはずで。それが出来れば日本と東南アジアでのMは大盛況が来ると信じております。

A New Platform for Brigham and Women's Hospital

Robert Beckley, Pasha Roberts
Michael Monroe, Jim Marra,
Frederick L. Hiltz, Ph.D., P.A. McManus

1 Introduction

In 1989, Brigham and Women's Hospital (BWH) in Boston faced an unpleasant situation. The current computing platform was running out of capacity and processing power, and was based on technologies that did not have a long life. Its crude data communications and GUI support capabilities limited the ability of the platform to support sophisticated clinical applications.

Demands for computing and application sophistication were increasing. The hospital projected that the current platform would begin to hinder application development within two years.

In response to this problem, the hospital is converting its applications to the New Platform (TNP). TNP involves the conversion of the existing MIIS applications to MUMPS and running the applications on a very large network of Intel-based processors.

The conversion to TNP has now reached its second phase of development and there is no reason to believe that TNP will not meet its goal of providing an integrated hospital information system for at least 4000 users.

2 Background

The system in use at BWH since September, 1983, employs an internally developed integrated database that covers almost all hospital functions. There are currently 96 modules which include ADT, outpatient registration and scheduling, laboratory, pharmacy, results reporting, payroll, and accounts receivable. All of these applications run on a network of Data General minicomputers using the MIIS operating system and language. MIIS is a proprietary language written by Meditech, Inc., and is in the process of being phased out.

The network currently supports over 1700 users, connected to any of 14 different minicomputers by serial links. The hardware limit for the network is 15 machines. At the present rate of growth, the hospital will run out of lines to allocate to terminals in less than a year.

BWH strove to create a system that would use a more standard architecture and be expandable, in order to avoid the problems associated with proprietary technology and capacity limitations. MUMPS, an

Brigham and Women's Hospital, Information Systems

10Vining Street, second floor

Boston, MA 02115 USA

TEL 617-732-7058 FAX 617-732-5187

ANSI standard language, was chosen as the development language. The LAN would be composed entirely of Intel processor based computers.

MUMPS runs on mainframes, minicomputers, and PCs, and is installed all over the world. The hospital selected Data Tree Mumps (DTM), the version of MUMPS that makes the best use of networked PC technology. Workstations and servers are currently all IBM PS/25, but IBM compatibles would work equally well. Most of the processing occurs directly on the workstation; so adding users taxes the network very little. With all of these standard parts, TNP should weather the gyrations of the marketplace.

3 Architecture Overview

As PCs become more powerful, so will TNP. The current base model workstation is 16 Mhz 386SX machine, but as costs drop a 33Mhz 486 box might replace that for the same price.

TNP was designed as a collection of interconnecting parts made to look like one big computer. This makes the tasks of the hospital's application programmers easier, since they never need worry about which particular piece of hardware holds a certain database. They can address all files logically. The Brigham will continue to develop applications internally; however, TNP provides hooks for outside software to function.

The original plan for TNP called for 100 such applications, each living on its own server, to be spread across the network. Each of these servers would support an average of 40 workstations in a ring. A rigorous queuing analysis showed that the scale-up from one ring to a network of 4,000 users ought to work. BWH decided to implement this design in three stages: a prototype phase, in which one small system would run under the new architecture; a limited deployment phase, where a limited number of users across the hospital would connect to a few additional applications; and full deployment, in which all applications would run under TNP, and hence the old platform could be dismantled.

4 Implementation Phases

The prototype local ring, containing the Accounts Payable server and workstations updating the database, went live in March, 1991. The objective of this conversion was to determine feasibility. This was not a perfectly smooth period, by any means. The system came up for one day, then suffered a major crash on the second day of operation that left it inaccessible to users. However, AP returned with full functionality and no data loss on the third day. The system has never missed its weekly generation of checks to vendors. Most users acclimated quickly to the new software.

In the meantime, TNP had entered the phase of limited deployment. Over the second half of 1991, workstations were installed in the patient care areas at BWH. In November, 1991, the Clinical Information (CI) system came on line in TNP. Using the same windowed (as opposed to roll and scroll) interface that AP uses, this set of programs allows clinicians to look up lab results for patients 24 hours a day, seven days a week. The results themselves are still entered in the old MIIS system and downstreamed to MUMPS. The objective of the CI conversion was to determine the ability of TNP to handle multiple queries that would span multiple servers.

In addition, the hospital's decision support system, CHASE, was converted to TNP in September,

1991. This conversion allowed the hospital to test the ability of TNP to support large batch processing.

In the full deployment phase, the remaining systems, such as Chemistry, Hematology, and Microbiology, will be rewritten in MUMPS and each put on its own server. This last phase will take at least three years to complete, as all of the current MIIS systems must be converted to MUMPS. Their servers and workstations will join the LAN. Hence with "only" 170 or so users on the LAN as of now, the system is still an order of magnitude or more smaller than it will become. This phase is underway now.

5 Architecture Detail

The architecture of the Brigham network has remained consistent with the original two-tiered design that was specified three years ago.

The network consists of a series of 18 local IBM token rings, allocated by geographical and load considerations. Each of these runs at 4Mbps over twisted-pair wiring to the workstations. A bridge connects each local ring to a 16Mbps backbone token ring, where the Novell servers provide file service and access to PC applications, such as WordPerfect, Quattro Pro, and Paradox. Many of the workstations are and will be diskless 386SX machines, which boot across the network using Novell's Remote Program Load. Notably, this Network Operating System (NOS) architecture has remained stable through an operating system migration to Novell 3.11 from IBM Lan Server.

The above is only the smaller half of the network. As mentioned earlier, the lion's share of hospital applications will be developed internally using DataTree MUMPS. Most workstations run and never leave DTM, entirely unaware of DOS, Novell, or their ability to run WordPerfect. The DTM network has its own two-tiered design, using the above-mentioned local token rings as a basis. Each local ring has a DataTree MUMPS server, which contains hospital data germane to that area. For example, the server attached to a local token ring in the Hematology laboratory would contain the Hematology database. To provide enterprise-wide data access, each server is attached to a second 16Mbps backbone token ring, dedicated to MUMPS servers. Thus, each server has two token ring cards: one local, and one backbone. The server software intelligently handles database requests from local clients, dispensing data from cache, locally-held databases, or across the backbone from another server, as appropriate.

DataTree MUMPS uses an applications interface called NETBIOS as its vehicle for networking. There are limits: NETBIOS will only support 254 NETBIOS connections per machine, and the token ring itself is similarly limited. Server communications require two sessions per server, therefore creating a limit of 127 servers.

Local client communications require one session, limiting each server to 254 clients. This makes, then, the theoretical maximum capacity to be 127 servers times 254 clients, or 32,258 workstations. The Brigham's goal of 4,000 easily fits within this limit, using 100 servers with an average of 40 clients each. Currently, there are 25 servers on line, serving a total of 170 clients.

6 Network Management

As we enter the age of the large enterprise network, we see very quickly that network management emerges as a dominant issue. The list of needs can be imposing: file and device sharing for thousands of

workstations, inventory control, security, statistics, backup, alert mechanisms, emergency procedures - all need to come into place somehow if staff are to keep their heads above water. The mechanisms for such control are available at three levels - at the wire level, at the NOS level, and at the application level. The Brigham employs controls at all three levels. The flexibility and power of DataTree MUMPS have encouraged a great deal of development at the applications level, as one can use APIs and direct queries of the network to gather information from the network and assimilate it in MUMPS.

For example, user security was a major complication, since users have to access three systems: the aging MIIS network, the new MUMPS systems, and Novell. Of course, most users would not accept or understand the need for three separate id/password combinations to access each, nor is this necessary. Today, a user is able to enter changes to her/his password or security levels in MIIS, have the change pass downstream into MUMPS, whereupon a MUMPS implementation of the Novell API enacts the necessary changes among the Novell servers. This is generally processed within seconds of the user's initial change. This is the kind of multi-level solution that will become more and more necessary as systems spread across platforms.

As the systems are required to have no scheduled downtime, system backup and fail-over become important issues. Each MUMPS server has a "shadow", a hot backup which is constantly maintaining an exact copy of the server's databases on its own disks. In the case of server failure, an operator chooses a menu selection named "Make Shadow into Server". Within two minutes of an server crash, an up-to-date, exact copy of the original server is available. All of the client sessions are automatically taken over by the new server.

This approach also allows system backup to be done on the shadows, so that the servers can remain on line and available around the clock.

Another application that has been developed in MUMPS is the "airline screen" monitoring system. Its name refers to the display, which shows an entry for every MUMPS and Novell server in the network, as a display at an airport shows the status of incoming flights. Each server runs self-diagnostic software once a minute, and as error conditions are encountered the node will change color on the "airline screen" display. Also, certain errors are corrected automatically by the alert system, if possible. In essence, this system makes the network self-reporting and self-cleaning. It allows operators, programmers, and managers to determine the status of the system at a glance. The number of clients on each server is displayed as well, so that a rough estimate of load can be taken. Much more detailed statistics are collected, tabulated, and graphed automatically by the system every six minutes, if the manager would like more information. Most of this information is about MUMPS activity; just this year the market is seeing the introduction of similar systems for Novell networks, such as LANAlert and NetWare Management. This is a welcome development and will be useful into the future.

7 Conclusion

The goal of the designers of TNP was and continues to be the creation of an integrated, enterprise-wide hospital information system to serve 4000 or more users. As it turns out, no other such system exists on this scale. A prediction of the outcome of the effort by the Brigham and Women's Hospital would, therefore, be something of a guess. From the perspective of those working on the system, however, all has

thus far proceeded as hoped.



新 開 発 環 境 E a s y - T o o l の 開 発 Development of A New Developing Environment Named "Easy-Tool"

大櫛陽一、三澤哲夫、古田土義裕、小西千恵子、高橋政志、平野健一
Yoichi Ogushi, Tetsuo Misawa, Yoshihiro Kodato,
Chieko Konishi, Masashi Takahashi, Kenichi Hirano

システム開発の一層の効率化と標準化、実行システムの高速化を進めるために、Easy-Toolと名付けた新しい開発・実行環境を開発した。Easy-Toolは、Easy-PatternとEasy-Partsから構成されている。Easy-Patternは、お手本のプログラム群である。Easy-Partsは、カスタマイズ用の部品群である。Easy-Toolにより開発されたプログラムは、一つの入力用画面が一本のプログラムに納まり、プログラムが非常にコンパクトで、管理がしやすい。一つのプログラムサイズは約20KBである。また、プログラムスワップとディスクアクセスが少ないため、実行スピードが高速化された。

(キーワード：開発環境、マルチウインドウ、イージーツール、イージーパターン、イージーパーツ)

We have developed a new environment for developing and executing applications that are named "Easy-Tool". We are aiming to promote efficiency, standardization and high performance. Easy-Tool is constructed by Easy-Pattern and Easy-Parts. Easy-Pattern is a group of standard programs. Easy-Parts is a group of utilities for programming. An application for one ordering screen is described by only one routine. The routines are very compact. One routine size is about 20K byte. We can maintain system more easily. The executing time is more rapid because of decrease of program swapping and disk access.

(Keywords: multi-window, easy-tool, easy-pattern, easy-parts)

1 目的

東海大学医学部では、1989年よりM言語(MUMPS)によるシステムの開発をおこなってきた^{1,2)}。これまでは住友電工のワークベンチを使用して、ベースパターンの自動コーディングを行い、内部処理や出力処理についてのオンコーディングを追加してプログラムを開発してきた。この方法は、ワープロ感覚で入力画面を作成でき、画面やコーディング標準化にも役立ってきた。

今回、システム開発の一層の効率化と標準化、実行システムの高速化を進めるために、Easy-Toolと名付けた新しい開発・実行環境を開発したので報告する。

東海大学医学部医学情報学
〒259-11 神奈川県伊勢原市望星台
TEL 0463-93-1121 ext.2140 FAX 0463-96-4301

Department of Medical Informatics, Tokai University School of
Medicine
Bouseidai, Isehara, Kanagawa 259-11, Japan
TEL 0463-93-1121 ext.2140 FAX 0463-96-4301

2 Easy-Toolの概要

Easy-Toolは、Easy-PatternとEasy-Partsから構成されている。Easy-Patternは、お手本のプログラム群である。洋服に例えるとイージーオーダである。イージーオーダでは、体のサイズに合ったものを選んだ後で、腕や足の長さに合わせて調整を可能している。この微調整用のコーディングに使われるのがEasy-Partsである。

ウィンドウ用エミュレータとして、従来からのMWE80（住友電工）を使用しているが、エミュレータの持つ多くの機能の内、ウィンドウのオープンとクローズ機能に限定して使用しており、その他はISO標準言語の範囲で作成している。このため、他のエミュレータやGUIへの移植は簡単である。

2.1 Easy-Pattern

Easy-Patternには、主画面とサブウィンドウ画面の標準プログラムが用意されている。各プログラムには、変更不可セグメントと、追加変更および設定セグメント、自動コーディングセグメントがある。変更不可セグメントでは、メインフロー制御、ウィンドウ番号の自動管理、画面表示、編集入力、入力チェックなどが行われる。初期値設定では、画面タイトル、ウィンドウサイズ、ファンクションキー表示内容などを設定する。各入力フィールドの項目名、画面上の位置と色、入力フィールドの長さ、また必要に応じて、入力前後の処理セグメント名、ヘルプ処理セグメント名、入力フィールド前後のガイダンスは、一行のプログラム内テキストとして設定する。入力チェックの内、パターンチェック式、範囲チェック式、間接実行によるチェック式、外部ルーチンによるチェックルーチン名は、項目名ごとにグローバルテーブルに登録され共有化されている。従って、この初期値と入力パラメータ行の設定だけで、実行可能なプロトタイプが完成する。入力前後の特殊処理、ファンクションキーとヘルプキーなどが押されたときの処理は追加コーディング可能となっている。画面制御（消去、カーソルアドレッシング、文字飾りなど）とプリンター制御（文字サイズ、改行巾、文字飾りなど）に関しては、その機能に決められたラベルのみをコーディングしておき、Easy-Partsの%^TCRTCを実行し、機種を選択するとESCシーケンスが自動コーディングされる。

2.2 Easy-Parts

部品の集まりであり、特殊処理、追加コーディングのために、次の外部プログラムを用意している。

(1) 入力用及びウィンドウ制御用 (DO呼出)

これらはEasy-Patternの変更不可セグメントで使われており、通常のプログラマは必要としない。

%ZJINS	編集入力処理（一行用）
%ZJINM	”（複数行用）
%ZFKEY	ファンクション表示処理
%ZSELPG	選択入力処理（ページング方式）
%ZSELRL	”（スクロール方式）
%ZWOPN	ウィンドウオープン処理（ウィンドウ番号の自動管理を含む）
%ZWCLS	ウィンドウクローズ処理（ ” ）

(2) 日付関係ユーティリティ (外部関数)

$\$\$\%ZDNOW(MOD)$ システム日付
 $\$\$\%ZDTOM(X, MOD)$ X日の翌日
 $\$\$\%ZDENDM(X, MOD)$ X日の月末日
 $\$\$\%ZDAFTD(X, Y, MOD)$ X日からY日後の日付
 $\$\$\%ZDAFTM(X, Y, MOD)$ X日からY月後の日付
 $\$\$\%ZDCNT(X, Y)$ X日からY日までの日数
 $\$\$\%ZDCHK(X, MOD)$ 省略日付を正式日付に変換 (一般用)
 $\$\$\%ZDCHKF(X, MOD)$ " " (先付け日付用)
 $\$\$\%ZDCHKP(X, MOD)$ " " (前付け日付用)
 *MOD="S" YYYYMMDD (Standard)
 "P" YYYY. MM. DD (Pointed)
 "J" GYY. MM. DD (Japanese)

(3) 時刻関係ユーティリティ (外部関数)

$\$\$\%ZTNOW(MOD)$ システム時刻
 $\$\$\%ZTCNVH(X, MOD)$ \$Hタイプからの変換
 $\$\$\%ZTCHK(X, MOD)$ 省略時刻を正式時刻に変換
 *MOD="AP" AM HH:MM または PM HH:MM
 "24" HH:MM (24-hour clock type)

(4) 年齢・曜日計算ユーティリティ (外部関数)

$\$\$\%ZAGE(X, Y, MOD)$ 誕生日(X)と基準日(Y)から年齢を返す。
 *MOD="Y" 年齢、"M" 月数
 $\$\$\%ZYOB1(X, MOD)$ X日の曜日と平日・休日を返す。
 *MOD="N" 数値(1~7)、“E” 英語(SUN~SAT)、
 "J" 日本語(日~土)、“H” 平日(W)・休日(H)

(5) 全角・半角変換ユーティリティ (外部関数)

$\$\$\%ZCNVH(X)$ 半角変換 テーブル $\%ZTDZH$ が必要
 $\$\$\%ZCNVZ(X)$ 全角変換 テーブル $\%ZTDHZ$ が必要

(6) ローマ字カナ変換ユーティリティ (外部関数)

$\$\$\%ZROMA(X)$ ローマ字をカナ文字に変換

(7) 数学関数 (外部関数)

$\$\$\%ZSINR(X)$ sin(ラジアン)
 $\$\$\%ZSIND(X)$ sin(度)
 $\$\$\%ZCOSR(X)$ cos(ラジアン)
 $\$\$\%ZCOSD(X)$ cos(度)
 $\$\$\%ZATAN(X)$ arctan(ラジアン)

\$\$^%ZEXP(X)	exp(X)
\$\$^%ZNEXP(X)	exp(-X)
\$\$^%ZEXP10(X)	10のX乗
\$\$^%ZLOG10(X)	常用対数
\$\$^%ZLOGN(X)	自然対数
\$\$^%ZROOT(X)	平方根

(8) その他

\$\$^%ZBCNT(X)	文字列のバイト数を返す。
\$\$^%ZFORM3(X, Y, Z)	X 数値 Y 1=3桁ごとカンマ付き、0=カンマ無し Z 丸め位置 (100 0.01など: default=1)
\$\$^%ZKHENO(X)	シノニム文字化 (ゞとズを同義語化など)
^%TCRTC	端末・プリンターのESCシーケンス自動埋め込み
^%ZPARAD	常用ローカル変数の設定(画面用ESCシーケンスなど)
^%ZPOPIN	ローカルプリンターへの切替え
^%ZPCLS	画面表示への切り戻し

3 結果と検討

一つの画面用のプログラムが、一本に納まり、プログラムが非常にコンパクトで、管理がしやすくなっている。東海大学医学部の予約システムのメイン画面の例では、ワークベンチによって作成されたプログラムは4本で、ファンクションキーとヘルプ処理用などのグローバルが作成され、4本プログラムのルーチンサイズの合計は約50KBであるが、Easy-Toolによって作成されたプログラムは一本でルーチンサイズは約20KBであった³⁾。

また、プログラムスワップとディスクアクセスが少ないため、実行スピードが高速化された。このため、端末のLAN直結化も全面的な効果として発揮されることがわかった。従来のシステム(端末はRS232Cの9600ボー接続)ではメイン画面からヘルプウインドウの表示に数秒要したが、新システム(端末はLANボードにより接続)では、ほぼ瞬間的に表示されるようになった。

今後機会があれば、Easy-Toolを他のエミュレータやWindows対応にもしていきたいと考えている。また、Share-ware化により、M言語の普及に貢献できればと考えている。

文献

1. 平野健一、横山幸次、中繁実、柳田彰男、清水裕史、松本昭、大櫛陽一、田久浩志、藤枝正秀、永井英保：大規模な病院情報システムの構築、日本MUMPS学術大会予稿集、17、32-32、1991。
2. 大櫛陽一：病院情報システムから地域医療システムまで—東海大学医学部・医学情報部—、医療とコンピュータ、5(3)、1-6、1992。
3. 高橋政志、小西千恵子、松本昭、大櫛陽一、尾崎恭輔、松山正也：東海大学病院における予約システムの開発について、病院管理、30(1)、124、1993。

M言語におけるオブジェクト指向環境 Objectoriented Environment in M language

今泉幸雄
Yukio Imaizumi

手続き言語の時代からあるM言語にも、最近オブジェクト指向環境が取り入れられている。M言語にオブジェクト指向環境を追加したEsiObjectsについて考察した内容を述べる。EsiObjectsと標準M言語の主な違いは、EsiObjectsはオブジェクト指向環境であるので、メッセージの文法、カプセル化、データ抽象化とオブジェクト・タイプ、ポリモーフィズム、継承等の機能がある。EsiObjectsの開発環境は、オブジェクトブラウザやユーティリティ形式のプログラム作成道具である。EsiObjectsにおける継承、ラベルや変数の有効範囲とメソッドの概念と、アプリケーションの事例についても述べるとともに、今後の課題にも言及する。

(キーワード: EsiObjects、オブジェクト、クラス、インスタンス、メソッド、継承)

In current , there are the function of object oriented environment in M language what was appeared in procedure language periods. I am going to write the EsiObjects with object oriented environment in M language. The EsiObjects Application Development environment is based on the OO(Object Oriented) and M language under Windows3.1 technology. Of course , the major difference between EsiObjects and M language is that EsiObjects is OO , has a special message syntax , enforces encapsulation, data abstraction and object typing , polymorphism , inheritance and etc. The EsiObjects development environment provides a full set of programming tools in the form of object browsers and utilities. I am going to explain the concepts of methods , method labels and variable scoping , within the concept of inheritance and the theme in the near future in EsiObjects. Inheritance is the mechanism that gives the OO paradigm its power as unifying application development tool.

(Keywords: EsiObjects, Object, class, instance, method, inheritance)

1 はじめに

オブジェクト指向の考えが最初に論文に発表されてから20年になる。最近ではプログラミング言語や人工知能だけでなく、ソフトウェア全体に影響を与えて、標準化もすすめられて実用化された製品が市場にでている。勿論、M言語もMDC(M Development Committee in USA)を中心に標準化が進んでいる。本稿は、Windows3.1版M言語上で動作するEsiObjectsの動作環境、EsiObjectsのオブジェクトの概念と継承、標準M言語との違い、アプリケーションの利用と今後の課題について述べる。

2 EsiObjectsの動作環境

2.1 EsiObjectsのスタートアップとシャットダウン

MS-DOS V5.0をスタートアップ後にDOSのプロンプトの状態、MWIN(MSWindows3.1とDTWindows)と入力すると、EsiObjectsV1.0のウィンドーが開かれる。ここで、EsiObjectsV1.0のアイコンをクリックすると、User Informationのウィンドーが開かれて、UserIDとPasswordを入力する。Application Launch Pad(ALP:EsiObjectsのアプリケーション開発環境)のウィンドーがあらわれ、File、Tools、Utilities、Options、Helpの5つのメニューバーがあらわれる。これよりEsiObjectsによるアプリケーション開発等が出来る。EsiObjectsのシャットダウンは、FileメニューのShutdownコマンドをクリックすると実行する。

2.2 EsiObjectsのアプリケーション開発環境(ALP)

ALPにはFile、Tools、Utilities、Options、Helpの5つのメニューバーがある。その中で主なHelp、File、Toolsについて述べる。Helpは教育用を含めて、OOP(Object Oriented Programming)に関するターミノロジー(オブジェクト、クラス、継承、カプセル化、インスタンス等)の解説とEsiObjectsの自習機能と操作中のヘルプ機能等から構成されてる。Fileは開発したアプリケーション・プログラムの実行、作成宣言、終了、保存、シャットダウン等から構成されてる。Toolsはクラス・ブラウザ、オブジェクト・ブラウザ、トリー、プロトコール、レイアウト、シェルから構成されてる。クラス・ブラウザ、オブジェクト・ブラウザは既にReady-madeされてるクラス、オブジェクト(一般のOOPのインスタンスと解釈するとよい)の参照機能、トリーとプロトコールはクラス等を階層構造にて表現する機能、レイアウトは再配置機能、シェルは直接にクラスやオブジェクトにメッセージ・パッシング(Message Passing : 一般のOOPではメッセージ通信と呼ぶ)をDOコマンドで実行する便利な機能である。本稿ではこのToolsの機能を利用して、EsiObjectsのオブジェクトと継承を述べる。

3 オブジェクト

オブジェクトは変数(データ)とメソッド(動作の記述)がカプセル化(一緒になってる)されてる。オブジェクトはクラス・オブジェクト(以後クラスと呼ぶ)とインスタンス・オブジェクト(以後インスタンスと呼ぶ)の2つに分けられます。ここではEsiObjectsにおけるOOPについて述べる。

(図1 : Browser for the class Object)

3.1 オブジェクトの定義

EsiObjectの最上位クラスをスーパークラスと呼び、"Object"と言う名前になっている(著者はこの名前は大変誤解を招くと思われる)。クラスは定義の世界であり辞書の役割をする。インスタンスは現実の世界であり実際に存在する物を対象とする。例えば哺乳類や犬とはクラスであり、Aさんの家のポチはインスタンスである。EsiObjectsでは多くのクラスがReady-Made(標準で準備されてる)で

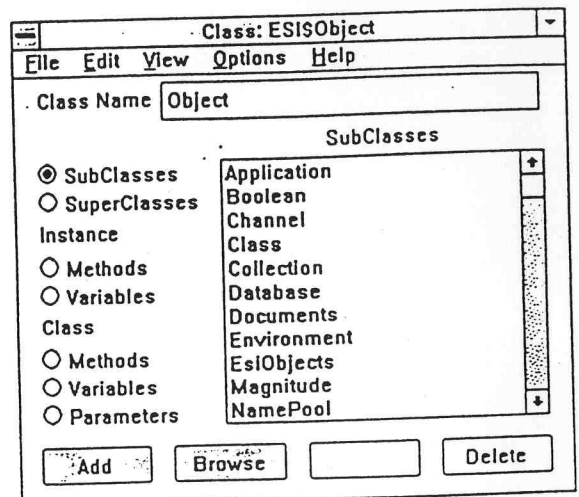


図1 Browser for the class Object

あり、利用者定義のクラスは全てスーパークラスであるObjectの下にリンクされる。インスタンスはクラスに実際の値が入って生成(定義)され、その時にOID(Object Identifier: オブジェクト識別)が自動的かつユニークな識別番号が貸与される。

3. 2 データの抽象化におけるクラスとインスタンス

例えば、Aさんのポチ、Bさんのラッシー、Cさんのケンタという犬の特徴をまとめると、"飼い主の犬"という特徴が抽出される。具体的には飼い主、名前があり、足が4本、人に懐く等である。ポチ、ラッシー、ケンタは、"飼い主の犬"というクラスのインスタンスである。ここで考えられるのはクラスとインスタンスの関係は1:nである。現実世界では複数のインスタンスからクラスを抽出する場合もあり(犬と飼い主の犬)、逆にクラスから複数のインスタンスを生成する場合もある(数学の定理と実際の方程式)。

3. 3 オブジェクトの階層構造

オブジェクト(クラスとインスタンスの両方を含む)は階層構造になっている。Objectと呼ばれるスーパークラスが一番上に存在して、その下位のクラスをサブクラスと呼ぶ。サブクラスの関係には上下関係である親クラスと子クラス、同じ親クラスから生成された子クラスの同等関係をピアクラスと呼ぶ。インスタンスはあるサブクラスから実際の変数(データ)に値等が入って生成されて、最も下位のクラスに階層構造上リンクする。

3. 4 クラス構造

クラスはクラス・メソッド、クラス・変数、クラス・パラメータの3つから構成されて、カプセル化(一緒になって)されてる。クラス・変数はデータで標準M言語の記述と同じであり、クラス・メソッドの中で使用される。クラス・メソッドはクラスの動作を記述するプログラムの処理部分で、標準M言語の記述と同じであり、クラスやインスタンスの生成・操作・削除等を記述する。クラス・メソッド、クラス・変数の両方ともブラウザをみると、階層構造上で、I(Inherited: 上位クラスから継承される)、S(Shared: 当クラスで定義してサブクラスで有効となる)、P(Private: 当クラスのみで定義して他には有効でない)の区別がある。クラス・パラメータはメッセージ・パッシング時に利用する。

(図 2: The SolutionApp class 図 3: Class and Variable Browser 図 4: The Tree Window for SuperClasses of SolutionApp 図 5: The Method Browser for [Start.SolutionApp])

3. 5 インスタンス構造

基本的にはクラスと同じ構成をとるが、インスタンス・メソッドとインスタンス・変数の2つから構成されて、カプセル化(一緒になって)されてる。クラスと各々の違いは、インスタンス・メソッドはインスタンスの動作のみに対しての記述であり、インスタンス・変数はインスタンス・メソッドの中でのみ使用される。従ってメソッドも変数も使用目的と有効範囲が異なるので、クラスのととは別々のものである。クラス同様にインスタンス・メソッドとインスタンス・変数の両方とも、階層構造上のInherited、Shared、Privateの区別はある。

3. 6 メッセージ・パッシングとポリモーフィズム

クラスからインスタンスの生成、インスタンス間の通信、インスタンスの消滅等の全ての動作はメッセージ・パッシング(メッセージ通信)にて実行される。EsiObjectsでは全てのオブジェクト(クラスとインスタンスの両方を含む)間のやりとりはメッセージ・パッシングにて実行される。記述形式は次のとおりである。

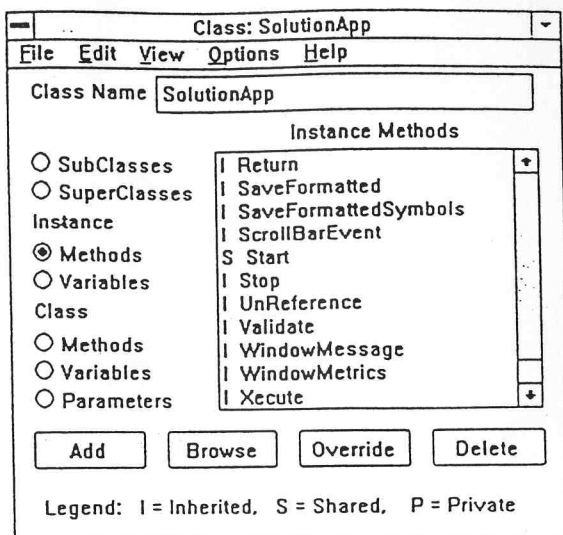


図2 The SolutionApp class

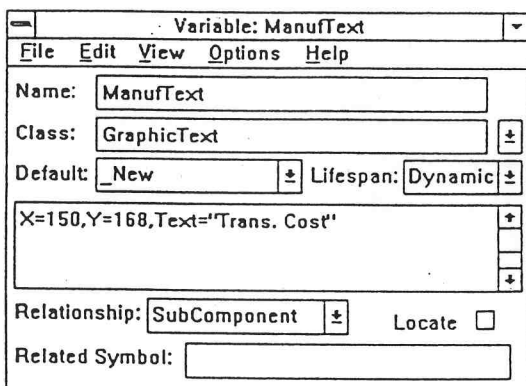


図3 Class and Variable Browser

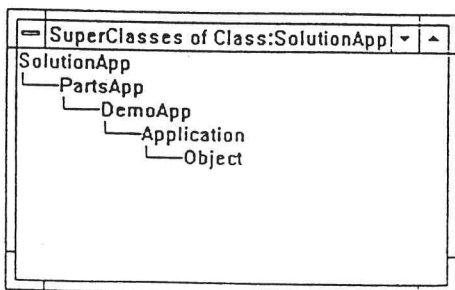


図4 The Tree Window for SuperClasses of SolutionApp

[Method. Object. Keywords](Parameter List)

Method :_(アンダースコア)で始まるクラスのメソッド名あるいは、インスタンスのメソッド名である。
 Object : 通信する相手のクラスの名前(アンダースコアで始まる)あるいは、インスタンスの名前である。
 Keywords : メッセージ・パシングの配布方法である。
 オプションであり使用されることは少ない。
 Parameter List : 相手に送る変数の値でありオプションである。
 基本はキー付きパラメータであり順序はない。
 Parameter1=Value1(, Parameter2=Value2,)
 Parameterの存在チェック(\$DATA)はメソッドが実施する。

事例としては

```
S PARTDB=[_New. _PartsDatabase](Rootnode="`xPARTS")
DO [Start. SAPP]
DO [Browse. _SolutionApp]
```

上記でおわかりのように、EsiObjectsは一般のOOP環境と同じくメッセージ・パシングはポリモーフィズムが確立してる。つまり異なるオブジェクト(クラスとインスタンスの両方を含む)を同一の通信方式(プロトコール)で動作させることが出来る。この柔軟さはプログラミングの再利用性等に大変メリットとなり、カプセル化されたオブジェクトにたいする機密性の保持にも有効となっている。

4 継承

メソッド、メソッドのラベル、変数の有効範囲の継承(インヘリタンス: Inheritance)について述べる。

4. 1 単一継承と多重継承

親クラスが持っているメソッド、メソッドのラベル、変数の内容を、子クラスが標準で参照可能であり、新たに子クラスで定義する必要のない場合(デフォルト)を継承と呼ぶ。例えば親クラスが自動車で、そのサブクラスにトラックと乗用車の場合に、自動車クラスで定義してるタイヤの数4つは、トラック・クラスと乗用車クラスはタイヤの数はデフォルトで4つになる。このように親クラスが一つのを単一継承(Single Inheritance)と呼ぶ。

子クラスから親クラスをみた時に、複数の親クラスがある場合を多重継承(Multiple Inheritance)と呼び、両方の親クラスのメソッド、メソッドのラベル、変数を継承する。

4. 2 メソッドとメソッドのラベルの継承

新しいクラスが生成される時に、親クラスのメソッドとスーパークラス(EsiObjectsでは"Objects")のメソッドが継承される。インスタンスが生成される時に、生成するクラスのメソッドとスーパークラスのメソッドが継承する。

メソッドのラベルはEsiObjectsの形式になっていて、メソッドと同じく継承される。

* {Label}^{Vector}

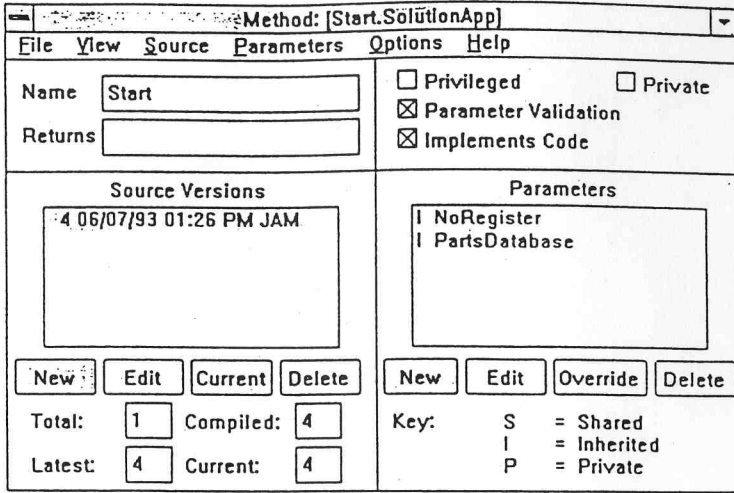


図5 The Method Browser for [Start.SolutionApp]

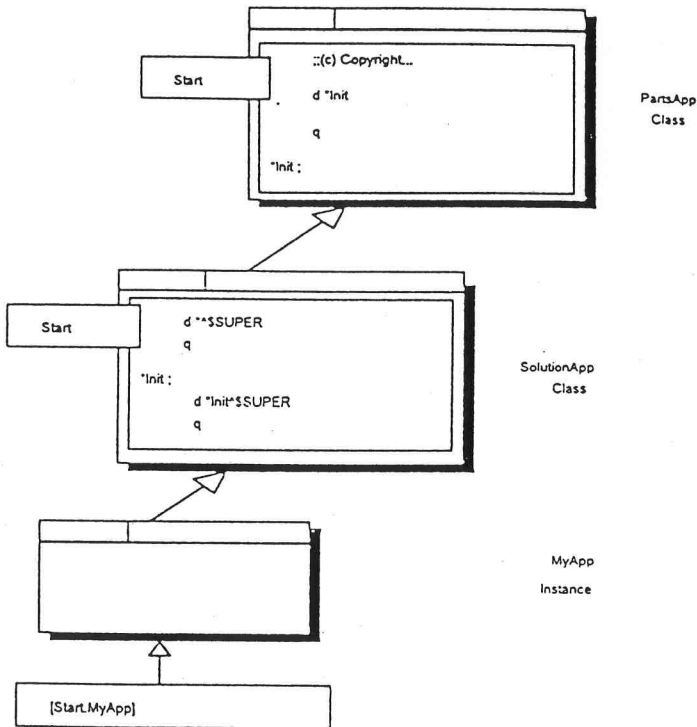


図6 Example of Method and Label Inheritance

*: 上位クラスから継承しての意味であるパブリック(Public) ラベルをあらわす。

*が無いラベルは 標準M言語と同じプライベート(Private) ラベルをあらわす。

Label: 標準M言語と同じである。

Vector: メソッドの探す階層構造の位置を示す。

事例としては

```
DO *init
```

```
DO *init~$SUPER ($SUPERはEsiObjectsの特殊変数)
```

(図6: Example of Method and Label Inheritance 図7: The Source Browser Window)

4. 3 変数の継承

標準M言語と同じく、変数の定義はSET命令です。

```
SET Name=Value
```

定義された名前と値はシンボルテーブルに格納されて、オブジェクト(クラスとインスタンスの両方を含む)が生成された時に付与されるOIDと組になる。

```
SET Name=[_New._ObjectClass](Name="ObjectA")
```

ObjectClassからObjectAという名前のオブジェクトを生成(Newメソッド)する。

これから変数の有効範囲について述べる。標準M言語の仕様になっている、ジョブ(Job)範囲有効なローカル変数(Local Variable)とファイル等で使用するグローバル変数(Global Variable)の2種類がある。また1990年度標準M言語仕様、NEWコマンドのローカル変数も含めて、EsiObjectsのオブジェクト(クラスとインスタンスの両方を含む)とメッセージ・パッシングを含めると次のような変数有効範囲図が描ける。

変数Name はEsiObjectsの有効範囲から命名方法を拡張してる。

```
SCOPE%NAME(S1, S2, ..., Sn)
```

SCOPE: 変数の有効範囲であり、英字一文字である。

T: Temporary 単にメソッドの呼びだしに使用

P: Parameter メッセージ・パッシングで使用するParameter Listの変数

I: Instance インスタンス・変数

C: Class クラス・変数

U: Universal NEWコマンドが仕様になる前の有効範囲と同じ。

全オブジェクト(クラスとインスタンスの両方を含む)に有効。

N: Named Pool スーパークラス(Object)の下位のクラスに保存する。

(図8: EsiObjects Object Oriented Execution Context 図9: Scoping of Variables)

4. 4 インスタンス変数のライフスパンと格納場所

インスタンス変数のライフスパン(Lifespan)は動的(Dynamic)と静的(Static)の2種類ある。クラスからインスタンスが生成される時に選択される。動的は参照時に作成されて後に変更可能な値であり、画面の高さ・広さ等の変数に利用される。静的は全てのインスタンス生成時に作成されて変更不可能な値であり、タイトル等の変数に利用される。

インスタンス変数を指定する事により、特別な場所に格納出来る。勿論グローバル・ファイルであり、NAME(S1,S2,...,Sn)の表現となる。例えば^VESPARTS("TEST"),PARTS("TEST")となる。

(図10: The Locate Button and Location Fields)

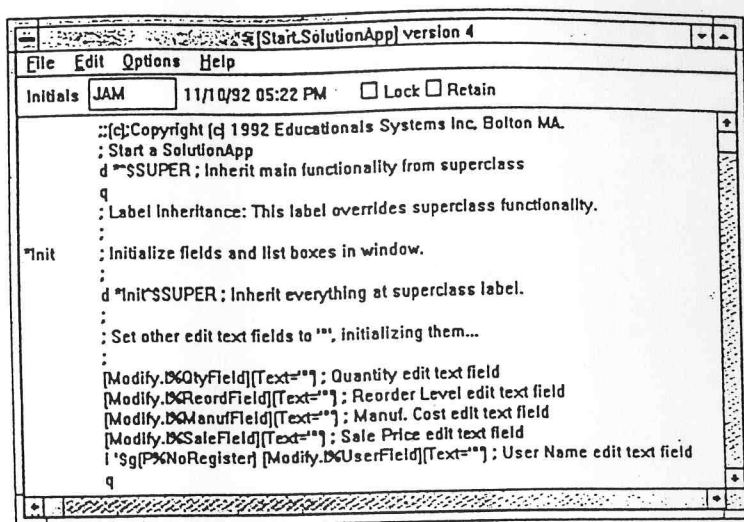


図7 The Source Browser Window

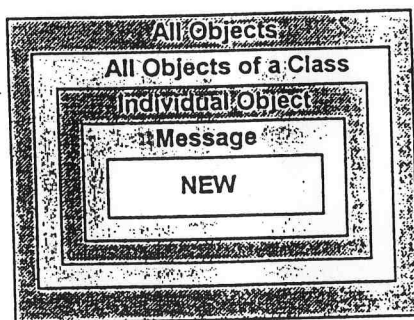


図8 EsiObjects Object Oriented Execution Context

	Sub-routine	Message	Object	Objects of a Class	All Objects
New (1990 New'ed Local)	X				
Parameter	X	X			
Temporary	X	X			
Instance	X	X	X		
Class	X	X	X	X	
Universal (Traditional un-New'ed local or global)	X	X	X	X	X

図9 Scoping of Variables

4. 5 特別変数と特殊関数

EsiObjectsがOOPを導入するにあたって、20種類の特別変数(Special Variables)と13種類の特殊関数(Functions)を追加されている。

5 EsiObjectsと標準M言語の主な差異

すでに必要に応じてEsiObjectsの拡張部分を述べてるが、ここではプログラミングに関係する部分について述べる。NEWコマンドは、ローカル変数にだけ使用可能であり、Temporary,パラメータ,インスタンス変数等では使用出来ない。

NEW X,Y,ZはNEW L%X,L%Y,L%Zは同等でありLはローカル変数を意味する。KILLコマンドは、引き数無しの場合禁止される。ポインターは、間接的指定のポインター指定が標準M言語で多く用いられるが、EsiObjectsはメッセージ・パッシングによりOIDで実施できる。間接的指定はメソッドの名前、Parameter list、XECUTEコマンド等にて利用可能である。NakedReferenceはEsiObjectsでは使用出来ない。

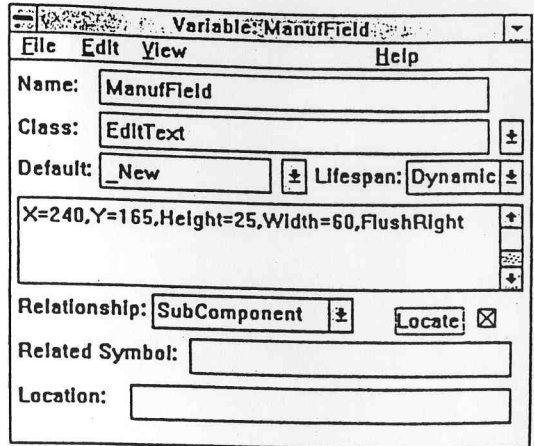


図 10 The Locate Button and Location Fields

6 アプリケーションの利用

```
Object ---Database ---- DemoDatabase --- PartsDatabase
```

```
      |                                     |=====>PARTSDB
      + Application--- DemoApp ----- PartApp -----SolutionApp
```

```
|=====>SAPP
```

```
-MyaApp
```

```
----- クラス関係 =====> インスタンスを意味する。
```

事例として、部品倉庫管理のPartsAppクラスとSolutionAppクラスが提供されていて、利用者がPartsAppクラスからMyAppクラスを生成したとする。MyAppクラスは生成時に、親クラスからメソッド、変数、クラス・パラメータが継承されるが、MyAppクラスでのShared,Privateのは新たに定義する。

```
S PARTSDB=[_New._PartsDatabase](RootNode="^xPARTS")
```

PartsDatabaseクラスにメッセージ・パッシング(NEWメソッド)にて、PARTSDBというインスタンスを生成した。データベースとしてはRootNodeにて定義する。

```
S SAPP=[_New._SolutionApp](PartsDatabase=PARTSDB)
```

SolutionAppクラスにメッセージ・パッシング(NEWメソッド)にて、SAPPというインスタンスを生成した。使用するデータベースはPartsDatabaseのと定義する。

```
D [Start.SAPP]
```

SAPPインスタンスにメッセージ・パッシング(Startメソッド)にて起動する。

7 考察のまとめ

EsiObjectsは、標準M言語経験者がオブジェクト指向について習得しやすい。それはオブジェクト指向についての自主学習機能がついていて、マニュアルが標準M言語と比較する記述例(特にメソッドのプログラミング)が多いからである。2番目は、ツールが完備されてる開発環境である。オブジェクト(クラスとインスタンスの両方を含む)の利用者による追加・変更・削除がマウスのウインドー操作で大変容易にできる。また、変数やメソッドの継承がトリー図(階層構造)により大変理解しやすい。3番目は、メソッドのプログラミングがしやすい。Ready-madeのオブジェクトのメソッドのプログラムが公開されていて、オブジェクト指向の特別変数と特殊関数が豊富である。最後は、欠点としてマニュアルの事例の詳細性に欠ける。例えば部品倉庫管理(PartsApp)の事例において、業務の前提条件やデータ内容等に不足がありインスタンスを生成するとき条件等に理解しにくい。

8 おわりに

従来M言語の特徴を生かしながら、オブジェクト指向の開発環境を取り入れたEsiObjectsの概念と実際のクラスやインスタンスの生成、起動、消滅とプログラムの再利用性について、調査した結果について報告しました。今後はEsiObjectsがDTMだけでなく、他のM言語での動作とM言語間でのオブジェクトの流通性が高くなれば良いと思っています。

[謝辞]今回M言語におけるオブジェクト指向環境を考察するにあたって、ハードウェアとソフトウェア(EsiObject,DTM)を貸与してくれた、日本ダイナシステム(株)の嶋氏と鈴木氏に感謝します。

参考文献

- 1 "EsiObjects Getting Started Guide" (1993)
- 2 "EsiObjects User's Guide" (1993)
- 3 "EsiObjects Programmer's Reference Guide" (1993)
- 4 今泉幸雄 "MUMPSによるオブジェクト指向型データベース管理システムへの考察", 第19回日本MUMPS学会大会, pp71-76, 1992
- 5 今泉幸雄 "これならわかるオブジェクト指向(手続き言語時代の技術者を迷わしたオブジェクトとは何か?)", 第20回日本MUMPS学会大会, 1993

M 環境と外部世界との融合の試み Attempts to integrate M environment and outer worlds

佐藤比呂志
Hiroshi Sato

M 言語が提供するソフトウェア開発環境は、それだけで大抵の実用システムを作成できるほど強力である。いままでM 環境は、必要とされる機能を順次自分自身の環境に取り込む努力を続けてきた。M 言語の使いやすさ、実用性は、これらの努力によるところが大きい。

しかしながら、昨今のコンピュータ環境の急激な進歩に対して、M 環境の対応の遅れが目立ってきた。

M が今後とも発展するためには、従来のように何でもM 環境に取り込もうとするのではなく、M 以外の世界との関係を深め、共存するための努力が求められてきているのではないだろうか？

その様な観点から、現在弊社で試みているM 環境と外部世界とを結び付けるいくつかの試みを紹介させていただきたい。

(キーワード：M 環境、M 言語、外部世界)

Software environments that are provided by M language are strong enough to establish most working systems by them own. We have been making efforts to integrate M environments and required capabilities step by step. It can be said that usability and practicability of M environments are mainly due to the efforts.

M environments, however, are significantly behind with recent rapid progress of computer environments.

In order for M to continue to develop, conventional efforts for integrating anything into M environment are not appropriate. It may be required to make efforts to have deeper relationship with worlds other than M and to coexist with them.

The following introduces some of the attempts which are made in order to link M environments and outside worlds together by our company.

(Keywords: M environment, M language, outer world)

1 DSM API for Windows

高性能PCの普及に伴い、MS-WINDOWSインタフェースを使用したアプリケーションを使用したい、開発

日本デジタルイクイップメント株式会社
西日本第一統合システム部
〒530 大阪市中之島2-2-2ニチメンビル
TEL 06-222-9211 FAX 06-222-9408

Digital Equipment Corporation Japan Digital Consulting
Nichimen Building, 2-2-2, Nakanoshima, Kita-ku, Osaka 530,
Japan
TEL 06-222-9211 FAX 06-222-9408

したいという要求が強くなってきた。

そこで、既存のDSM 環境にPCのMS-WINDOWS環境からアクセスするためのネットワーク・ツールを開発した。

そのツール名は、DSM API for Windowsである。

このツールを使用してPC 側からコマンド実行、グローバル、ローカル変数へのアクセス、ルーチンの実行、M 関数の実行をDSM サーバーに依頼することが可能になる。

このツールとPC 側のツール(Visual Basic等)を組み合わせることにより、MS-WINDOWS のGUIを駆使したDSM アプリケーションを作成することができる。

このツールの利点

- 低コストでGUI M アプリケーションを作成できる。
- 習得が簡単
M 言語をご存知の方であれば、このツールが提供するAPIを理解するのは、とても簡単である。
- DSM のほとんどの機能を利用することができる
- クライアント・サーバー・コンピューティング

このツールの弱点

- データの引き渡し
この問題は、このツールに限らず、M 環境と他環境を結合する時にいつも問題となる。
M 言語と他の言語でデータを引き渡す場合問題となるのは、M 言語の変数構造が他言語、他環境には理解できないという点である。この制限のため、引き渡せるデータは、単純な文字列や数値データに限定され、M の高度な構造をもった複雑なデータを引き渡すことができない。
- PC 側に M 以外のツールが必要
M 言語以外にVisual Basic等の習得が必要になる。

2 SQL/DSM

この製品を使用することにより、PC上の様々なツール(MS-Access MS-EXCEL, Visual Basic 等)から他のリレーショナル・データベースと全く同じ様にDSM のグローバル変数にアクセスすることができる。このことは、M を全く知らないエンドユーザーが、DSM グローバルを様々な角度から取り出すことができることを意味し、DSM 環境で、EUC(End User Computing)の道をひらくことが可能となる。

このツールの利点

- 様々なPC ツールからDSMデータベースを利用可能
- EUC が可能
- クライアント・サーバー・コンピューティング

このツールの弱点

- M 特有の機能は使用できない。
M のことを意識せずにアクセスできるということは、逆にM の特色をだすことが難しいことを意味し、他のリレーショナル・データベースと差別化が難しくなる。

3 MIDDLE (M Interface Data Definition Language Extension)

M環境と他環境でデータを交換する場合の現状の問題点は、スカラーデータのみでの交換に限定されており、配列やより複雑な構造をもったデータの交換ができない点である。また、他の言語に存在するデータ・タイプの区別もM言語にはないので、データ・タイプを意識したデータ交換もできなかった。

この問題を解決するためのメカニズムであるMIDDLEを現在開発中である。

MIDDLE 開発の目的とゴール

- DSMとC言語間のシームレスな相互データ交換を実現する。
- DSMと既存のC言語で記述されたプログラムまたは、ランタイム・ライブラリとの結合を可能とする。
- データを交換するための特別なルーチンを介在させることなく実現する。
- C言語で記述された関数と同じ様にDSMのルーチンを呼び出すことを可能にする。

MIDDLE とは

インターフェース定義言語を提供する。

- C言語の複雑なデータ構造をM言語のデータ構造にマッピングする。
- MルーチンとC言語の関数プロトタイプをマッピングする。
- 既存の外部呼び出し機構(ECALL)とCallable interfaceを拡張したもの。

MIDDLE によって可能になること

DCE RPCとM環境との結合

ORB(Object Request Broker)とM環境との結合

標準ランタイム・ライブラリとの結合

既存のC言語パッケージソフトウェアとM環境の結合

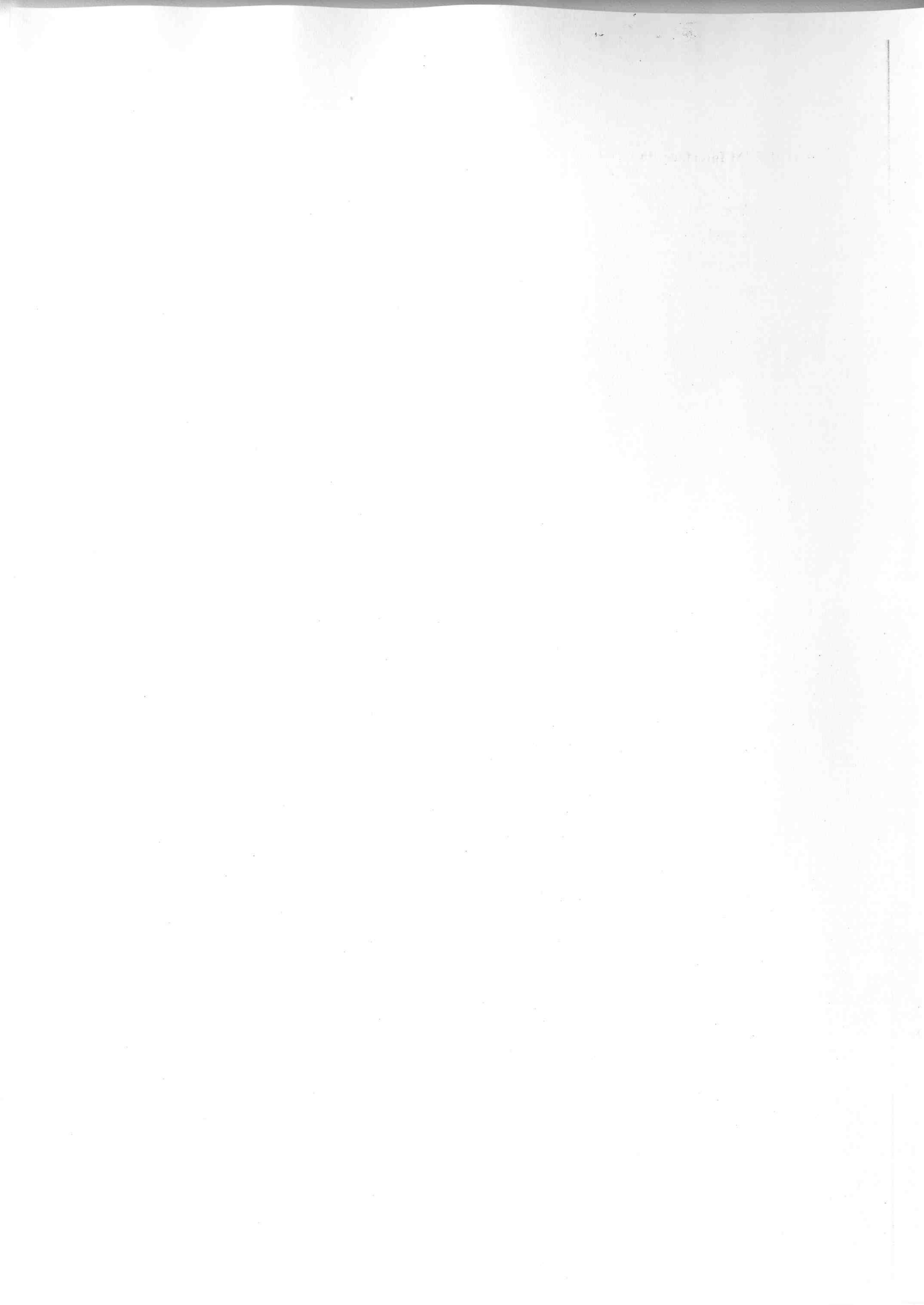
4 おわりに

Mを取り巻く外部環境は、日々刻々と変化している。これらの変化にMは遅れることなく追従していくことが求められている。

今回紹介した試みは、それを解決するためのほんの一例にすぎないし、すべての問題を解決するものでもない。

今後ともM環境と今後重要になるであろうオブジェクト・モデルに基づいた新しい外部世界とをどの様に融合すべきかなど考えていかなければならない問題がいくつかある。

また、その様な融合された環境でのMの強み、メリットが何であるのかをはっきり打ち出せなければいけないのではないだろうか。



第21回日本エム・テクノロジー学会大会を振り返って

第21回日本エム・テクノロジー学会大会 大会長

岡田 好一

1 大会の概要

第21回日本エム・テクノロジー学会大会は、1994年8月15日から3日間、茨城県つくば市、筑波大学医学専門学群で開催されました。参加者は、154名です。13の企業の展示、初級・中級のチュートリアル、ワークショップ、海外からの4名のお客様を含む10の特別講演・招待講演、15の一般演題の発表がありました。会場は3日間とも盛況でした。これも学会員、参加企業の皆さまの努力のたまものと、大会長としてお礼を申し上げます。

2 第21回大会の目標

大会を始めるに当たり、私なりの目標を設定しました。

本年2月に、M言語は世界初の国家規格である、JISとなりました。昨年は、JISの秒読みの段階でしたので、この際、ぜひ一般にもアピールする大会をと考えました。

幸い、東京から1時間、しかも日本の名だたる企業の研究所が立ち並ぶ、筑波の地の利がありました。そこで、まず、いままでM言語を知らない人にも気軽に参加できる雰囲気作りを心掛けました。具体的には、企業展示の充実とチュートリアルの雰囲気作りです。

実は、大会を開催してから気付いたのですが、大会会場の臨床講義室の前のロビーの入り口は、封鎖することができません。ですから、結果として、上述の参加人数に入らない、企業展示のみをご覧になった方も多いと思います。企業展示がずいぶん、にぎやかだったので不思議だったのですが、そのような理由があったのだと思います。つまり、偶然ですが、オープンな雰囲気作りができたと思います。

対外的には、「オープン」がM言語に対する魅力・安心感となると思います。つまり、M言語という生産性のきわめて高いデータベースが、他のシステムと簡単につながる、という点です。

M言語をよく知っている者は、M言語ですべて可能なのに、と考えがちです。実際のところ、それは間違っていない。しかし、新しく参入する方にとっては、このM言語の孤高の姿勢は、少なからぬ障害となります。

最近のM言語は、外部との接続が簡単になりました。このことは、M言語がほかの世界からの参加を歓迎している、という姿勢と映るでしょう。逆に、我々M言語の信奉者が、より広い世界に参画できる、ということでもあります。

3 大会を終えて

現在、私は東海大学にいます。筑波でエム・テクノロジー学会を開催できる、千載一遇のチャンスを皆さんは生かしたことになります。私も楽しく大会長を勤めさせていただきました。ありがとうございました。

第21回日本エム・テクノロジー学会大会

日程 1994年 8/5 - 8/7
会場 筑波大学医学専門学群 臨床講義室B
大会長 岡田好一 (筑波大学)
大会副会長 木村一元 (独協医科大学)

プログラム委員会

委員長 木村一元 (独協医科大学)
委員 今泉幸雄 (サンド薬品株式会社)
委員 嶋 芳成 (日本ダイナシステム株式会社)
委員 田久浩志 (東邦大学)
委員 本多正幸 (千葉大学)

M言語フェア実行委員会

委員長 小倉 勉 (住友電気システムズ株式会社)
委員 今井敏雄 (日本デジタルイクイップメント株式会社)
委員 松本重雄 (日本MSM株式会社)
委員 山本純史 (日本ダイナシステム株式会社)

協賛

住友電気システムズ株式会社
日本ダイナシステム株式会社
日本デジタルイクイップメント株式会社
株式会社 高崎共同計算センター

第21回日本エム・テクノロジー学会大会プログラム

8月5日

M言語初級講習会

日本エム・テクノロジー学会

特別講演(1) 司会 岡田好一 (筑波大)

高田彰 (筑波大)

筑波大学附属病院における病歴資料・情報の共有化と情報システム

ワークショップ

今泉幸雄 (サンド薬品)・岡田好一 (筑波大) : 進行役

オブジェクト指向とM言語

8月6日

一般講演(1) 大学病院 座長 田久浩志 (東邦大)

鈴木隆弘 (千葉大) 他

千葉大学医学部附属病院における、病床管理システムの導入について

山下芳範 (福井医大)

福井医科大学新総合病院情報システムにおけるM言語の利用と概要

笹川紀夫 (島根医大) 他

島根医科大学教育研究システムの概要とシステム運用について

土田伸 (西川町役場) 他

西川町総合保健医療福祉サービスにおける地域保健・医療・福祉行政

総合支援システムの構築

一般講演(2) 新世代M言語 座長 鈴木利明 (日本ダイナシステム)

今井敏雄 (日本DEC)

Mとクライアント/サーバーコンピューティング

前海好昭 (日本DEC)

Open環境で動作する新世代DSM

今泉幸雄 (サンド薬品)

EsiObjectsの概要

海外報告(1) 司会 木村一元 (独協医大)

山下芳範 (福井医大)

北米MTA報告、MDC報告

特別講演(2) 司会 山本和子 (島根医大)

里村洋一 (千葉大)

M言語によるクライアント・サーバーシステム構築

総会

司会 河村徹郎 (鈴鹿医療科学技術大)

特別報告 大櫛陽一 (東海大): 世話人

JIS原案作成について

特別講演(3) 司会 河村徹郎 (鈴鹿医療科学技術大)

若井一朗 (マンプス研)

M言語JIS化のインパクト

招待講演 司会 里村洋一 (千葉大)

M.アマラル (千葉大)

M Technology in Brazil: Status Report

プロダクト紹介

J.Jhung (OpenTech)

E.K.Pape (InterSystems)

DTM Product Directions

R.P.Mappes (Micronetics)

MUMPS Market Assessment and Opportunity

8月7日

一般講演(3) Mツール(1) 座長 笹川紀夫 (島根医大)

馬場謙介 (国立埼玉病院)

異巾文字列処理の考察

田久浩志 (東邦大)

GUIツール、MGRの概要と紹介

馬場謙介 (国立埼玉病院)

文字データの検索に関する考察

大櫛陽一 (東海大) 他

新開発環境Easy-Toolの開発

一般講演(4) Mツール(2) 座長 藤江昭 (住友電気システムズ)

馬場謙介 (国立埼玉病院)

経皮的冠動脈拡張術(PTCA)施行症例データベースに使用した

データなどの読み込みに関するSmall Concepts

入院病歴管理システム・経皮的冠動脈拡張術施行症例データベースに

応用した呼び出し機能を持つ編集可能疑似READ命令

波川智亮 (日本ダイナシステム) 他

MS-WindowsとDTM環境におけるGUI開発方法の比較

佐藤比呂志 (日本DEC)

M環境と外部世界の融合の試み

木村一元 (独協医大) 他

獨協医科大学病院薬剤部、薬剤管理支援システム

特別企画勉強会 (M言語システム中級)

住友電気株式会社

日本デジタルイクイップメント株式会社

日本ダイナシステム株式会社

「日本Mテクノロジー学会」ご入会のご案内

日本Mテクノロジー学会（日本MTA）は、M言語（MUMPS）の利用・改良・普及を目的とした団体で、個人や法人が加入して活発な活動を行っております。M言語はANSIにFORTRAN及びCOBOLに続いて3番目の標準コンピュータ言語として制定され、米国連邦情報処理標準言語にも採用されました。さらに1992年5月にはISO標準言語、1995年2月にはJIS標準言語として制定されるに至っております。一方、近年のコンピュータのダウンサイジングの流れにあって、ユーザーも着実に増えつつあります。

日本MTAは先に述べたような目的に向けて種々の活動を続けておりますが、貴方にも、是非とも日本MTAに参加し活動を盛り上げて頂きたいとご案内申し上げる次第です。

A. 日本MTAの活動

- 1) 年次学術大会、研究会や講習会の開催
- 2) M言語に関する技術情報の提供
 - PC通信Nifty-SERVE上にMUMPSフォーラムを設置
 - MTAニュースの発行
 - 各種資料の配布
- 3) 学術雑誌「Mumps」の出版
- 4) M言語改良仕様の検討・・・米国M Development Committeeと連携
- 5) 国際MTA、各国MTA（MUG）との交流
- 6) M言語のJIS改訂
- 7) ソフトウェアの公開流通

B. 会員の特典

会員になることにより次のような特典が考えられ、充分満足頂けるものと考えられます。

*個人会員の特典

- 1) 日本MTA年次大会、M言語関係学術集会、研究会、講習会のお知らせ
- 2) 日本MTA主催の学術集会、研究会、講習会などの参加費用の割引
- 3) M言語に関する各種資料の実費提供
- 4) 流通、ソフトウェア（MTAPAL）の低額頒布
- 5) 「MTAニュース」の無料配布
- 6) M言語ベンダーの折々のプロダクツ紹介・パンフレット・カタログ類の頒布
- 7) 雑誌「Mumps」の無料配布

・上記の各種活動を通じて、M言語に関する全世界の最新の技術情報が得られます。

*法人会員の特典

法人会員は「日本MTAの目的に賛同する法人で、日本MTAの目的を遂行するために積極的に事業を後援する事を表明した者とし、正副各1名の代表者を登録し、正副代表者とも個人会員と同等の資格を持つ」こととなります。尚、正副代表者には正会員と同様の日本MTAの役員としての道があります。

- 1) 日本MTA主催の集会には5名迄、会場費、講習会費などを会員割引
- 2) 日本MTA主催の医療人、企業人を対象とする講習会へ法人会員から優先的に出講
- 3) 日本MTA主催の集会への出品、展示に関する料金の割引
- 4) 日本MTA学術大会論文集、MTAニュース等への広告費の割引
- 5) 法人会員のプロダクトのパンフレット、カタログ類の会員への頒布
- 6) ユーザー法人にはM言語ベンダーないしシステムエンジニアの紹介
- 7) 日本MTAの流通パッケージ(MTAPAL)を割引料金で利用
- 8) MTAニュースを単なる広告ではなく、新しいプロダクツの紹介等の質の高いPRのために利用可能

注意) 法人会員は、国際MTAが設けている施設会員と企業会員に相当するものですが、学校法人・国立施設など税法上非営利団体扱いの法人を非営利法人とし、国際慣例よりも40%低い基本会費を申し受けます。その他は企業法人ないしベンダー法人としての会費を申し受けます。ご入会の手続きは「法人会員入会申込書」によってお願い申し上げます。

- ・上記の各種活動を通じて、M言語に関する全世界の最新の技術情報が得られます。
- ・M言語ユーザ間、M言語を取り扱うベンダー・メーカー間とのコミュニケーションが充実します。

C. 会費

ア) 個人会員

入会費 ￥4,000.

年会費 ￥6,000.

イ) 法人会員

入会費 ￥10,000. (営利・非営利法人共通)

年会費 ￥50,000. (1口) ←営利法人

￥30,000. (1口) ←非営利法人

注意) 会計年度は、毎年4月1日から翌年3月31日までです。

D. ご入会手続き

- 1) 入会資料請求<電話・FAX・郵便>
- 2) 事務局から送付された「会員登録票」(法人会員の場合は正・副代表者の「会員登録票」及び「法人会員申込書」)に必要な事項を記入の上、事務局までお送り下さい。
- 3) 郵便払込、銀行振込で入会金、年会費を事務局に納金して下さい。
- 4) 事務局から会員登録完了通知と会員証、資料他をお送りします。
(入会日は入会費・年会費納入日です。)

E. 入会費・年会費お支払方法

日本MTA事務局より会費の請求がございましたら、以下の何れか方法でお支払下さい。但し、お手数料は振込人払いとさせていただきますことをご了承下さい。

ア. 郵便振替 口座番号：01440-8-4520
 加入者名：日本Mテクノロジー学会

イ. 郵便貯金 口座番号：15330-12879211
 加入者名：日本Mテクノロジー学会
 事務局代表 山本 和子

ウ. 銀行口座 口座：山陰合同銀行 出雲支店
 口座番号：普通口座 027-2904176
 名義人：日本Mテクノロジー学会
 山本 和子(やまもと かずこ)

「日本Mテクノロジー学会」規約

第一章 総 則

第1条 本会は日本Mテクノロジー学会 (M Technology Association of Japan)という。

第2条 本会の事務所は幹事会の承認を経て、学会長が指定するところに置く。

第二章 目的および事業

第3条 本会は「M言語」並びにこれに関する情報システムの利用、応用、改良、並びに普及を行うことを目的とする。

第4条 本会は前条の目的を達成するため次の事業を行う。

- 1) 学会大会、フェア、研究会、講習会などの開催
- 2) 学会誌、ニュースなどの刊行物の発行
- 3) M言語の日本語装備の標準化
- 4) M言語の標準装備の監視
- 5) 海外のMTA (MUG) などとの連携活動
- 6) 内外の関連諸学会との連絡ならびに協力活動
- 7) M言語利用技術の相互交換の促進、本会に提供された資源の整備、管理ならびに会員への還元
- 8) その他目的達成のために必要な事業

第三章 会 員

第5条 本会会員は個人会員と法人会員からなる。

- 1) 個人会員は本会の目的に賛同し、本会の対象とする領域、又はそれと関連する領域において活動する個人とする。
- 2) 法人会員は本会の目的に賛同する法人で、本会の目的を遂行する為に積極的に事業を後援する事を表明したものである。法人会員においては正副各1名の代表者を登録するものとする。正副代表者は個人会員と同等の資格を有する。

第6条 本会に入会を希望する者は所定の申込書に入会金及び会費を添えて本会事務所に申し込まねばなら

ない。

第7条 本会会員は、毎年所定の会費を前納しなければならない。

第8条 本会会員で住所変更のあったものは速やかに住所変更届を、また退会しようとするものは退会届を本会事務所に提出しなければならない。本会会員で、住所不明となるか催促にも拘らず2か年を越えて会費納入遅滞のあったものは退会の扱いを受ける。物故会員は退会の扱いを受ける。

第9条 本会の規約に背く行為のあった会員は、幹事会の議決を経てこれを除名することができる。

第四章 役員その他

第10条 本会に次の役員を置く

1) 学会長	1名
2) 日本Mテクノロジー学会大会長(以下「大会長」という)	1名
3) 日本Mテクノロジーフェア実行委員長(以下「フェア実行委員長」という)	1名
4) 幹事 庶務財務担当	1名
国際担当	1名
流通担当	1名
広報担当	1名
雑誌担当	1名
ネットワーク担当	1名
M言語標準化担当	1名
J I S ・ I S O担当	1名
5) 会計監事	1名
6) 評議員	若干名

第11条 各役員の選出または構成を次のように定める。

- 1) 評議員に欠員が生じた場合、学会長は評議員会の推薦者を総会に諮り、その承認を得て決定する。評議員の定数は学会長が定める。但し、各評議員の構成割合は会員の職域構成割合に近いものとする。
- 2) 学会長及び会計監事は、評議員会の推薦者を総会に諮り、その承認を経て決定する。
- 3) 幹事は学会長が推薦し、総会の承認を経て決定する。学会長と幹事は併任できない。
- 4) 大会長は学会長が幹事会の推薦者を総会に諮り、その承認を経て決定する。
- 5) フェア実行委員長は学会長が幹事会の推薦者を総会に諮り、その承認を経て決定する。

第12条 各役員の任務は次のように定める。

- 1) 学会長は会を代表し、総会、幹事会、評議員会の議長となる。
- 2) 大会長は、年次日本Mテクノロジー学会大会を総括する。
- 3) フェア実行委員長は、年次日本Mテクノロジーフェアを総括する。
- 4) 庶務財務担当幹事は、本会に関する庶務及び全ての資金及び財産の管理を行う。また、最新の名簿の管理、総会その他の議事録の管理を行う。
- 5) 国際担当幹事は、海外のMTA (MUG) 組織との連携並びにM言語開発委員との協力を司り、その他の国際的協力を行う。
- 6) 流通担当幹事は、M言語応用プログラムのユーザー間相互交換の促進、MUGプロトタイプ・アプリケーション・ライブラリー (MUGPAL) などM言語資源の整備、管理、維持、会員に対する資料提供等のサービスを行う。
- 7) 広報担当幹事は、Mテクノロジーニュース等を通じ広報活動を行う。
- 8) 雑誌担当幹事は、学会誌「Mumps」の編集を兼ね、出版の進行を司る。
- 9) ネットワーク担当幹事は、ネットワークを活用した会員間のコミュニケーションの向上を図る。
- 10) M言語標準化担当幹事は、M言語の標準化を図る。
- 11) ISO・JIS担当幹事は、M言語のISOとJIS標準制定に関することを司る。
- 12) 会計監事は、年次会計の監査を行い総会に報告する。

第13条 各役員の任期を次のように定める。

- 1) 学会長、幹事、会計監事の任期は、4月1日より翌々年3月31日までの2年間とし再任を妨げない。
- 2) 大会長の任期は、前学会終了時に始まり学会の残務処理の終了までの期間とする。
- 2) フェア実行委員長の任期は、前Mテクノロジーフェア終了時に始まりMテクノロジーフェアの残務処理の終了までの期間とする。
- 3) 評議員の任期は特に定めないが、4年間続けて評議員会に出席しなければ評議員資格を失う。

第五章 会議および委員会

第14条 (総会)

- 1) 総会は本会の最高の議決機関である。
- 2) 総会は学会長が毎年1回召集する。但し、幹事会の議決による場合または会員の5分の1以上から請求された場合、学会長は臨時総会を召集しなければならない。
- 3) 総会の議長は学会長とする。
- 4) 次の事項は総会に提出してその承認を受けなければならない。
 - a. 事業報告および収支決算
 - b. 事業計画および収支予算
 - c. その他幹事会が必要と認めた事項
- 5) 総会の成立に必要な出席者数は会員のうち50名または10%の少ない方を上回る数とする。
- 6) 総会の議決は本規約に別に定めるものの他、出席会員の過半数による。

第15条 (幹事会)

- 1) 学会長が必要に応じて召集する。但し、幹事の過半数から請求があった時は、学会長は幹事会を召集しなければならない。
- 2) 幹事会の議長は学会長とする。
- 3) 幹事会は学会長、大会長、フェア実行委員長、幹事、会計監事により構成される。
- 4) 学会長は必要に応じて各種委員会の委員長を出席させることができる。
- 5) 幹事会の議決は構成員の過半数による。

第16条 (評議員会)

- 1) 学会長が毎年1回召集する。但し、学会長は必要に応じて臨時評議員会を召集する。
- 2) 評議員会は学会長の諮問に答え本会の重要案件を審議する。議長は学会長とする。
- 3) 評議員会は学会長、会計監事、Mumps 編集委員、新評議員を総会に推薦する。

第17条 (学会誌 Mumps 編集委員会)

- 1) 雑誌担当幹事は必要に応じて学会誌 Mumps 編集委員会を召集する。
- 2) 学会誌 Mumps 編集委員会の議長は雑誌担当幹事とする。
- 3) 学会誌 Mumps 編集委員は編集委員会が任命する。任期は3年とし、再任を妨げない。

第18条 (各種委員会)

- 1) 学会長は必要に応じて幹事会の議を経て各種委員会を設置、統合、分化、改廃することができる。

第19条 (日本Mテクノロジー学会大会)

- 1) 本会は年1回以上の日本Mテクノロジー学会大会を開催する。

第20条 (日本Mテクノロジーフェア)

- 1) 本会は年1回以上の日本Mテクノロジーフェアを開催する。

第六章 資産および会計

第21条 本会の資産は次の通りとする。

- 1) 本会の設立当初からの財産
- 2) 入会金および会費
- 3) 事業に伴う収入

- 4) 資産から生ずる利子など
- 5) 寄付金品
- 6) 負担金
- 7) その他

第22条 本会の資産は、学会長及び庶務財務担当幹事が管理する。

第23条 本会の重要な財産（基本財産）に関しては、これを消費し、または担保にしてはならない。但し、本会の事業遂行上止むを得ない理由があるときは、幹事会の出席者の2/3以上の議決と総会の出席者の3/4以上の議決を経てその一部に限り処分し、または担保に供することができる。

第24条 本会の事業計画およびこれに伴う収支予算は、年度毎に学会長および庶務財務担当幹事が編集し、幹事会の議決を経て総会の承認を得なければならない。

第25条 本会の事業報告書および収支決算は、年度毎に学会長および庶務財務担当幹事が作成し、会計監事が監査し、幹事会の議決を経て総会の承認を得なければならない。

第26条 本会支援のため各種団体よりの負担金、寄付、研究費などの交付があった場合、幹事会の承認により本会の資産として受け入れる。

第七章 規約の変更ならびに解散

第27条 本規約の改正は幹事会および総会において各々出席会員の2/3以上の議決を経なければならない。

第28条 会を解散するには総会において出席会員の3/4以上の同意を必要とする。

第29条 会の解散に伴う残余財産は、法律による制限のあるものの他は世界保健機構（WHO）に寄付するものとする。

第八章 付 則

第30条 本会の略称を日本Mテクノロジー学会、英文略称をMTA-JPという。

第31条 本会の入会費、年会費は別に定めるものとする。

第32条 学会長は本会の発展に功績のあった特定個人に対し名誉会長、名誉会員の称号を与えることができる。

第33条

- 1) 本規約は1977年10月29日より発効するものとする。
- 2) 本規約は1979年 9月14日より改訂し発効するものとする。
- 3) 本規約は1987年 7月29日より改訂し発効するものとする。
- 4) 本規約は1991年10月31日より改訂し発効するものとする。
- 5) 本規約は1992年 8月 1日より改訂し発効するものとする。
- 6) 本規約は1992年10月29日より改訂し発効するものとする。
- 7) 本規約は1993年 4月 1日より改訂し発効するものとする。
- 8) 本規約は1994年 8月 6日より改訂し発効するものとする。
- 9) 本規約は1995年 9月30日より改訂し発効するものとする。

「Mumps」投稿規定

(1991年 7月10日制定)

(1994年12月 1日改正)

本規定は日本MUMPS学会誌「Mumps」に、会員が自発的に寄稿する論文（以下投稿論文という）に関する必要事項を定めたものです。学会誌「Mumps」には、編集委員会が依頼する原稿（依頼原稿）も掲載しますが、それについての必要事項はそのつど定めます。

1. 論文の主題

投稿を受け付ける論文の主題は、コンピュータシステム／言語であるMUMPSに直接、間接に関係するものとします。

例えば、MUMPSの利用技術についての考案や開発、MUMPS言語についての言語仕様や提言、MUMPSシステム装備、MUMPSと他の世界とのインターフェース、MUMPSの教育など、MUMPSに関係するあるいは関係しそうなテーマについて広く受け入れます。ただし、他の雑誌に掲載された、あるいは投稿中の論文はお断わりします。

2. 投稿論文の種類

投稿論文は次の6種類に限ります。

1) 原著論文

未投稿で、論文の主要部分に独創性、独自性のある論文。既に発表した問題について別の視点からまとめた論文も未投稿原著論文であり得ます。また、応用開発、調査等であっても、その過程での創意工夫や独自性があれば原著論文の対象とします。

2) 総説

ある主題について、過去の研究業績を詳細にまとめ文献を伴って記述し、その主題に関する現状と将来展望を明らかにした論文。

3) 研究速報

新しい研究成果が原著になるほどにはまとまっていないが発表に価値があると考えられるもの。

4) 技術ノート

作成したプログラムや新しいシステムの紹介など、MUMPS技術に関する論文で、会員の相互の利益になるとと思われるもの。

5) フォーラム

意見、提案、提言、感想、著書や学術集会の紹介など、上記以外で会員の利益になるとと思われるもの。

6) Letter to the editor

原著論文に対する質問やコメント、日本MUGの活動に関係のあるコメントなど。

3. 投稿論文の長さ

原則として下記の表の通りの長さとしします。原稿用紙（横48字×縦41行=1968文字）で刷り上がりページ1枚となります。ただし、これを越える場合でも、編集委員会が必要と認めた場合には別に定める超過料金を支払って掲載することができます。

70 「Mumps」投稿規定

論文の種類	論文のページ数（刷り上がり）
原著	10ページ（以内）
総説	30ページ
研究速報	6ページ
技術ノート	6ページ
フォーラム	4ページ
Letter to the Editor	1ページ

4. 投稿者の条件

- 1) 筆頭著者は日本MTA会員であること。
- 2) 共著者も原則として会員であることとします。

5. 原稿の送付

オリジナル原稿とそのコピー2部を下記編集委員会宛てに送ってください。原稿到着日を投稿の受け付け日としその日付を誌上に明記致します。

原稿送付先・連絡先

〒259-11

神奈川県伊勢原市望星台

東海大学医学部医学情報学内

雑誌Mumps編集委員会宛

Tel:0463-93-1121 FAX:0463-96-4301 NIFTY-serveID:E-Mail PFB00710@niftyserve.or.jp（岡田）

6. 掲載の採否

投稿された原稿は、編集委員会が依頼する2名の査読者が査読します。そしてその査読者の意見を考慮して編集委員会がその原稿の採否を決定します。査読の結果によっては、原稿の内容や論文の種類を修正変更することを投稿者にもお願いすることもあります。

7. 原稿作成要領

1) 原稿の構成

投稿原稿はおよそ次の構成に従って作成してください。

- a) 論文の題名
- b) 著者名、所属、所在地
 - a) とb) は日本語と英語の両方を記入して下さい。
- c) キーワード・・・8語以内（日・英）
- d) 和文要旨・・・200字から400字
- e) 英文要旨・・・200wordsから300words
- f) 本文
- g) 謝辞・・・・・・・・必要に応じて
- h) 文献リスト

文献の引用は本文中の引用箇所に出現順に通し番号 [1] , [3 - 5] 等を記し、本文の末尾に一括

して引用番号順に並べて下さい。雑誌の文献は引用番号、著者名、論文題名、雑誌名、巻号、最初と最後の頁数、西暦年号の順です。

単行本の文献は引用番号、著者名、題名、書名、版数、引用頁、発行社、発行地、西暦年号の順です。

(例)

1. 福井太郎：糖尿病患者管理システムの開発，医療情報学，10(2):30-35(1990).

- i) 図表 … 図や表は別に一括して縮尺可能なカメラレディの図表原稿を添付し本文のどこでそれらに言及しているかを原稿のワク外に明示してください。
- j) 特殊文字…特殊文字は原則として禁止しますが使用される場合は使用位置を通常の校正の記号等を用いて朱書してください。

2) 投稿原稿（原稿用紙で提出）

原稿はワープロで、横48文字×縦41行を1頁として作成して下さい。手書きでも受け付けます。

なお、原稿には表紙をつけ、表紙にはつぎの事項を記入してください。

表紙…題名

連絡先（氏名・住所・電話・FAX）

原稿の種類

原稿の枚数（本文・図・表別に）

別冊希望部数（50部の倍数）

その他…特殊文字等を使用されている場合は明記して下さい。

3) 印刷原稿（フロッピーまたは電子メールで提出）

採用が決定した印刷原稿は、ワープロ（一太郎、MS-Word、またはMS-DOSのテキストファイル）のフロッピーまたは電子メールで提出して下さい。

（フロッピー作成時の注意事項）

- *横48文字×縦41行が1頁になって印刷されます。ただしワープロではスペースを入れないで字数に関係なく連続打ちをして下さい。改行印もスペースでなくリターンキーで入れて下さい。
- *別に横48文字で紙に印刷または手書きした原稿を添えて下さい。これを見本にして活字を組みます。横48文字目に「。」がこないように文章を工夫して下さい。
- *原稿は題名（日本語）、題名（英語）、著者名（日本語）、著者名（英語）、著者所属・住所（日本語）、著者所属・住所（英語）、和文抄録、キーワード（日本語）、英文抄録、キーワード（英語）、本文の順で同一ファイル名に保存して下さい。
- *ワープロ中には図表や図表の挿入位置の表示を入れないで下さい。図表の挿入位置は紙に印刷した原稿のワク外に朱書して下さい。イタリック等の特殊文字も紙の原稿の中に朱書して下さい。
- *図表は必ず1図を1枚の紙に印刷して下さい。そのままカメラレディで印刷します。
- *英数字は半角にして下さい。倍角その他特殊文字や罫線を使用しないで下さい。

8. 別刷

著者は別刷を最低50部買取ることとします。別刷の料金は別に定めます。別刷の部数は校正稿提出時に申し出ていただければ、50部単位で増刷いたします。

大櫛 陽一（東海大学医学部）	里村 洋一（千葉大学医学部）
大谷 元彦（藤田学園保健衛生大学医学部）	高橋 隆（京都大学医学部）
木村 一元（独協医科大学医学部）	林 恭平（京都府立医科大学医学部）
河村 徹郎（鈴鹿医療科学技術大学医用 工学部）	山本 和子（編集委員長、雑誌担当幹事） （島根医科大学医学部）

原稿募集！

雑誌「Mumps」の原稿を募集しています。ふるってご応募下さい。

応募方法: 原稿をワープロのフロッピーまたはNIFTYで紙に印字した原稿も添えてお送り下さい。

図表は別にしてカメラレディで印刷できるものをお送り下さい。

原稿送付先・連絡先

〒259-11 神奈川県伊勢原市望星台

東海大学医学部医学情報学内

日本Mテクノロジー学会 雑誌編集委員会宛

Tel: 0463-93-1121

FAX: 0463-96-4301

NIFTY-serve ID: E-Mail PFB00710@niftyserve.or.jp (岡田)

編集後記

巻頭言、論文、合わせて10編原稿を御投稿いただき、第20巻を発行することができました。御投稿下さった方々に感謝いたします。なお、平成7年10月1日から編集委員長が東海大の岡田好一先生に代わりました。長らくお世話になりました。今後共よろしく御愛読の程お願い申し上げます。

(山本)

Mumps(The Official Journal of M-Technology Association-Japan)

第20巻

1995年9月30日発行

発行者 日本Mテクノロジー学会
会長 河村徹郎
〒510-02 三重県鈴鹿市岸岡町1001-1
鈴鹿医療科学技術大学医用工学部医用情報工学科
Tel: 0593-83-9666
FAX: 0593-83-9666

編集者 日本Mテクノロジー学会 編集委員会
委員長 山本 和子
〒693 島根県出雲市塩冶町89-1
島根医科大学医学部医療情報学講座
Tel: 0853-23-2111 (内線2941)
FAX: 0853-25-2764
NIFTY-serve ID: HAH00247 (山本)

印刷 株式会社学術出版印刷
〒554 大阪市此花区春日出中2-14-9
Tel: 06-466-1588
FAX: 06-463-2522

◆ Mってなに？

言語とデータベースが一体となった
プログラミング言語です。

◆ なぜMなの？

すべてのクライアントが
データベースの熟練者ではないから。

◆ だからM！

ANSI, ISO/IEC, JIS
標準言語だから。

弊社のM言語の情報を以下のWWWサーバーで公開しています。

http://www.dec-j.co.jp/ic/solution_catalogue/dsm/dsm.html

最新のM言語に関するご質問は、最寄りの日本DEC営業所、もしくは下記までご連絡ください。

東京：03-5349-7090

大阪：06-222-9211

日本DEC

日本 デジタル イクイップメント 株式会社

本社/東京都杉並区上荻1-2-1 〒167 ☎03(5349)7111

病院経営の基盤安定には
最適なシステムが必須です。
だから、アクセル

アクセル効果①

病院全体の業務効率を高め、患者の待ち時間短縮を実現(サービス向上)

病院の各部門で発生した情報の加工・追加・変換・時間管理などを、利用する人のニーズに合わせて自動的に行ないます。これにより業務効率を高めるとともに、待ち時間の短縮をはじめとする患者サービスの向上を実現します。

アクセル効果②

患者情報、経営情報を一元管理(情報の有効利用)
各部門で入力されたすべての情報を蓄積、患者ごとに一元管理し、診断や経営戦略に役立つ各種情報に整理できます。その情報は部門ごとに設計された入力画面でマウスなどパソコン端末のマンマシン機能を使って容易に共同利用できます。

アクセル効果③

システム化の規模に合わせ段階的に導入(経済性)
システム化の規模や部門業務の増加に合わせ、段階的にアクセルパッケージ及び世界標準UNIXコンピュータなどのハードウェアから選択できる分散処理方式です。つまり(最初の一時期に必要な投資額の削減)が図られ、システムの効率的運用を実現します。

アクセル効果④

ドクターの診療判断や研究のサポート役として
(機能の充実)

一元管理された患者情報から基本属性や診療歴の即時表示、検査結果変動分析などの診療支援システムのほか統計解析などの研究支援システムが充実しています。

アクセル効果⑤

ネットワーク型システム(発展性)

さまざまな検査機器やすでに導入済みのコンピュータと接続し、異機種間ネットワークを実現。さらに、病院全体のLANを構築し、統合情報ネットワークへと発展できます。

病院総合情報システム

sumi **ACCEL** 病院経営を強力にサポートします。

ネットワーク型病院/診療所総合情報システム—アクセル

◆ 住友電気工業株式会社

情報通信システム事業部 MEシステム部システム営業課

◆ 住友電工システムズ株式会社

応用システム事業部

(お問合せは、上記両社とも)

東京都港区元赤坂1-3-12 〒107 ☎(03)3423-5880(代表)

大阪市西区土佐堀1-2-37 幸福ビル11F 〒550 ☎(06)447-7151(代表)

名古屋市東区東桜1-1-6 住友商事ビル 〒461 ☎(052)963-2755(代表)

CONTENTS

■ Editorial

Future of JIS M Language.....Yoshinari Shima 1

■ General Remarks

Japan Industrial Standard of M..... 3

Yoichi Ogushi, Masayuki Honda, Akira Fujie, Takashi Takahashi,
Yoichi Satomura, Ichirou Wakai, Kazuko Yamamoto, Tetsuro Kawamura,
Kazumoto Kimura, Yoshinori Yamashita, Yoshinari Shima, Yasumichi Kobayashi,
Shigeo Matsumoto, Hajime Nagai, Minoru Iwata, Masashi Maruo,
Hideo Kayama, Seisaku Uchiyama

The USA MUMPS Activities.....John Dumas 13

Background and Technical Features of DTM and DTM-J.....Eric K. Pape 17

The Future of M.....Robert P. Mappes 23

■ Original Articles

A New Platform for Brigham and Women's Hospital..... 31

Robert Beckley, Pasha Roberts, Michael Monroe,
Jim Marra, Frederick L. Hiltz, Ph.D., P.A. McManus

Development of A New Developing Environment Named "Easy-Tool"..... 37

Yoichi Ogushi, Tetsuo Misawa, Yoshihiro Kodato,
Chieko Konishi, Masashi Takahashi, Kenichi Hirano

Objectoriented Environment in M language.....Yukio Imaizumi 41

■ Articles

Attempts to integrate M environment and outer worlds.....Hiroshi Sato 51

■ Report: 21st M-Technology Conference

On 21st M-Technology Conference.....Yoshikazu Okada 55

Information for MUG..... 59

Instruction for Authors..... 69

Editor's Postscripts..... 73