

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。

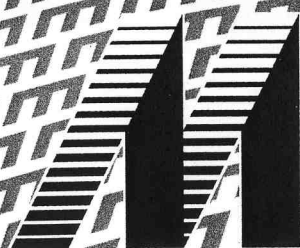
Copy right : M Technology Association - Japan

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.19, 1994

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	
21世紀のソフトウェア界を救うのは、我々かもしれない・・・・・・・・・・岡田 好一	1
■原著	
Internationalization of M: Progress and Responsibilities・・・・・・・・・・Richard F. Walters	3
中央検査部トータルシステムの再構築・・・・・・・・・・中尾 満, 森田 寛二, 藤井 厚夫, 東畠 正満, 翼 典之, 花坂 志郎, 平山 清和	15
汎用シェルと医療向けシェルの知識表現と推論機構・・・・・・・・・・今泉 幸雄, 大櫛 陽一	23
M言語よりのPC版SASシステムの利用・・・・・・・・・・田久 浩志, 今泉 幸雄	31
MとTPモニターとの統合の試み・・・・・・・・・・今井 敏雄, 佐藤 真美	37
文字検索による日本病理剖検輯報日本語データベースの構築・・・・・・・・・・馬場 謙介, 藍沢 繁雄	43
異巾文字列処理・・・・・・・・・・馬場 謙介	53
■技術ノート	
SQLによるDSMデータベースへのアクセス・・・・・・・・・・辰己 岳欣	59
最新DSMの特徴と今後のプラン・・・・・・・・・・佐藤 比呂志	63
■ 第20回 日本MUMPS学会大会 報告	
第20回 日本MUMPS学会大会を終えて・・・・・・・・・・山本 和子	67
■ 日本MUG事務局からのお知らせ	
「日本エム・テクノロジー学会」ご入会のご案内・・・・・・・・・・	75
「日本エム・テクノロジー学会」規約・・・・・・・・・・	78
■ 資 料	
投稿規定・・・・・・・・・・	83
編集後記・・・・・・・・・・	87
表紙装丁・・・・・・・・・・岡田 好一	

21世紀のソフトウェア界を救うのは、我々かもしれない

We can save computing environment in the 21st century

第21回日本エム・テクノロジー学会大会長 岡田 好一

この文章を書いているのは、1995年3月である。現在、パソコン界では、OSの主導権争いが日に日に激化している。Microsoft対IBMである。

MicrosoftのWindows 3.1のシェアは、世界的に圧倒的で、IBMのOS/2はおそらく1/10にすぎない。しかし、Windowsは完全に進化が止まってしまった。技術的には、Windows 3.1は、いかにもパソコン的な、不完全なOSである。一方のOS/2は、現時点で考えられる最高のOSの一つである。重要なことは、その最上級のOSが、ガレージブランドの激安DOS/V機でも、なんなく動いてしまうところにある。

DOS/V機は、とっくの昔に32ビット対応しており、そこに投入される技術は、かつての大型機の様相を呈してきた。これ以上の進化は、汎用機では当面考えられないところまで来た。

OSも、OS/2の登場により、従来のスーパーミニコンと、なんら変わらない状況となった(それよりも使いやすいくらいである)。しかも、OS/2では、従来のDOSで開発されたプログラムがOS/2本来のプログラムと(単に平行ではなく)協調して動かすことができるのである。

つまり、OS・ハード共に、今までの汎用機路線の最高の段階に達した、と言いきれる状況が生まれたのである。

もちろん、個々のアプリケーションに特化した、おそろしく高性能な化け物計算機があるにはある。しかし、どれも、ゲームには向かないし、安くもできないであろう。

このような状況下(世紀末とも言う)で、何か起こらない訳がない。絶対に起きる。

一つは、汎用並列計算である。これは、ある意味では、すでに完成している。

OS/2でも複数CPUが使えるようになるらしい。CPUはプロセス単位ではなく、スレッド単位で使われる。つまり、一つのプログラムが複数のCPUで処理できるのである。

言語とはいえば、第五世代計画のKL1には、複数スレッド(KL1ではプロセスという)が協調動作する仕掛けが盛り込まれている。KL1の難点は、論理プログラミングする必要がある、敬遠する人が多く、商品化されていない事である。

M言語は、もともと言語の中に、平行動作が記述できる。これはM言語が現時点で、著しい優位性を持っていることを意味すると、私は思っている。

二つ目以降は、推論システム、ニューロコンピュータ、メカトロニクスといった、一見踊り場に差しかかったと見える技術の大改革である。これは本稿とは関係ない。

おそらく、この稿をご覧になる方は、M言語に関心がおありだろう。M言語は、発達し尽くした現在のパソコンの能力をとことん使いきる事ができる、数少ないデータベースシステムの一つである。

そして、その次がまだあるのである。いや、あるようにしなければならないのは、実は私であり、この稿

2 21世紀のソフトウェア界を救うのは、我々かもしれない

を読んで下さった方々であると思う。

Internationalization of M: Progress and Responsibilities

Richard F. Walters
Department of Computer Science
University of California, Davis, CA 95616-8562 USA

Abstract

Computers developed in English-speaking countries, and almost all programming languages are based on the English language. With the advent of microcomputers, decentralization of control of systems began, and a gradual recognition of the importance of the end user emerged. As the growth potential in computing is shifting to the international market, it is increasingly evident that computers must be more effective in supporting non-English applications. A review of optimal functionality for internationalization is presented which includes features required from applications. M stands well ahead of other high-level languages in supporting internationalization, but much remains to be done to match the optimal requirements suggested herein. This report concludes with some suggestions for short-, medium- and long-term actions that could be taken by the MTA community in Japan to promote the continued leadership of M in the support of internationalization.

1. Background

Although there is some uncertainty as to who should be credited with the invention of the electronic computer, it is unquestionably true that the first commercial computers originated in English-speaking countries. As programming evolved from machine instructions through assembly languages to high-level languages, their natural language basic was exclusively English. In fact, Commander Grace Hopper, who is credited with development of COBOL, one of the first high level languages, often told the story that her invention was set back two years when she suggested to her US Navy superiors that a **compiler** might also be written to accept commands written in French instead of English. As she described it, this statement raised such serious doubts in her superiors' minds that they deferred any support for a considerable time.

We deal with an industry that relies on English. In the days of batch programs and mainframes, there was no need to consider the language preference of the user. Some accommodation of "foreign" words and character sets had to be made, but the underlying assumption was that computers "thought" in English.

This situation prevailed through the 1970s. But the seeds of change were already sown (once again, in the USA): microprocessors appeared in the mid-1970s, and microcomputers (the precursors of personal computers) first appeared on the market in 1975. (One member of MTA-Japan owned one of these early systems, which now resides in a Microcomputer Museum at the University of California, Davis.) True, the natural language used for the operating systems and programming languages for these systems was still English, but decentralization of computing power began, and the wants and needs of the end user began to play an important role in computer design and evolution.

No one can say that this revolution of the common user was easy. Data processing professionals (especially hospital administrators) resisted the emergence of locally controlled systems at every step. Large manufacturers continued to push their large mainframes. Even Digital Equipment Corporation, which had grown to become the second largest computer manufacturer by building minicomputers, was slow to accept the significance of the microcomputer (remember the DEC Rainbow?).

Slowly but surely, however, the inexorable momentum caused by the growing power of personal computers eroded the entrenched position of centralized systems. With this change came the realization that end users could have a voice in the design of new systems. One result of this

change (which has a bearing on M's evolution) was the trend towards Graphic User Interfaces, or GUIs. GUIs are significant because, unlike dialogue-driven user interactive systems, which are controlled by the program developer, GUIs present the end user with a series of options over which the programmer, once the applications are written, has no control. *The user is in charge*.

As we have said, most of these changes first appeared and flourished in the USA, where English-speaking people expressed their demands for better user interfaces and more user control. In time, however, personal computers were increasingly common in non-English-speaking countries. Here, too, users were starting to develop their own insistence on user interfaces that were consistent with their linguistic and cultural preferences.

Coupled with this change came the realization that computer markets were likely to expand more rapidly outside the US than internally. Computer hardware and software vendors began to realize that it would be essential for them to pay serious attention to non-English markets if they were to continue to grow at rates they had become accustomed to.

And so it was that, in the latter half of the 1980s, international computer organizations began to take seriously the need to address the computational environments of users in non-English cultures. The International Standards Organization, ISO, formed several task groups to study these issues. From their work emerged POSIX, ISO's Technical Advisory Group aimed at defining an international standard for operating systems that would accommodate non-English languages and cultural requirements [1]. A Task Group on the internationalization of programming languages was formed, and one of the most influential documents produced for that group was N776, prepared by the Japanese delegation to that Task Group [2]. In the world of database standardization, the ANSI-ISO committee concerned with the extension of SQL clearly articulated some fundamental requirements of databases in non-English languages [3]. Although these specifications were not incorporated into the SQL-2 standard, they remain important elements under consideration for SQL-3.

From this background, we can see that the timing is right for developers of programming languages to start thinking seriously about standards that accept different character sets and that permit effective input, storage, and manipulation of multiple natural languages. It is in this setting that we examine the role that M is taking in this international race for acceptance.

2. The History of MUMPS in Internationalization

M has been used in non-English applications for many years. In Europe, the use of 8-bit extensions to ASCII made it possible to support western European languages beginning in the late 1970s. Japanese versions of M, capable of displaying and storing kanji and han-kaku kana first appeared in the early 1980s. A public domain version was adopted by the MUMPS Systems Laboratory in Japan by the use of some non-standard extensions to MUMPS that provided minimal support for processing of 2-byte characters. In addition, M implementations using Chinese, Cyrillic, Korean, Hebrew, eastern European characters (ISO 8859 -Latin 2), and probably other versions have appeared as non-standard extensions to M.

These efforts showed that it was not hard to extend M to provide limited functionality with other character sets. However, there are several drawbacks to these versions. First, they are not a part of the standard, and in fact they add features that are not approved by the ISO M standard. Second, they are bi-lingual, with English as one language and a single additional language available in each of the versions alluded to above. Third, they do not provide complete functionality. For instance, \$ORDER was not modified to provide proper collation, even for single character codes such as the 8-bit set used in western European languages.

These shortcomings were recognized in the 1980s. The MUG-Japan meeting of 1984 was one of the first conferences featuring questions about internationalization of M. From this beginning, the MUMPS Development Committee began to work on extensions to M that would facilitate internationalization. Drawing on major contributions from MDCC-Japan and MDCC-Europe, the

full MDC gradually and painfully worked its way through a large set of problems that ultimately led to adoption of some significant extensions to M that represent a start towards internationalization. Over the ensuing four to five years (from 1987 on), a number of important elements fell into place that make internationalization of M more feasible. Among these factors are:

- The move to standardize some system-specific elements of M;
- The extension of M to provide interconnectivity between different M implementations and distributed systems;
- The Publication of a report by the Gartner Group (management and marketing consultants) pointing out the vital significance of international markets for the growth potential of M;
- Comments submitted by ISO representatives noting the deficiencies of the 1990 M standard with respect to internationalization;

These factors, coupled with the increasing interest in internationalization by hardware and software vendors and standards bodies, led to a weakening of the resistance by implementors of M to accept internationalization as a part of the language. With the adroit guidance of a few leaders in the international M community, a significant number of the principal M implementors agreed to adopt some important measures into the 1994 (?) M standard. Acceptance of these components created further delays in submission of the new updates for the standard, and it is a tribute to the full MDC membership that these delays were finally accepted as essential to the continued success of the language.

The net result of these historic events was that MDC adopted in June, 1993, a set of language extensions to M that greatly facilitate internationalization efforts. The chief components of these changes are summarized below. However, it is important at this point to make two statements:

The Internationalization of M *is not yet complete.*

and

The responsibility for further progress in internationalization of M *rests primarily on the shoulders of the M community outside the USA.*

Reasons for stressing these two points will be provided later in this paper.

3. A Brief Overview of Desirable Features for Internationalization

Before entering into a detailed analysis of the status of internationalization of M, it would be helpful to consider briefly some of the features that, in the best of worlds, would be helpful to support internationalization of high level programming languages as well as end-user applications. The review that follows is only a general overview, touching on many required features without going into complete detail on any of them.

- Character sets: Users should have access to more than one character set. The characters should be supported by appropriate methods for input, storage, and display in ways that are acceptable to users of that character set dealing with different natural languages supported by that character set.
- Cultural Conventions: Users should be able to view automatically data displayed in forms compatible with their cultural preference. Different cultural assumptions in presenting dates ("12/31/93" "31/12/93" "93.12.31" "Heisei 5.12.31" etc.), times, monetary values, formatted numerics (1,000.33 vs 1.000,33 etc.), postal address sequences and many other conventions should appear in the appropriate format to suit the end user's preferences.

- **Character Manipulation:** It should be possible to perform operations such as language-specific pattern match, collation, and other string operations on the various character sets supported by a system.

- **Programming Language Syntax:** A programming language should support non-ASCII character sets as data. Ideally, it should also support other character sets used for variable names and for labels within a program.

- **Multilingual Environments:** It should be possible to define *default* environments that assume a given cultural setting, including the character set, pattern match adjustments, cultural conventions, collation preferences, and other features alluded to earlier. It should also be possible to change these default specifications, or at least to have a separate task running concurrently on the same machine with a different set of default specifications. Further, it should be possible to have *mixed strings* of characters operating under the same environment. Additionally, it should be possible to *override* some default specifications (e.g., Pattern match) to permit multi-lingual string operations.

- **Operating System Support:** Users should be able to interact with the operating system (as opposed to the programming language) in their own native language. Messages, commands, and other elements governing interaction between user and system should be adapted to provide a user dialogue that is culturally and linguistically familiar to the end user.

These are some of the more important features required of a fully internationalized computer environment. There are at present no systems that even begin to address these items in their totality (not even the new M standard). However, the new M comes closer than any other high-level programming language in meeting more of these criteria.

4. New Features of M that Support Internationalization

The proposed 1994(?) standard contains a number of features that make it possible for M to support other character sets and non-English languages. This section presents a few of the more important additions to the new standard that facilitate multilingual functionality.

4.1 Structured System Variables

In 1977, when M was first adopted as a standard language, a number of language elements were referred to as "implementation-specific," meaning that the functionality was defined in a general way, but it was not required that all implementors adhere to precise implementation features. Many such elements were considered specific to the system on which M was installed: devices available, storage capacities, and methods for obtaining information about system configuration. For these reasons, commands such as OPEN and VIEW, and functions such as \$VIEW and \$STORAGE behaved slightly differently on various standard implementations.

This feature was acceptable for a number of years. With the increasing need to provide inter-implementation connectivity, however, it became necessary to seek new ways to standardize language elements that had previously been considered the choice of the implementor. The general form of these system-related additions to the standard were encompassed by the addition of a new language element in M: the **Structured System Variable**, or ssvn. The Following structured system variables have been defined and incorporated in the new standard:

```

^$CHARACTER
^$DEVICE
^$GLOBAL
^$JOB
^$LOCK
^$ROUTINE
^$SYSTEM

```

Of these, `^$CHARACTER` and `^$GLOBAL` are the two most important ssvn's in the context of internationalization. `^$CHARACTER` permits vendors to define more than one character set as being available on a system. Further, this ssvn allows the system to specify the exact characters to be used, the pattern codes to be used, and the collation sequence to be adopted for the character set. It is entirely feasible to consider installation of more than one specifications for the same actual character set merely by assigning each a different name. In this way, the ISO 8859-1 (Latin 1) character set used in western Europe could be separately configured to provide collation and pattern match features specific to German, French, Spanish, and Dutch (to mention just a few options), and these definitions could reside on the same M implementation.

The power of `$ORDER` for English text manipulation has been a strong feature of the M language since incorporation of string subscripts (which was initiated in the late 1970s but formally adopted in the 1984 MUMPS standard). However, not until the new M standard was the same functionality offered to users in non-English languages. It is now possible for system implementors to define new *collation algorithms* that will adhere to the collation preferences of a given language and application. In Japanese, for example, it might be possible to define collation by the pronunciation of kanji compounds by reference to a dictionary of accepted kanji compound words and names. This collation would be reflected in the subscript of both local and global variables defined under the ssvn specifying that collation sequence. Alternatively, a radical/stroke count collation table might be defined for those preferring to look up kanji by that method. (The kanji compound name/word approach would undoubtedly be more useful in most application contexts.)

As implied in the preceding paragraph, it is also possible to specify a specific character set and collation algorithm for a global variable, using the `^$GLOBAL` ssvn. Once defined, however, the character set and collation algorithm applicable to a named global cannot change. The use of ssvn's to define different collation algorithms for different globals is essential in multilingual applications, where equivalent data can be indexed by different character sets, each collated in the manner most suitable to that language.

Other features supporting these major elements are also available in the new standard. Within the scope of `^$CHARACTER`, definitions currently exist for the following supporting terms:

"INPUT"	the input transformation that converts from charset2 to the defined charset1;
"IDENT"	the algorithm used to define characters acceptable in names;
"PATCODE"	the method used to define the action of specific pattern codes in the given charset;
"COLLATE"	the collation algorithm to be used for that charset

In order to extend the functionality of two other character manipulation features of M, the definitions of `$ASCII` and `$CHAR` have been modified to accept use of different charsets.

These (plus a few other associated language elements) enable M programmers to use a variety of system-defined character set profiles. No other high-level language has begun consideration of the processes adopted in definition of M, and it seems likely that M will remain ahead of the competition for at least another decade in terms of standardized functionality available to its users as a result of this beginning.

5. Discussion: Current Status of M Internationalization

These features, approved for incorporation in the next M standard (which may be as soon as 1994), provide a unique and significant opportunity for internationalization of the language. However, it is very important to realize that *the task is not over yet!* Careful inspection of the elements incorporated in the standard reveals the important fact that, unless character sets are defined to serve the needs of a given language and/or culture, they will not be accepted as standard M components.

In fact, the portability section of the new standard specifies that **Character Set Profiles** must be registered through the MUMPS Development Committee in order to become officially acceptable components of the language.

A comparison of the desirable features listed earlier with those features incorporated in the new M standard suggests that the battle is not yet over. Although M is ahead of its competition, it is by no means in a position where one can be complacent about the work thus far accomplished.

An important point that must be understood in the context of multilingual and multicultural internationalization is that the new M Standard, in its current form, is a facilitating agent, not a final solution to the many problems of internationalization. The concept of an MDC-approved **Registry** of character sets, pattern match codes and their meaning, and collation sequences means that these specifications **remain to be developed**. Equally important, no vendor is going to implement any of these features until they have been approved by MDC for Registry into the accepted regional components that, taken together, will permit internationalization.

A corollary to the concept of a Registry is that, although a given local set of specifications for charset, patcode, and collation sequence may be accepted by MDC, it is not required that every vendor incorporate every registered set of codes for every implementation of M. Since there are likely to be some performance penalties in providing multiple regional specifications, implementors are much more likely to offer customized versions for individual clients, including only those regional charsets, that the client stipulates as being required. An American user marketing in Europe will not need (at least immediately), a hanzi or kanji charset, and a Japanese user may feel that inclusion of ISO 8859-2 (Latin 2: East European) is unnecessary.

Further analysis suggests that the international M community has an opportunity to provide, once again, the leadership that will be required to make further progress in extending M to provide full internationalization functionality. The responsibility for capitalizing on this opportunity (someone has stated that most "problems" can be restated as "opportunities") rests primarily on the shoulders of the international M community for the following reasons:

- American implementors and software developers are not yet fully convinced that internationalization is essential.
- Implementors are especially conservative when it comes to adding "required" functionality that might affect performance of applications that do not require that functionality.
- Many aspects of definition of specific elements related to non-English languages and to cultural preferences in different areas are best understood and defined by local users and developers familiar with those conventions and requirements.
- The judgement of native English speakers as to "what is best for internationalization" is not likely to be trusted by other members of MDC, with good reason. Such individuals may press too hard for some items considered of secondary importance and at the same time omit features that are recognized as vital by the international community. Vendors in particular will accept much more readily pressure from representatives of end users who demand new language elements in M rather than trust the opinions of English-speaking "experts" whose experience in internationalization is limited.

Given this background, it is reasonable to expect that, unless the international M community takes the lead in exerting pressure on MDC to continue the important process of internationalization of the M standard, it *will not happen*. In the next sections, I propose some thoughts as to some steps that might be most constructive in pressing for progress in this arena.

6. Continued Internationalization of M: Opportunities for the M Community in Asia and Europe

The process of continued internationalization of M can be considered from several viewpoints. In this section, I would like to focus on three different perspectives that have direct bearing on action steps that can be taken by the M communities in Japan, Europe, and Brazil in promoting the more rapid progress of M towards becoming fully internationalized.

6.1 Short Range Tasks to Support Internationalization

The new M standard paves the way for regional groups to define standard approaches to charsets, patcodes, and collation sequences. The standard does not, however, address any of the particulars of these issues, and in fact all such items will be available as MDC-approved registers of specifications. *The only body that should tell the full MDC what charsets, patcodes and collation sequences are required by the Japanese community is MDCC-J*. That group needs advice from you, the users of M in Japan. A number of issues need to be addressed, ones with which most of you are already much more familiar than I.

First, the question of charset(s) needs to be addressed. "Gaijin" like myself are generally unaware of the considerable diversity in codes used by different segments of the Japanese computer industry. It will be necessary to decide which charset(s) should be approved as standard. This decision should probably be cast in a forward-looking mold that may even influence the convergence of different character sets as used in Japan today. Serious questions remain about the incorporation of the third level of the JIS X 0212 standard (the supplementary kanji added to the Level 1 and Level 2 sets currently in use).

The matter of patcodes also may require reconsideration (beyond the patcodes previously adopted as an informal standard in Japanese versions of MUMPS). Is there a reason to distinguish between the different levels of kanji JIS codes? Can Japan take the lead, with the multiple alphabetic fonts available in JIS X0208 to establish some patcodes that distinguish between Latin, Greek, and Cyrillic (as well as diacritically extended Latin)? There are opportunities for the MDCC-J to provide important leadership as a function of these known characteristics of the JIS codes which should be seriously considered at this time.

Collation algorithms also require definition, and here again, your unique problems of dealing with kanji that have multiple pronunciations places you in a position of potential leadership for other languages and cultures. If you assume that collation using \$ORDER by pronunciation is desirable, then it will be necessary to define the authoritative reference, published or generated by your own efforts, that would serve as the lookup table for these kanji compounds (names included). Once again, the considerable complexities of this subject are beyond the knowledge of all but a very few foreigners, but the work done could easily serve as a model for dealing with Chinese dialects and with the hanja characters used in Korea.

As a matter of political expediency, it might be advisable in the short term future to raise the question of character size and the need for updated replacements for \$X and \$Y to accommodate, among other things, the configuration of kanji, hankaku, and related textual presentations in display and hard copy. This is politically expedient because the need for this updating is acute even in English, where M lags seriously behind other languages in its ability to deal effectively with proportional spacing, mixed fonts, and other graphic aspects of output. Support of this factor from the viewpoint of English-speaking users would add to the positive image of some other elements less obviously needed by those individuals.

These are the most important elements that should be agreed on, put into formal definitions, and submitted as soon as possible to the MDC for registry.

6.2 Intermediate Term Opportunities for Leadership in Internationalization

Although completion of the items required in the first registry for internationalization of M are important, there are some equally important issues that should be addressed soon. M's potential to provide models for other programming languages in these fields is considerable, but without leadership from MDCC-J, this opportunity might be lost. A few of the most important issues are:

- International Character Sets.

The current controversy of Unicode vs ISO 10646 has only partially been resolved. If MDCC-J, in conjunction with MDCC-E, were to take on some of the problems of M treatment of these character sets in dealing with *several* natural languages, the entire computing community would sit up and take notice. Resolution of issues dealing with treatment of stylistic and linguistic differences in the use of commonly derived kanji in Chinese, Japanese and Korean would provide a model that is desperately needed. Establishing a multilingual collation algorithm that permits use of multi-octet, multilingual character codes would be a contribution that could *only* be achieved by a close cooperation between specialists in Europe and Asia, with Japan providing the leadership for the Asian sector. The resolution of patcodes applicable to multilingual character sets referenced in the previous section could serve as the framework for dealing with the more complex multilingual character sets of ISO 10646 or Unicode.

- Dealing with Mixed Character Sets.

The M community currently has no portability standard for the exchange of character sets of different regional or national standards. Although this feature need not become a component in the M standard, some form of registry for interchange of character sets between M implementations should be provided. Whether it draws on accepted escape sequences or other conventions is a matter for discussion among representatives of MDCC-J and MDCC-E, but this matter should be incorporated into a registry. (Once again, without leadership from the international community, this point will continue to be ignored by the American representatives on MDC.)

- Developing a Prioritized set of Features that should be Added to M to Complete the Job of Internationalization.

As noted earlier, many elements required for complete internationalization remain to be incorporated in the next standard of M, which one hopes will be addressed within the next four years. Although individuals like myself might want to see all these features accepted immediately, it is unrealistic to expect the implementors of M to accept immediately such potentially radical changes as user-defined charsets, patcodes and collation algorithms. It is the responsibility of groups like MDCC-J, therefore, to examine those features that remain to be incorporated for complete internationalization, and to bring such a list, *in prioritized order*, to the MDC for consideration.

I would at this point like to inject one small personal bias. It seems to me especially important to modify the M syntax to permit **override** of default system environment features such as patcode, so that mixed-language operations could be performed in M. An important addition to the new standard dealing with Networking defines a **Vertical Bar Syntax** to standardize overrides of system/path locations of globals and routines. This important concept of overriding default conditions is very close (but not quite identical) to the override syntax needed for multilingual processing. I believe that this issue can and should be addressed very soon, since it is already partially resolved, so that M users will truly be able to deal with multilingual applications.

Beyond this small personal preference, I urge the Japan MTA community to consider seriously the most important "missing elements" that should be included in the near future versions of the M standard, and then, working with MDCC-E and MDC, ensure that these features get the constant attention and coordination with M implementors required to ensure their early acceptance into subsequent M standard releases.

6.3 Long-Term Opportunities for Leadership in Internationalization

It is much easier to address the immediate tasks at hand than to back off and view the opportunities that might open up over the next few years. In this section, I would like to propose some of the more indirect ways in which real progress towards internationalization might be accomplished, not only for M but in the broader sense of the needs of our various countries and cultures to communicate more effectively.

There is an American saying, "Everyone talks about the weather, but no one does anything about it." In the M community, we have talked a great deal about internationalization (I am one of the worst offenders), but until we start *doing something* about it, there will be no direct pressures on vendors to make real progress with those features we will require for really effective multilingual uses of M.

I believe that there is something that can indeed be done in this regard and that the MTA-Japan community is in a good position to provide leadership in this regard. I offer a few sketchy thoughts of projects that might serve to stimulate further goal-oriented evolution of M towards serving the international community.

- Development of Internationalization Utilities

The process of working with more than one charset, more than one language, and more than one version of an application is ideally suited to the windows environment. To take one example, the work of Satomura and do Amaral on multilingual medical dictionaries would certainly be effectively displayed if multiple windows with different default environments were displayed on the same screen [9].

The newly approved Windows Applications Programmer's Interface (WAPI) is a perfect vehicle for developing a set of utilities that could be generalized to serve not only the dictionary work of Satomura and others, but also to serve effectively in multiple situations where two or more languages are to be viewed simultaneously. I believe that the richness of the display and storage situations encountered in Japanese would provide unique insight into the design of optimal multilingual utilities. Opportunities abound for developing effective input, alternative display, mixed character processing as well as display in user-friendly screens.

Once such utilities have been developed they should be applied to some real world situations. One candidate for such an application is described in the next section.

- Multilingual Translation of Applications Software

FileMan has been translated into Japanese [4,5], Chinese [6], Finnish [7], German [8], and most likely into several other languages as well. Unfortunately, each of these projects has been done in a bilingual mode. There are many inefficiencies of the translation process using the current versions of M, and there has been no attempt to capitalize on the potential for shared development and true multilingual approaches to the problem that would not only facilitate translation into other languages, but also upgrading existing translations to new versions.

It might be possible to form a working group composed of representatives of several cultures (e.g., Japan, China, and Germany) to consider development of a new translation methodology for FileMan that would facilitate both updates and inclusion of new languages. That study might first seek to define *interlingua* forms of internal representation of FileMan text, and then develop natural-language specific components that represent the internal form of each phrase. The study could then analyze the FileMan package and develop a set of guidelines and recommendations for future modifications of FileMan itself that would further simplify the translation process. Finally, the group might prepare a set of guidelines for subsequent initial development of software designed specifically to be translated to other languages.

The choice of FileMan as an example is made only because there is a significant body of work already accomplished that could facilitate development of a truly multilingual approach. As an alternative, MTA-Japan end users might identify another package that appears to have real merit for translation, and approach the developers of that package to consider undertaking a similar approach.

- Offering Experience and Advice to M Implementors

The reluctance of M implementors to move too rapidly towards acceptance of internationalization features as standard components in a future version of M is based on their uncertainty as to the meaning, implications, and demand of these components. With the experience gained from projects like those just discussed, MTA-J software developers could show M implementors the way to develop and use much-needed features supporting internationalization. This assistance might be helpful in accelerating the adoption of prioritized language enhancements aimed at supporting internationalization applications.

- Review/Update of ISO Internationalization Effort

ISO, through multiple subcommittees and work groups, has invested a good deal of time and energy on the problems of internationalization. However, these processes have not been effectively coordinated. Indeed, there is minimal overlap between committees, and sometimes the activities of one committee overlap significantly with those of another. For example, language issues such as collation are raised in the context of databases, operating systems, programming languages, and library systems. At this point, however, M is the only language that is actually doing something concrete in the area of internationalization. Our work should not only serve as a model for other groups, it should also help other groups understand better the options for coordinating the disparate activities of the many groups concerned with internationalization, so that more rapid and consistent progress might be made in defining and then implementing new internationalized standards.

7. Conclusions

When M first appeared, it was more than a decade ahead of the programming languages of its time. It held some aspects of that leadership through the 1970s and early 1980s, but the advent of user-oriented systems in the 1980s placed it at a disadvantage in recent years. M has taken an important step towards regaining its leadership position with the new M standard. With your help, M can continue to surge ahead of its competition. The future is in your hands. Gambatte kudasai!

8. Acknowledgements

I am deeply indebted to Ichiro Wakai, MD, for his careful reading and constructive suggestions of preliminary drafts of this paper. The opinions expressed, however, are solely my own responsibility.

9. References

1. IST/JTC1/SC22/WG15: POSIX Work Group (many documents on internationalization of operating systems).
2. Japan National Standards Institute (1989), "Recommendations for Character Handling in Programming Languages," *ISO/JTC1/SC22/N776*.
3. Birdsall, D. (1989), "Character Sets and Collating Sequences," *ANSI X3H2-89-254*.
4. MUMPS Systems Laboratory (1982), *File Manager User's Manual V. 15 for Katakana*, Nagoya, Japan

5. MUMPS Systems Laboratory (1987), *VA FileMan User's Manual V. 17.3 for Japanese*, Nagoya, Japan.
6. Huang, F, and Zheng, T (1994), "Features of MUMPS Required in China: Insights from Translation of FileMan," *M Computing (in preparation)*.
7. Koskimies, Y, et al. (1983), *File Manager Tekninen Kuvaus*, Helsinki, Finland, 1983.
8. Moore, G.W., and Giere, W. (1993), "English to German Translation using VA File Manager," *M Computing* Vol. 1 No. 4 (*in press*).
9. Satomura, Y. and do Amaral, B. (1993), "Multilingual Medical Dictionaries for Natural Language Processing," *M Computing*, Vol 1, No.4 (*in press*).

中央検査部トータルシステムの再構築

Reconstruct of Total Laboratory System

中尾満¹、森田寛二¹、藤井厚男¹、東畠正満¹、花坂志郎²、平山清和²

Mitsuru Nakao, Kanji Morita, Atsuo Fujii,

Masamitsu Higashihata, Shiro Hanasaka, Kiyokazu Hirayama

1993年5月に開院した新病院は、発生源入力によるオーダリングシステムを採用し、病院トータルシステムの導入を図った。中央臨床検査部では、MUMPS言語を採用した従来からの部門システムを今回、新たに再構築を図った。今回のシステムは病院システムおよび日立検体搬送システムとの情報ネットワークを完成し、生理検査や総合診療科を含むオンラインシステムを構築した。

1 病院トータルシステムの概要

1. 1 ハードウェア構成と通信ネットワーク

病院ホストコンピュータはM770/8を2台配置し128MBの共有メモリーとデータベースの2重化を図り、システムのノードダウン方式を採用した。

薬剤部や放射線部などには部門システムとしてKシリーズのCPUを配置した。

中央検査部にはSUN4/670を2台配置し12台のターミナルサーバーを介して120回線を用意した。オーダー端末及び部門システムの入出力端末には主として約800台のFMRHL2を用いた。通信回線には410MbpsのMHLINKと10MbpsのDSLINKを介して情報のネットワークを図った(図1)。

1. 2 オーダーシステム

外来診療及び病棟業務は発生源入力を基本としたオーダリングとし、検査オーダーはM770を介しリアルタイムでSUN4に依頼情報が転送される。オーダーは生理検査・超音波検査・緊急検査を含む全ての検査室を対象とした。

1 大阪市大病院中央検査部

〒589 大阪市阿倍野区旭町1-5-7

TEL 06-645-2121

2 住友電工システムズ株式会社

〒550 大阪市西区土佐堀1-2-37幸福ビル

TEL 06-447-7150 FAX 06-447-7150

2 検査部門システムの概要

2.1 システムの概要

中央検査部門システムの関係概念図(図2)を示した。

新システムのソフトウェアはUNIX配下でMUMPSを選択した。臨床検査部では1976年よりPDP 11シリーズを用い、MUMPS言語による検査業務を行ってきた実績が有り、ソフトウェア資産や業務機能の継続を図りながら、新システムはオーダーリングシステムへの対応や検査業務の充実を目的とした。

今回採用したSUN4/670は2CPUを搭載し、1CPUのMIPS値が27MIPSの能力を持ち、従来から指摘されていたMUMPSのレスポンスの低さを大きくカバーすることが期待される。更に、2台のSUN4をルーチン系と緊急検査及びオンライン系の2系統に分散しCPU負荷の軽減を図るとともにダウン時の検査業務処理への対応を考慮した。

2.2 検査受付処理と検体採取の準備

検査部中央採血室では、外来患者の採血及び入院患者の凝固系採血を実施している。患者の受付はIDカードにより行い、FMR画面上に検査可能オーダーを黄色反転表示させ患者の受付処理を行う。なお、受付処理と同時に病院HOSTコンピュータに対して受付情報を転送し、医事課システムに送られる。受付処理された情報から8種類の採血管は自動バーコードラベル添付装置により準備され、その他の採取ラベルは専用バーコードプリンターにより打ち出され、採取ワークシートと照合し採取容器を患者毎に用意する(図3)。入院患者の採取容器は前日15時までに入力された検査オーダー分に対し、検査部において上記装置セットにより用意し、各病棟には院内用自動搬送台車を用いて搬送する。検査依頼の受付の流れを次に示した。

2.3 バーコードラベルの仕様

検体採取容器に張り付けられるバーコードラベルは12桁の数字をNW7フォーマットのバーコードと患者氏名や採取容器名称など13種類の情報を印字した。

このラベルに印字されたバーコードにより検査部システムや検体自動搬送システムにおける情報の確認に利用される。

2.4 検体受付到着処理

検体部門システムへの検体到着処理はバーコードリーダーにより行う。なお、検体検査の多くは検体自動搬送システムとの情報ネットワークにおいて自動的に検体到着処理が行われる。

3 検体自動搬送システムとの連携

3.1 システムの概要

今回開発した検体自動搬送システムは、検査部門システムとオンライン化し、検体検査業務に必要な多くの機能を集約することを図った。検体自動搬送システムの主な機能は次に示す。

- イ. 検体到着受付処理
- ロ. 遠心分離
- ハ. 自動分析装置用検体分注
- ニ. 血清分離分注

ホ. 検体保存

ヘ. 精度管理

検体自動搬送システムはHILAS(L-780/10)により一元管理されている。

検体自動搬送システムの概要図4は次に示した。

図中の自動分析装置22台はHILASにオンラインされ、検査結果はリアルタイムでSUN4に転送される。

4 生理検査室におけるオンライン化

生理検査室では、今回新たに心電図解析システムであるMUSEシステムや超音波ファイリングシステムとのオンライン化を図った。依頼情報や患者情報は検査部門システムから情報を転送し、検査結果としての数値情報や解析コメントは検査部門システムを経由して病院ホストコンピュータに転送される。

5 検査結果の報告

検査結果は、検査部門システムによる結果の最終チェックを行った後、病院HOSTコンピュータに転送される。一方、検査部門システムにより各種報告書の発行が行われ、同時にHOSTコンピュータには最終報告として再度結果の転送を行う。更に、病歴システムに対して磁気テープを介して報告書の患者情報を渡すことにより、シングルピッカーによるカルテの取り出し指示を行う。

6 まとめ

新病院立替えにあたり病院トータルシステム化にともない、検査部門システムの再構築を行う機会を得た。

今回の再構築により検査情報の病院ネットワークが出来上がった。中央検査部では新たな業務として総合診療科での緊急検査の支援や検査部コンピュータシステムの24時間稼働による緊急検査の実施のほか、病棟に対しては検体検査採取容器の検査部からの供給など、業務の拡大を図った。新システムの運用に入ってからまだ数カ月を経過した段階ではあるが、検査部トータルシステムの初期の目的はおおむね達成されたと考えている。

大阪市立大学医学部付属病院 検査部門システム構成図

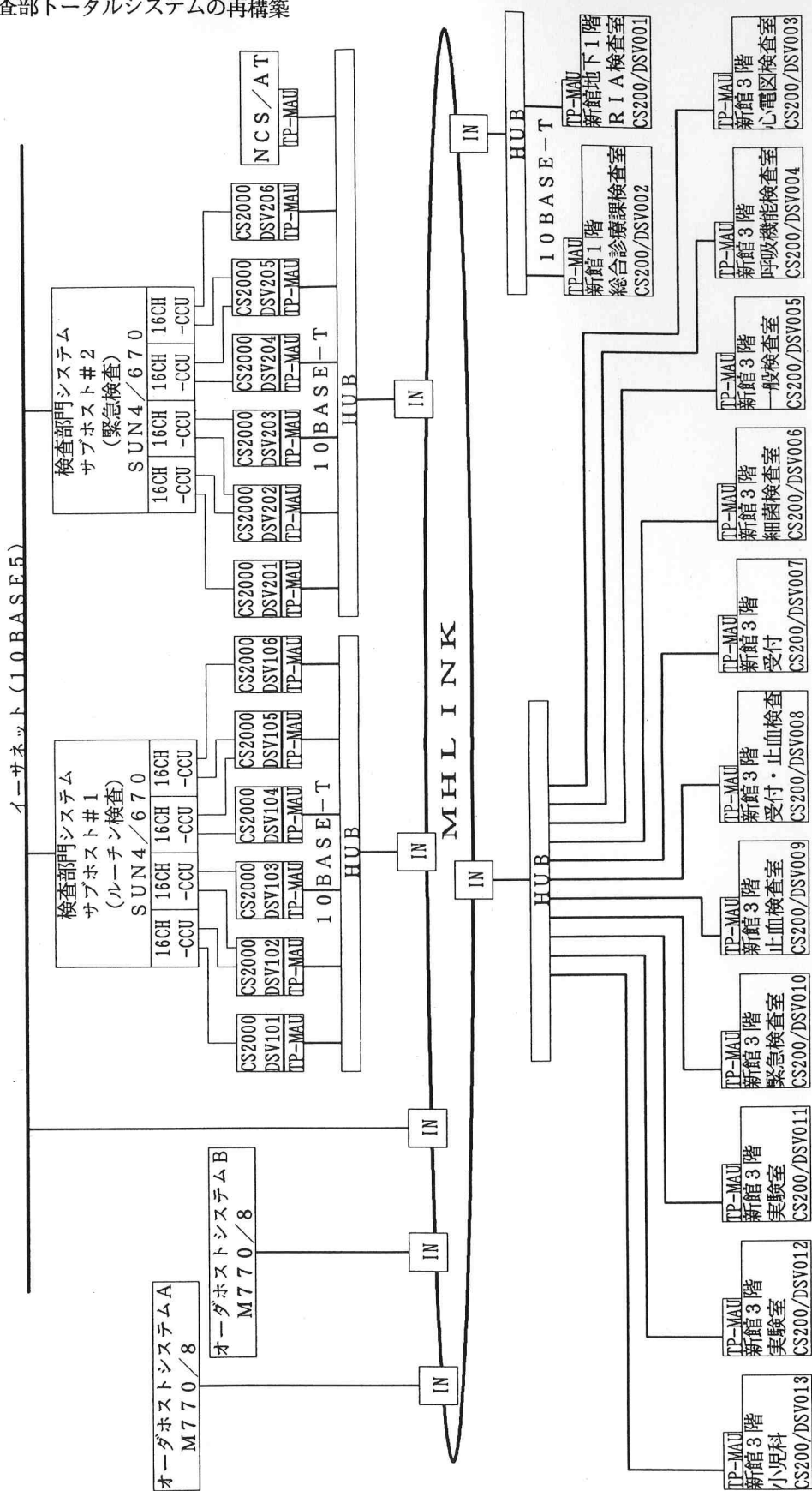


図1 大阪市立大学医学部付属病院 検査部門システム構成図

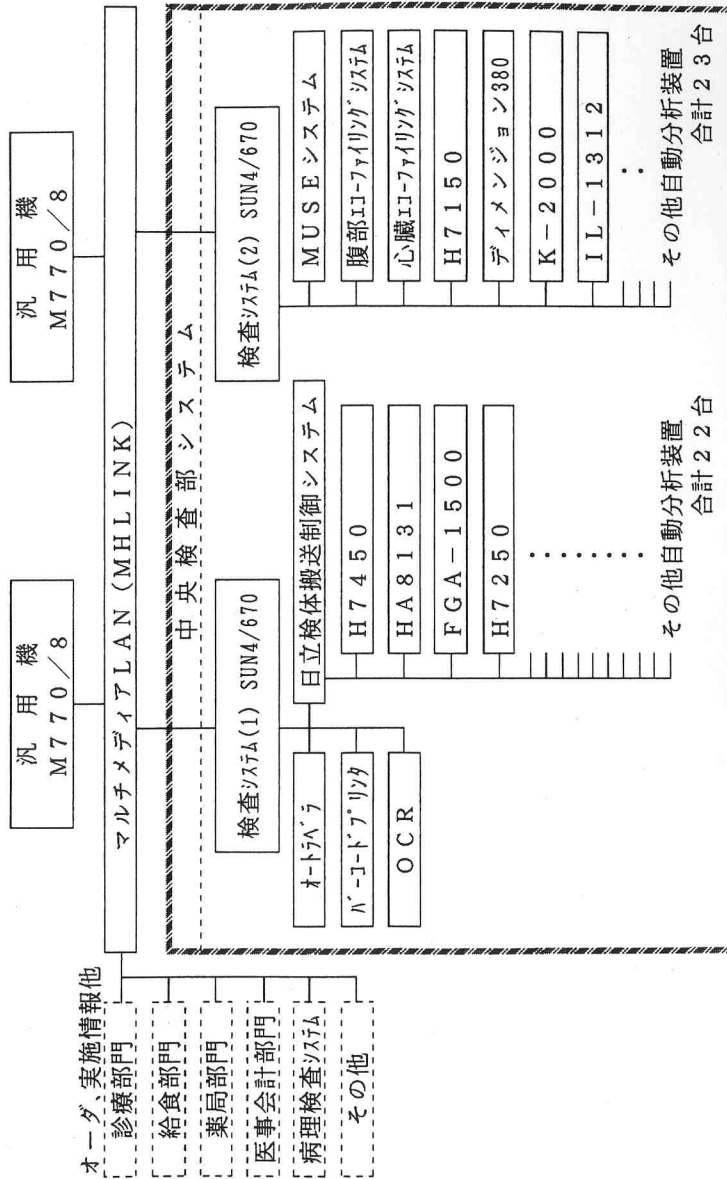


図2 大阪市大中央検査部システム関係概念図

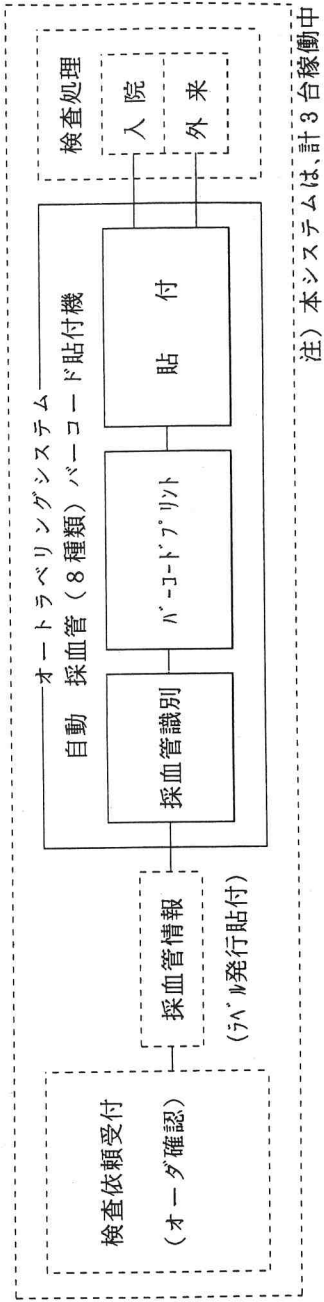


図3 中央検査部システムに於ける受付オートラベリングシステムの概要

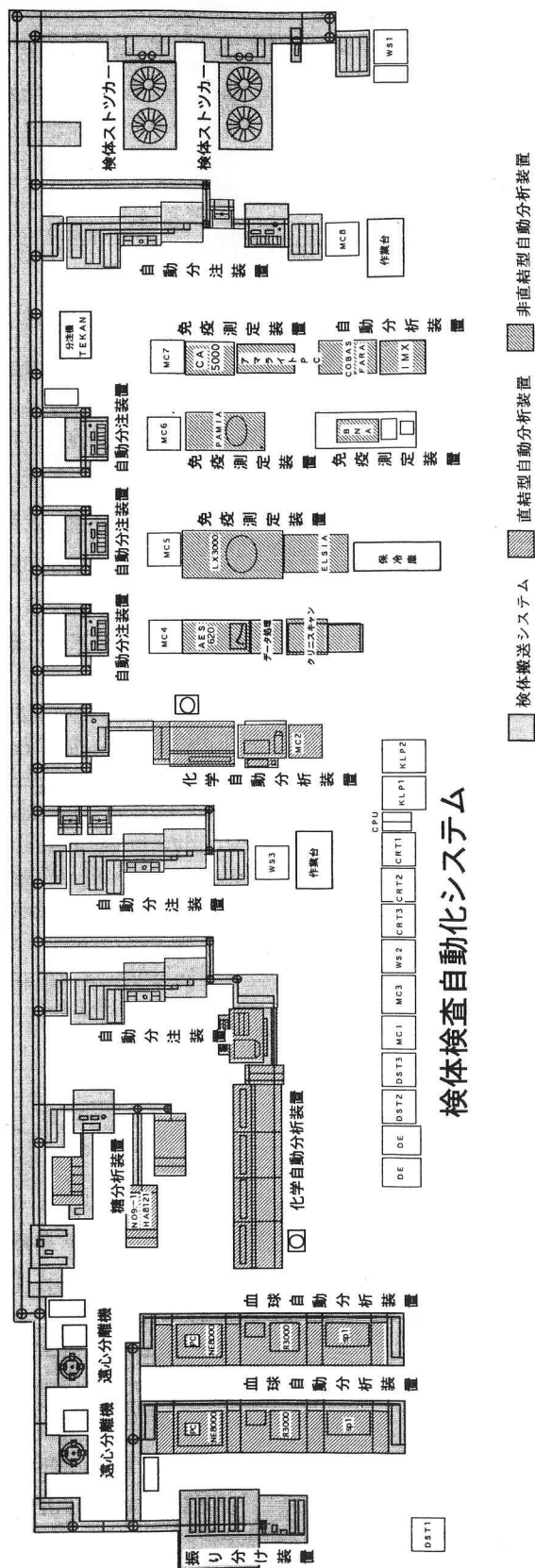


図4 中央検査部システムにおける検体搬送制御システムの概要

汎用シェルと医療向けシェルの知識表現と推論機構

Knowledge representation and inference engine in general expert shell and medical expert shell

今泉幸雄¹、大櫛陽一²

Yukio Imaizumi, Yoichi Ogushi

知識表現として、汎用シェル向けにプロダクション・ルール（以下ルール）と医療向けにフレーム知識を用いた。これらの知識表現に対して、推論機構ではルールベースのエンジンを作成した。まず、全ての事実データに対して、前向き推論が働いた後に、後向き推論が動作して利用者への問い合わせをする。汎用シェルでは中間仮説を展開しながら推論する。医療向けのフレーム知識は推論される前までに、1スロット知識が1ルールに展開される。利用者からみると異なる知識表現に対して、同じ推論機構を採用した実例を報告する。

（キーワード：知識表現、プロダクション・ルール、フレーム、推論機構、前向き推論、後向き推論、汎用シェル、医療向けシェル）

In this expert shell system, there are production rule (referred as rule) for general expert shell and frame representation for medical shell. In this inference engine, we have made a rule base engine for each of rule and frame. Once the inference engine invokes the forward engine to all data, it invokes the backward engine to request to users the raw data. The general expert shell uses medium hypothesis during inference. Before the medical expert shell invokes the inference engine, it translates 1-slot knowledge to 1-rule. We hereby report, there is a different knowledge representation at viewpoint of users, but that these shell use the same inference engine.

(Keywords: knowledge representation, production rule, frame, inference engine, forward engine, backward engine, general expert shell, medical expert shell)

1 はじめに

エキスパート・システムを作成するには、シェル（構築ツール）を用いるのが通常である。シェルは知識表現・推論機構・ユーザーインターフェース・知識ベース管理等から構成される。われわれは汎用シェル向けの知識表現としてルールと、ドメインシェル（特定専門向け）の対象として医療向けシェルのフレーム知

1 サンド薬品株式会社筑波総合研究所

〒300-33 茨城県つくば市大久保8

TEL 0298-65-2205

2 東海大学医学部病院管理学教室

〒259-11 神奈川県伊勢原市望星台

TEL 0463-93-1121

1 SANDOZ PHARMACEUTICALS LTD.

Ohkubo 8, Tsukuba-shi Ibaraki 300-33, Japan

TEL 0298-65-2205

2 Division of Hospital Administration, School of Medicine,

Bouseidai, Isehara-shi, Kanagawa-prefecture 259-11, Japan

TEL 0463-93-1121

識表現（以下フレーム）を作成した。推論機構としては異なる知識表現に対して、前向き推論と後向き推論を組合せたエンジンを開発した。経緯としては先に汎用シェルを開発したので、汎用シェルのルール知識表現、推論機構、医療向けシェルのフレーム知識表現の順で以下に報告する。

2 汎用シェルのルール知識表現

利用者がルールを編集（入力・変更・削除を総称して編集と呼ぶ）する時は、知識ベース編集を実行する。利用者がルールを管理しやすくする目的で、グループ知識（内部ではメタ（"META"））を用いる。1グループ知識は複数のルール集合より成り立っている。グループ知識単位に"診断型"と"計画型"の区別をする。

2.1 知識ベース編集

編集の単位は1グループ単位で、かつグループに属する1ルールが対象となる。1ルールは"もしも***ならば、***の疑いがある"というIF（条件部）, THEN（結論部）の表現を採用し、条件部は複数の表現が可能である。結論部で結論名に対する確信度を入力する。1ルールの表現としては3種類存在する。

- (1) 条件部が事実データで結論部が中間仮設名
- (2) 条件部が事実データで結論部が結論名
- (3) 条件部が中間仮設名で結論部が結論名

下記に実際の実行例を示す。

知識ベース編集(ルール知識の編集)

ルール知識名は"グループ知識名(英数字5文字以内)"+"数字(3文字以内)"からなる。大きくは条件部(if:もしも***ならば)と結論部(then:***の疑いがある;確信度は0.7)から構成される。条件部の日本語は必ず主語+格助詞(はorが)+動詞(例:鼻水がある)とし、結論部は名詞句(風邪の疑い)に確信度は0.0<1.0あるいは"YES"となる。

- | | |
|-----------------------|--------------|
| 1. ルール知識名(グループ知識名+数字) | : YIKB010 |
| 2. 結論部の名詞句(英数字20文字以内) | : 風邪の疑い |
| 3. 結論部の確信度 | : 0.7 |
| 4. 条件部1(英数字30文字以内) | : 鼻水がある |
| 5. 条件部2(英数字30文字以内) | : 熱が38度以上である |

2.2 知識ベースファイル

知識ベースファイル(^ZFKB)はグループ知識("META")とルールに分けている。ルール名は"グループ知識名+数字"から構成されているので相互に参照できるようになっている。推論実行の初期処理として^ZFKBは、1ルールの条件部の一部をキーとした知識テーブル(ZTKB)として展開される。

^ZFKB

| ★設定情報

| ->>^ZFKB(KBFLNM) :

| ①KEY : 知識ベースファイル名 VX(8) (標準名として"DEFAULT")

| ★ : 知識ベースファイル情報

| ->>^ZFKB(KBFLNM, "META") :

| ②KEY : Meta知識の定義 FA(4)

| |

| | ->>^ZFKB(KBFLNM, "META", METANM)

| | ③KEY : 個々のMeta知識の名前 VX(6)

| |

| | ->>^ZFKB(KBFLNM, "META", METANM, "NMDATE")

| | ④KEY : 作成定義1 FA(6)

| | ★ : 作成定義1情報

| | ->>^ZFKB(KBFLNM, "META", METANM, "COMENT")

| | ④KEY : 作成定義2 FA(6)

| | ★ : 作成定義2情報

| | ->>^ZFKB(KBFLNM, "META", METANM, "KBBODY")

| | ④KEY : 作成定義3 FA(6)

| | ★ : 作成定義3情報

| |

| ->>^ZFKB(KBFLNM, "RULE") :

| ②KEY : ルール知識の定義 FA(4)

| |

| ->>^ZFKB(KBFLNM, "RULE", RULENM)

 ③KEY : 個々のルール知識の名前 VX(8) (METANM+

 ★ : ルール知識の内容 +"1" ---> "n")

3 推論機構

メニューから推論実行が選択されると、推論の初期化として知識テーブル(ZTKB)を作成して、利用者から事実データの入力をうながし、事実データの全てに対して前向き推論が実行される。前向き推論結果が表示された後に、後向き推論が実行されて、利用者に対してまだ入力されていない事実データの値の入力を要求して推論する。後向き推論が終了すると両推論の結果が表示される。

下記に推論の実行例を示す。

推論実行

今から推論をする準備をします。

知識ベースのファイル名を入力してください：KB001

ルール数は20で、作成者は鈴木一郎で、

作成日は1991年5月4日で、最新更新日は1991年5月10日です。

事実データは主語、助詞("は","が"), 述語の日本語とする、

例として[熱は高い, 鼻水が多い]の形式となる。

ADD/DATA(事実データの追加:終了は改行キーのみ押す): 下痢は米水

ADD/DATA(事実データの追加:終了は改行キーのみ押す): 熱は39度以上

ADD/DATA(事実データの追加:終了は改行キーのみ押す): 下痢の頻度は多い

ADD/DATA(事実データの追加:終了は改行キーのみ押す):

つぎの処理を選択してください

ADD/DATA, SHOW/DATA, RUN(推論) : SHOW/DATA

事実データの内容です。

1. 下痢は米水

2. 熱は39度以上

3. 下痢の頻度は多い

以上3件である。

ADD/DATA, SHOW/DATA, RUN(推論) : RUN

これから(前向き)推論を開始します。

中間仮設として ==> 風邪の疑いがある。

用意された事実データからの推論が終り、

これから(後向き)推論を開始します。

涙目があるの症状はありますか (Y/N/<U>) : W

Why (理由)

花粉症の確認をします。

涙目があるの症状はありますか (Y/N/<U>) : T

Trace on

(trace: 質問生成の処理を実行しています)

涙目があるの症状はありますか (Y/N/<U>) : E

Explain (説明機能)

ルール番号(RB101)が起動されてます。

涙目があるの症状はありますか (Y/N/<U>) : N

No

(trace: 質問生成の処理を実行しています)

咳はでるの症状はありますか (Y/N/<U>) : W

Why (理由)

風邪の確認をします。

咳はでるの症状はありますか (Y/N/<U>) : T

Trace off

咳はでるの症状はありますか (Y/N/<U>) : Y

Yes

脱水があるの症状はありますか (Y/N/<U>) :

Unknown

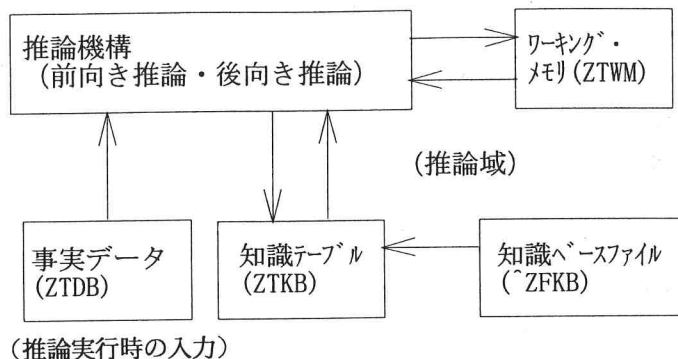
推論結果は以下のとおりです。

1. 風邪である。 1.00
2. コレラの疑いがある。 0.75

推論の実行は終了しました。

(If any key return , we will come to previous menu.)

推論のメカニズムの概略としては次の図になる。



推論域となるワーキング・メモリー(ZTWM)を中心に説明する。ワーキング・メモリーに格納された候補数は推論終了したときに表示される候補数となる。推論中に新しい候補名が入ると"+1"(1加算)される、初期値は"0"である。知識ルールが推論実施されると、ワーキング・メモリー(ZTWM)に出力される、例えば下記の知識テーブル(ZTKB)が事実データテーブル(ZTDB)あるいはZTWMと一致すると

```
ZTKB("RB204")="NOTRUN" ^ "YES"; "下痢" $ が "米水" $ "YES" ^
コレラ" ^ 0.7 "RRA30";
```

ZTWM("コレラ")=2; RF305, YES ^ RB204, 0.7 この場合は"コレラ"はすでに候補として知識ルールの"RF305"も一致していた。

全ての推論の実施終了時にワーキング・メモリー(ZTWM)の内容が表示(一部確信度などは計算する)されて推論の終了となる。

ZTWM

| ★設定情報

|

| ->> ZTWM(INFCNDNM):

①KEY: 知識ルールを推論した結果の候補名(例として"風邪の疑い", "コレラ") VX(10)

★: 推論した候補名に対する参照頻度や知識ルール名などの情報
(★: データノードを示す。)

4 医療向けシェルのフレーム知識表現

医学教科書は疾患名(病名)に対して考えられる複数の症状(データ)を並列的に記述されているものが多い。知識の編集単位も疾患名単位のが扱いやすいのでフレームで表現する。知識の入力方法としては疾患名(結果 OR 結論)を先に入力してから、その症状名(原因 OR 条件)を複数個入力(n個)する。

4.1 フレーム知識表現の解釈

(例) コレラや風邪は疾患名と考える。

コレラ ----- 発生頻度 1.0% (事前確率)

+

+-----> 熱が39度以上 10% (事後確率)

+-----> ふんが米水の状態 60%

+-----> 皮膚に赤い湿疹 10%

+.....

(注意)事前確率<==公衆衛生学と社会医学:季節などに変化する。

事後確率<==臨床医学:医学上の定説。

疾患名と症状の因果関係は基礎医学(病理など含む)。

疾患名単位で入力された知識は、実際に推論をするまでには、ルール単位に変換されていることである。上記の疾患名の”コレラ”フレームは3つのスロット知識からなっていると解釈し、3つのルールに変換される。確信度は事後確率*0.01*事前確率として計算する。上記の例としては熱が39度以上あればコレラの疑いは $0.6 (=60 \times 0.01 \times 1.0)$ という1ルールになる。1ルールは条件部、結論部は一つになる。

4.2 フレーム単位の知識編集

疾患名単位をグループ知識(あるいは”META”)として次のように入力をする。

知識ベース編集(グループ知識)

- | | |
|--|----------------------|
| 1. グループ知識の名前 | : YIKB |
| 2. 結果(結論)(英数字10文字以内) | : cholera |
| 3. 発生頻度(事前確率) | : 1.0 |
| 4. コメント(日本語30文字以内) | : 冷凍の魚介類の輸入品 |
| 5. 参考文献(日本語30文字以内) | : 医学概論(at page 120) |
| ***条件(日本語20文字以内:”主語”+”は”or”が”+”述語”)と 事後確率(1から100の間)*** | |
| 6. 条件: 熱が39度以上 | 確率: 10 |
| 7. 条件: 下痢がひどい | 確率: 45 |
| 8. 条件: 皮膚が赤い湿疹 | 確率: 30 |
| 9. 条件: ふんが米水の状態である | 確率: 55 |
| 10. 条件: | 確率: |
| 11. 条件: | 確率: |

4.3 知識ベースファイル (医療向けシェル)

フレームを格納する時に、グループ知識名の下位構造にグループ共通情報となる部分("META")とルール単位("RULE")に分けられる。ルール名は知識編集にて保存を指示すると自動的に"グループ知識名+数字"という名前で保存される。

下記に知識ベースファイル(^ZFKB)の構造を示す。

^ZFKB

| ★設定情報

| ->>^ZFKB(KBFLNM) :

| ①KEY : 知識ベースファイル名 VX(8) (標準名として"DEFAULT")

| ★ : 知識ベースファイル情報

| ->>^ZFKB(KBFLNM, METANM) :

| ②KEY : 個々のMeta知識の名前 VX(6)

| グループ知識名を^ZFKBでは、メタ知識名(METANM)と

| ->>^ZFKB(KBFLNM, METANM, "META", "NMDATE")

| ③KEY : 作成定義1 FA(6)

| ★ : 作成定義1情報

| ->>^ZFKB(KBFLNM, METANM, "META", "COMENT")

| ③KEY : 作成定義2 FA(6)

| ★ : 作成定義2情報

| ->>^ZFKB(KBFLNM, METANM, "META", "KBBODY")

| ③KEY : 作成定義3 FA(6)

| ★ : 作成定義3情報

| ->>^ZFKB(KBFLNM, METANM, "RULE", RULENM)

| ③KEY : 個々のルール知識の名前 VX(8) (METANM+

| ★ : ルール知識の内容 +"1" ---> "n")

5 おわりに

今回は異なる(フレームとルール)知識表現にたいして、同じ推論機構が動作したことを報告した。エキスパートシステムの利用者からみると、自分の専門分野での知識・情報がコンピュータの表現と自分の頭の中の表現との差が少ないのがよいことである。本報告にて、利用者が汎用の知識表現やドメイン知識表現を選択して利用すればよい事が提案できたと思う。汎用シェルは医学部学生の91・92年度"知識工学"の授業の演習として採用し、医療向けシェルは同じく93年度の授業演習として採用予定でいる。次の課題としては医療向けシェルの評価をすることと知識表現の中の日本語表現の類似性の認識をどう扱うかである。

参 考 文 献

1. Richard Fikes and Tom Kehler : "The rule and Frame-based Representation in reasoning" , Communications of the ACM , September 1985 , Volume 28 , Number 9 , pp904-920(1985)
2. M.Minsky : "A Framework for Representing Knowledge , n the Psycology of Computer Vision" (McGrawHill) (1975)
3. Koyama T. : "An Automatic Rule Generation and Evaluation Schema for AI Medical consultation systems" , 3rd Medical Informatics , pp1306-1310(1980)
4. 今泉幸雄 他 : " 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 , pp929-930(1985)
5. 今泉幸雄 他 : " MUMPSによるデータ解析のエキスパートシステム" , PCworld , 1988 11月号 , pp68-74(1988)
6. 今泉幸雄 : " 知識の矛盾抽出と修正法 -時制論理の考察-" , 第15回日本MUMPS学会(名古屋) , pp153-159(1988)
7. 今泉・大櫛 : " エキスパート・システムにおける知識表現と推論機構" , 第18回日本MUMPS学会(伊勢原) , pp133(1991)

M言語よりのPC版SASシステムの利用

M Language and PC-SAS

田久浩志¹、今泉幸雄²

Hiroshi Takyu, Yukio Imaizumi

DTMで作成したデータベースを用い、PC版SASシステムでデータ解析をする場合の有効性についての評価をした。そのため、M言語で検索したデータをシーケンシャルファイルに出力した後SASでデータを読み込む場合と、SASデータセットでデータを検索する場合の処理時間を比較した。1種類のファイルを用いる場合、2種類のファイルを結合する場合には、データ件数によってはM言語とPC版SASシステムを使用する方が処理時間が早かった。これらの事より、数万件のデータより一部を検索して解析する場合、ディスク容量に制限がありソートがしにくい場合にM言語とPC版SASシステムを用いると有効であると考えた。

(Keywords: PC-SAS, Data Tree Mumps)

1 目的

今回、福岡県の医療圏設定を評価する試み（患者数約40万人、医療施設数約4千）を行うにあたり、M言語(Data Tree Mumps)からSAS(PC版SAS)を用いる環境を、解析の補助手段として検討することになった。SASと多くのデータベース間にはSAS/ACCESS等のインターフェースをとる製品があるが、M言語環境とのインターフェースは存在しない。本論文では、IBM-PS55上でPC版SASとM言語を使用した時の、データの検索処理速度の評価について報告する。

2 M言語とSASのインターフェース手法

M言語（以下Mで示す）とSASとを利用する方法は幾つか考えられる。以下にMとSASとのインターフェースをとる方法について検討する。

A. SAS側でSAS/ACCESSを使用し、M側でSQLを実行する製品を使用する方法

既に、Mの上ではM/SQL等のSQLを解釈する製品がある¹⁾。この手法は、SAS,Mの両方で新規のプログラム開発の必要はないが、新規のソフトウェアの購入が必要となる。

1 東邦大学医学部病院管理学研究室

E-Mail Address Nifty-Serve MAF00072

Internet takyu@tansei.cc.u-tokyo.ac.jp

2 サンド薬品株式会社

〒300-33 茨城県つくば市大久保8

TEL 0298-65-2205

1 Department of Hospital Administration, School of Medicine,

Toho University

5-12-16, Omori-nishi, Ota-ku, Tokyo 143, Japan

2 Sandoz Pharmaceuticals Ltd.

Ohkubo 8, Tsukuba-shi Ibaraki, 300-33, Japan

TEL 0298-65-2205

B. SASでOMI(Open MUMPS Interconnect)を用いる方法

異機種、異装備のMUMPSアプリケーション間で、リモートデータベースを実現するためのネットワークプロトコル、Open Mumps InterconnectをSAS側で実現する手法である。M側よりSAS側にデータを転送してデータセットを作成し、その後に処理を行う考えである。この場合、SASでの新規プロダクトの開発が必要となる。

C. インターフェース用プログラムを開発する方法

アセンブラ、C等でM側のグローバル変数を取り込むインターフェースプログラムを記述する方法である。既にAccess File Description, Master File Descriptionの2つの定義ファイルを用いてFOCUSとDSMの間のインターフェースにこの手法が採用されている^{2,3)}。処理速度の上昇は期待できるが、M言語の動くプラットフォーム毎の新規プロダクトの開発が必要となる。

D. シーケンシャルファイルを経由する方法

M側のプログラムでSASのプログラムとデータを作成した後、一度シーケンシャルファイルに出力し、その後、SASを起動してプログラムを実行する方法である。どのM言語の動く環境でも実現可能であり、SAS側で新規の開発が不必要である。その反面、M言語の終了時間、SASでのデータの読み込みまでのオーバーヘッドが大きい短所がある。今回は、PC版SASシステムを用いることより、Dの外部ファイルを経由する方法を採用した。MでSASのプログラムを生成するプログラムは別途作成した。

3 M側でのSASプログラムの生成

SAS側のPROCステップをデータ解析エンジンとして使用することを考慮し、まず第一段階としてSASにおけるCHART, PLOT, PRINT, FREQ, CORR, MEANS, SUMMARY, UNIVARIATE, TABULATEの各PROCステップを利用することを計画した。そのためには、M側よりSASのプログラムを生成するルーチンを記述する必要がある。しかし、SASランゲージを使用したプログラムは、非常に奥が深いものがあるので、第一段階では最小限の機能を用いれば良いと考え、もし必要があれば生成されたプログラムを修正して使用すると考えた。これらを考慮して、プログラムとして%UU\$asを作成した。これはローカル変数に必要なSASランゲージを文字列として代入して、対応するプロシジャ名のルーチンをよべば、必要なSASプログラムを生成するものである。一方、PROCステップと異なり、外部ファイルを入力して、SASデータセットを作成するプログラムでは、読み込む変数の順番を指定しなければならない。そこで、データ入力プログラムの生成に関しては、ローカル変数に、読み込む変数の順番に対応して、変数名、FORMATに関するSASステートメントを代入するものとした。また、SAS側にわたすデータはカンマで区切った形式とし、SAS側でINFILE指定により読み込むものとした。

例：

```
M側 S A="MYDIR.BUF",A(1)="ID",A(2)="NAME $",A(3)="CLASS"
    S X=$$data^%UU$as(.A)
```

SAS側 data MYDIR.BUF;

```
infile 'out.dat' dlm="," missover ;
input ID      NAME      $      CLASS
;
run;
```

4 処理速度の比較

SASとMで、データを検索して新規のデータセットを作成するまでの速度を検討するため、福岡県の京筑地方の医療圏調査で利用した患者8460人と医療機関99施設のデータを用いた。患者のデータは、施設番号、患者番号等の55変数よりなる。1データあたりの容量は平均132byteである。医療機関のデータは施設番号、開設者の種類、標榜科目等全体で14変数、116byteよりなる。事前に施設毎の患者データは、100, 200, 500, 1000, 2000, 4660件に設定した。このファイルを2回使用し、16920件のデータも用意した。

使用した機械はIBM PS55/NOTE N23x, Memory 6 Mbyte, Hard Disk 80 Mbyteである。DOS/V上にQEMM-386とTRを搭載し日本語の使用できる環境としてPC版SASのVer.6.04を使用した。M言語はData Tree Mumps (以下DTM) Ver.4.3を用いた。

残念ながら、現段階ではDTMの内部よりSASのプログラム起動する手立てはない。そこで、DTMとSASを用いる処理時間の測定には、以下の手順でDTMでの検索とSASのプログラム実行を連続して行うバッチファイルを作成し処理時間を求めた。

```
DTMの起動 グローバル変数のデータ検索プログラムを実行
              結果をシーケンシャルファイルへ出力
              SASで用いるデータ読み込みプログラムの生成
SASの起動 データ読み込みプログラムをバッチで実行
```

次にSASのみで行う処理時間の測定は、M側のグローバル変数と同じデータをSASのデータセットで作成しておき、それからIF文で該当するデータのみを検索し、新規データセットを作成する場合の時間を求め、上記の時間と比較した。

4.1 1種類のファイルでの処理速度の測定

患者データのみを用いて、データ検索を行う場合の検索時間を比較した。患者データは条件1-4は8460件、条件5-6は16920件である。

条件1 SASデータセット上でn件検索し、新規のSASデータセットを作成する。

条件2 グローバル変数上でn件検索し、一度シーケンシャルファイルに出力。

それをSASで読み込み、SASデータセットを作成する。

条件3 SASデータセットよりn件検索し3変数で、新規データセットを作成する。

条件4 グローバル変数上でn件検索し、3変数をシーケンシャルファイルに出力。

それをSASで読み込み、SASデータセットを作成する。

条件5 条件3に同じ、但しデータ件数は16920件

条件6 条件4に同じ、但しデータ件数は16920件

図1に各条件の比較を示す。8460件を対象とする場合、変数全てを用いる場合、検索結果の件数が1200件程度までは、DTM側で検索したほうが処理時間は早かった。実際の多重分割表を作成する場合、変数は3,4個の場合が多い。これに相当する条件3,4を比較すると4600件のデータ検索までは、DTM側で検索したほうが処理時間は短かった。条件3と5を比較すると、SASのみでデータ処理を行う場合、オブザベーションの数に比例して処理時間が増加した。条件4と6とを比較すると処理時間はほとんど変わらなかった。これはグローバル変数の検索の場合、キーの数に比例して処理時間が増加し、データ数全体とは関係しないため

ある。

4.2 2種類のファイルを横方向に結合する場合の処理速度

患者のデータに、その施設番号をキーとして横方向に医療機関のファイルを結合する場合の処理速度を比較した。結合するデータは、キーの他に住所、施設種類の2変数である。SASデータセットのみで処理を行う場合、一度患者データを施設番号でソートしてから医療機関のデータをマージをする必要がある。そのため、事前に医療機関のデータセットは施設番号でソートを行った。DTM側では、まず患者データより施設番号を求め、これをキーとして医療機関のグローバル変数を求め、患者データに必要な変数を結合した。処理の条件を以下に示す。患者データは条件7,8は8460件、条件9,10は16920件である。

- 条件7 SASデータセットよりn件検索し、3変数で新規のデータセットを作成。それを施設番号でソートの後、施設番号をキーとして、医療機関データより住所、施設種類の2変数をマージする。
- 条件8 グローバル変数よりデータをn件検索し3変数を得る。
施設番号をキーとして医療機関データより住所、施設種類の2変数を結合し一度シーケンシャルファイルに出力。そのファイルをSAS側で読み込み、SASデータセットを作成する。
- 条件9 条件7に同じ、但しデータ件数は16920件
- 条件10 条件8に同じ、但しデータ件数は16920件

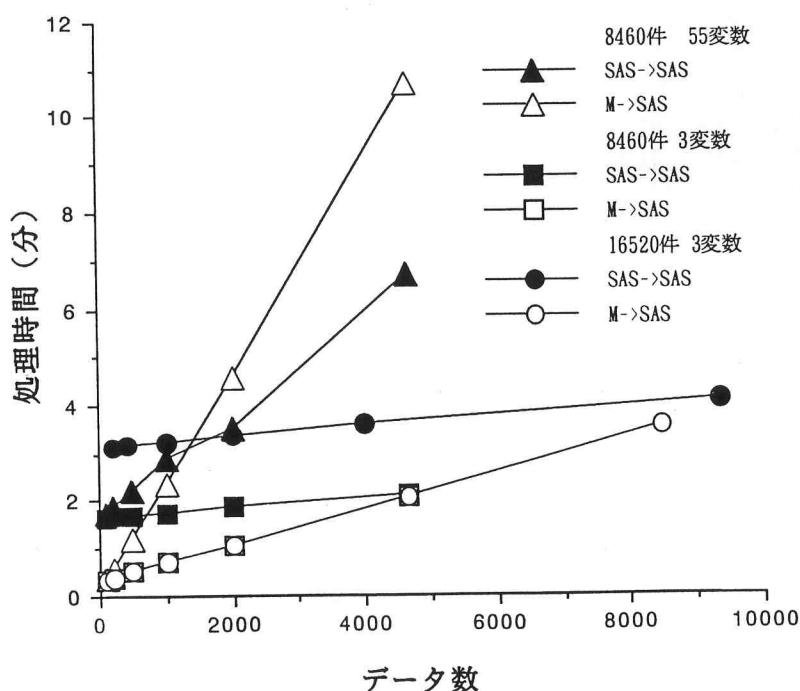


図1 1種類のファイルの処理時間

図2に各条件の処理時間の比較を示す。条件8と条件10で処理時間は同じである。どの条件でもデータ処理時間は、DTM側で一度検索したほうが早かった。

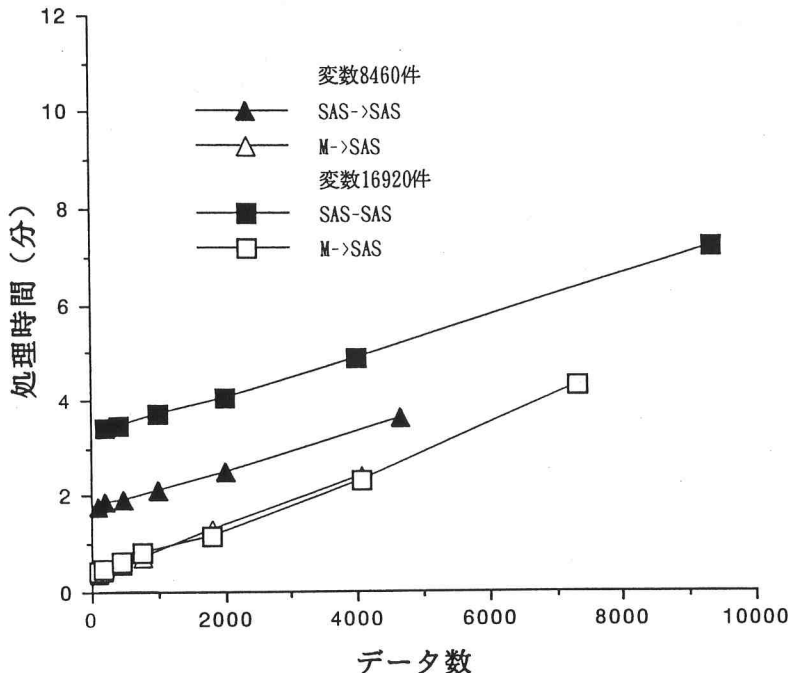


図2 2種類のファイルの結合処理時間

5 考察

今回の結果より、単一のファイルの使用では55変数全てを使用する場合、検索結果の件数が1200件程度までは、DTMで検索したほうが処理時間は短かった。使用する変数を3個に限定すると4600件のデータ検索まではDTMで検索したほうが処理時間は短かった。また2種類のファイルを結合する場合、8460,16920件のデータを対象とする場合はDTMで検索したほうが処理時間が短かった。これは、グローバル変数の構造とSASのデータセットの構造に起因するものである。SASデータセットはインデックスとファイル部分より構成され、その検索時間はデータ全体の数に比例してしまう。一方、複数のファイルを結合する場合、SASではソートとマージが必要となる。医療圏の調査では、施設数4000、患者数400000程度の大量データを扱う場合が頻繁に生じる。この事はディスクやメモリーの資源を豊富に使用できるVAX,UNIXなどの計算機環境では問題とならないが、PC版SASの動く環境ではかなり問題となる。

DTM,PC-SASの起動終了のオーバーヘッド、シーケンシャルファイルへのデータの入出力時間は、少量のデータではデータ処理効率に大きな影響を与える。しかし、大量のデータを扱う場合では、相対的に影響がすくなくなる。これらの事より、M言語とPC-SASを使用するような環境では、データの数が大量にある、使用する変数が少数で検索する対象が全体の一部の場合など、M言語の環境よりPC-SASを利用する価値はあると言えよう。また、VAX、UNIXなどの環境ではデータをM側からSAS側に渡す方法も、メールボックス経由などの高速な手段も考えられる。今後M側の資源を解析するのに、SASをデータ解析エンジンとして使用するのも有意義と考える。

6 まとめ

従来、M側でデータ処理を行う場合、その都度ルーチンを記述する必要があった。そのため、病院等で戦
略情報システムとしてMのデータベースを利用しても、重要な情報を迅速に提示するのは難しい問題であり、
誰でもが可能というわけではなかった。医療圏評価の解析をするにあたり、複雑な表を処理できる
TABULATE、多重分割表を生成するFREQなどは非常に有用であった。今後は、MとSASの長所を活かして
使用することが重要である。そのため、次の段階ではMでは不得手であるグラフィック機能を利用するため、
SAS/GRAPHの利用を考えている。

参 考 文 献

1. Jared S. Corman and Daniel Pasco: SQL and MUMPS, MUMPS Computing, Vol 22, No.1, P.24-27, 1992
2. Sebastian Holst: Integrating Digital Standard MUMPS(DSM) and the FOCUS 4GL, MUMPS Computing, Vol 22, No.1, P.29-31, 1992
3. Focus for VAX/VMS, Interface to Digital Standard MUMPS Release 6.1, FOCUS Information Builders, Inc. 1991

M と TP モニターとの統合の試み An Approach to Integrating M with a TP monitor

今井敏雄、佐藤真美
Toshio Imai, Mami Sato

MをM以外のシステムやツールと統合する際、データタイプの変換や、Mのダイナミックな特性をコンパイラ言語の静的な特性とどう折り合いをつけるかなどのインタフェース上の問題に直面します。この論文では、この問題に対するひとつの現実的な解決策として採用した第4世代言語アプローチについて述べます。統合の対象はACMS (Application Control and Management System)と呼ばれる弊社のオンライントランザクション処理モニターです。ある開発プロジェクトの設計フェーズで、ACMSシステムのフロントエンドすなわち画面処理部を、DSMとDSM Application Software Library(DASL)で処理することが決まりました。そのため、ACMSとDSMを統合するツールが必要となり開発を行いました。ツールは、「DASL for ACMS」と呼びます。このツールの概要について述べます。

(キーワード:TPモニター,4GL,DSM,DASL,ACMS)

While trying to integrate M into systems and tools outside, we will probably face some interfacing problems such as data-type conversion and how to meet the dynamic property of M with static one of compiler's language. This article describes a 4GL approach we adopted as a practical solution to the problems. The target for integration in our case is an OnLine Transaction Processing Monitor called DEC ACMS(Application Control and Management System). When a certain development project was in its design phase, we chose to employ DSM and DSM Application Software Library(DASL) as a front-end or presentation part of the system based on ACMS. We, therefore, needed and developed integration tools for ACMS and DSM, which we call "DASL for ACMS". We describe its outline in this paper.

(Keywords:TP monitor,4GL,DSM,DASL,ACMS)

1 はじめに

26年も前に誕生したMUMPSの実用性、使いやすさ、性能は、現在でもその魅力は十分あります。しかし、高性能PCの普及を契機とするコンピュータユーザ層の拡大にともない、GUIを駆使した使いやす

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 大阪府中之島2-2-2ニチメンビル
TEL 06-222-9211 FAX 06-202-1419

Digital Equipment Corporation Japan Digital Consulting
Nichimen Building, 2-2-2, Nakanoshima, Kita-ku, Osaka 530,
Japan
TEL 06-222-9211 FAX 06-202-1419

くて、高機能なものが世の中に出てきており、以前のような「MUMPSでなければならないんだ」という差別化された魅力と迫力が薄らいできていることも事実だと思います。Mが、今後も発展し続けるためには、M自身の魅力を高める努力と同時に、M以外の世界との関係を深め、その共存の中で生きていける力がMになればならないと思います。

MとM以外の世界との結合時に、Mで問題となるのは、以下の3点です。

1) タイプフリー

Mは文字列型のデータタイプのみで、データ定義のメタデータを持たない。そのため外の世界のデータタイプ/コードとの変換をどこでどう行うか？

2) 実行時評価

コンパイル型言語との統合時、そのスタティックな要素とMのダイナミックな要素をどう折り合いをつけるか？

3) シンプルフロー

GUIのイベント割り込みや、通信等での並行実行の非同期処理をどうハンドリングするか？

これらは、Mの世界の中だけで考える場合、Mの使い易さの源泉であり、大いなる特徴です。しかし、他の言語やシステムとデータや処理を分担し、統合利用する場合、逆に障害となります。この障害をなんらかの方法で、解決する努力をM側で行なわないとMは孤立化する恐れがあります。これに対し、MUMPS Development Committee(MDC)で議論されている成果(SQL、Windows、RPC、C等のバインディングやObject MUMPS)に大いに期待する所ですが、ジェネレータタイプの4GLアプローチで、この問題に対し、現実的な解決を試みた例がありますので、以下に報告します。

DSMとの統合の対象の外の世界は、弊社トランザクション処理システムのACMSです。

2 TPシステムとM

コンピュータシステムが社会の中核部分で利用されるようになればなるほど、より高性能で、かつ、信頼性の高いシステムが求められます。オンライントランザクション処理(OLTP)システムがそのひとつの解答です。弊社のApplication Control and Management System(ACMS)は、そのTPシステムです。

TPシステムは、通例、MのTime Sharing System(TSS)ベースのシステムとは全く異なる実行環境をもちます。それは、いわゆる、Client/Server環境、あるいは、Front-end/Back-end分散環境です。Mは、データベースアクセス手続も言語内部に取り込むという特異なアプローチで、プログラム言語とデータベースを一体化させ、さらに、端末アクセスも含めて、すべてをコンパクトに一体化させることで、最適化と使いやすさを図ってきました。これとは逆の方向、つまり、プログラム言語とデータベース処理を完全に分離するアプローチで最適化を図りAPIを標準化してきたのがRDB/SQLです。そのRDBをベースにするTPシステム(ACMS)は、更に、分離要素化アプローチを進めて、フロントエンドの端末処理部、バックエンドのデータベース処理/手続処理部、そして、それら全体を管理し、多端末/多トランザクションを効率よく処理するTPモニター部に機能分割しています。

このモジュール構造は、それぞれの要素の機能特化が図れると共に、中核モニター部がトランザクションと構成管理を行いますので、システムの高可用性(対障害性)、柔軟性と拡張性、データベースの一貫性とセキュリティ保持、協調分散処理等の達成に有効なアプローチです。しかし、Mのようにシンプルでないことは事実です。簡便コンパクトなMと、重厚モジュール型のTPシステムとの文化/発想の差を感じさせます。

Mは、その簡便さ(特に開発/保守性と性能)のメリットを保ちながら、TP処理や、Client/Server環境にどう適応/進化すべきかが、ひとつの大きな課題です。

今回、ある開発プロジェクトで、ACMSの端末処理パートシステムとしてDSMとDSM Application Software Library(DASL)を使用するため、ACMSとDSMとの統合ツールを開発しました。

ツール名称を「DASL for ACMS」と呼びます。

目的は、Mの特徴を生かし、端末/フロントエンド部の開発/保守生産性の向上と、TPシステムのフロント処理効率アップで、MがTPシステムのクライアントとなる試みです。

3 ACMSとAPI

ACMSは、タスク言語で定義した「タスク」の実行を行う「タスクサーバ」を制御核に、データベースアクセスを行う「プロセッシングサーバ」、端末処理を行う「プレゼンテーションサーバ」、および、タスクを呼び出す「クライアントプログラム」から構成されます。「クライアントプログラム」のひとつにユーザ作成のプログラムからタスクを呼ぶAPIである Customer-Written Client Program (CWCP)があります。CWCPを利用して、DSMとの結合を行います。DSM側の接点は、M標準規格化される外部ルーチン呼び出し External routine calling syntax (ECALL) を利用します。ECALLインタフェースプログラムはCで書きます。

ACMSの実装は、ポータブルなClient/Serverモデルを実現するため、Distributed Computing Environment(DCE)のRemote Procedure Call(RPC)をベースにしています。ACMSタスク言語コンパイラは、Cクライアントに対しては、対象タスクの引き数レコード定義を含むCヘッダーとクライアントスタブオブジェクトを生成します。クライアント側のプログラムであるDSMは、生成クライアントオブジェクトとのリンクを行います。

Mとの統合のために、インタフェース部で、タスクコール引き数レコード構造からフィールド切り出し/合成を行う処理、フィールド値のACMSサポートデータタイプ(OCTET, INTEGER, TEXT, NATIONAL TEXT, ARRAY, RECORD)とM変数(可変長文字列)の変換処理、外部(漢字)コード系と内部(データベース)コード系との変換処理等を行なわなければなりません。これらの処理を、アプリケーションタスク毎に、インタフェースCプログラムで書く場合、その数が多くなるとインタフェース部の開発、保守の負担が大きくなり、統合のメリットがなくなります。ここがMと他の世界との統合上、問題となるところで、Mがその世界に閉鎖的となる傾向の技術的素因だと考えます。この問題に対し、4GL(第4世代言語)アプローチでこの解決を試みました。

4 4GLアプローチ

ACMSとの統合で、採用した方式は以下のとおりです。

- 1) M側に、外部統合に必要なメタ定義情報を取得保持する。
- 2) 1)の外部メタデータをM側のメタデータにどう対応づけるかのマッピング情報を持つ。
- 3) 1)、2)をもとに、Mから外部呼び出しをするMドライバルーチンおよびインタフェースCプログラムを自動生成する(4GL)。

このことにより、Mアプリケーションは、生成Mルーチンを呼ぶだけで、あたかもM中のサブルーチン呼び出しのように外部アクセスが出来るようになります。外部世界のレコード定義分解/合成、データタイ

プ/コードの変換、外部エンティティデータとMエンティティデータのマッピング、外部呼び出しシーケンス/エラー等の統合にかかわる処理は、生成ドライバルーチンがハンドリングしますので、その詳細をアプリケーションから隠蔽し、アプリケーションの抽象度を高めることができます。また、アプリケーションとE CALLおよび外部の間に、ドライバ緩衝層を設けますので、結合下層/環境の変化、技術進歩に対しても、4 GLの進化で柔軟に対応できます。

ACMSとの統合で、上記アプローチを、DASLのオプション拡張として実現します。DASLは、DSM上の4 GLで、スクリーン/レポート/検索のDSMルーチンを生成します。また、DSMの標準開発実行環境を提供します。DASLには、グローバルデータ構造をはじめ各種メタデータ定義辞書を持っています。外部世界の定義情報をその拡張として保持することが自然と考えました。

5 DASL for ACMS の概要

ACMSの端末プレゼンテーションパートとして、DASL/DSMを有効に使うため、ACMSとDASLとの間のインタフェースプログラミングの開発生産性/保守性/品質を上げることを目的とするツールです。

DSMルーチン、DASLアクション部から、ACMSへのアクセス(タスクコール)が、サブルーチン呼び出し、

```
DO CALL^%DAACMS(group,task)
; group=タスクグループ名、task=タスク名
```

でアクセスできるようにします。以下の3つの機能が、主機能です。

1) タスク定義取り込み

ACMSタスク言語アプリケーションソースプログラムからタスク呼び出し情報(レコード定義、タスク引き数定義)を抽出し、DASLのタスク定義辞書データベースとして保持する機能。

2) タスクコール定義

ACMSタスクコール引き数レコードの各フィールドを、DASLのデータネーム(エンティティ名)やスクロールアレイ、DSM変数にどうマッピング(対応づけ)するかを登録する機能。このマッピング情報は、ACMS側とDASL側のインタフェーステーブルの役割をします。この仲介マップ情報を持つことにより、M側アプリケーションでは、DASL/DSM内の変数でタスクコールのデータを引き渡すことができ、ACMSとDASLとのエンティティ名の独立性と、関係の柔軟性を保証します。

3) タスクコールコンパイラ

1)、2)の情報をもとに、

```
DO CALL^%DAACMS(group,task)
```

で呼ばれる、ACMSタスクコールインタフェース(DSM)ルーチンを生成する機能。タスク定義、マッピング情報の変更に対しても再コンパイル処理で、インタフェースルーチンを再生成でき、インタフェース部プログラムの開発/保守性が向上します。

図1に、DASL for ACMSの全体構造を示します。●が主要構成部です。

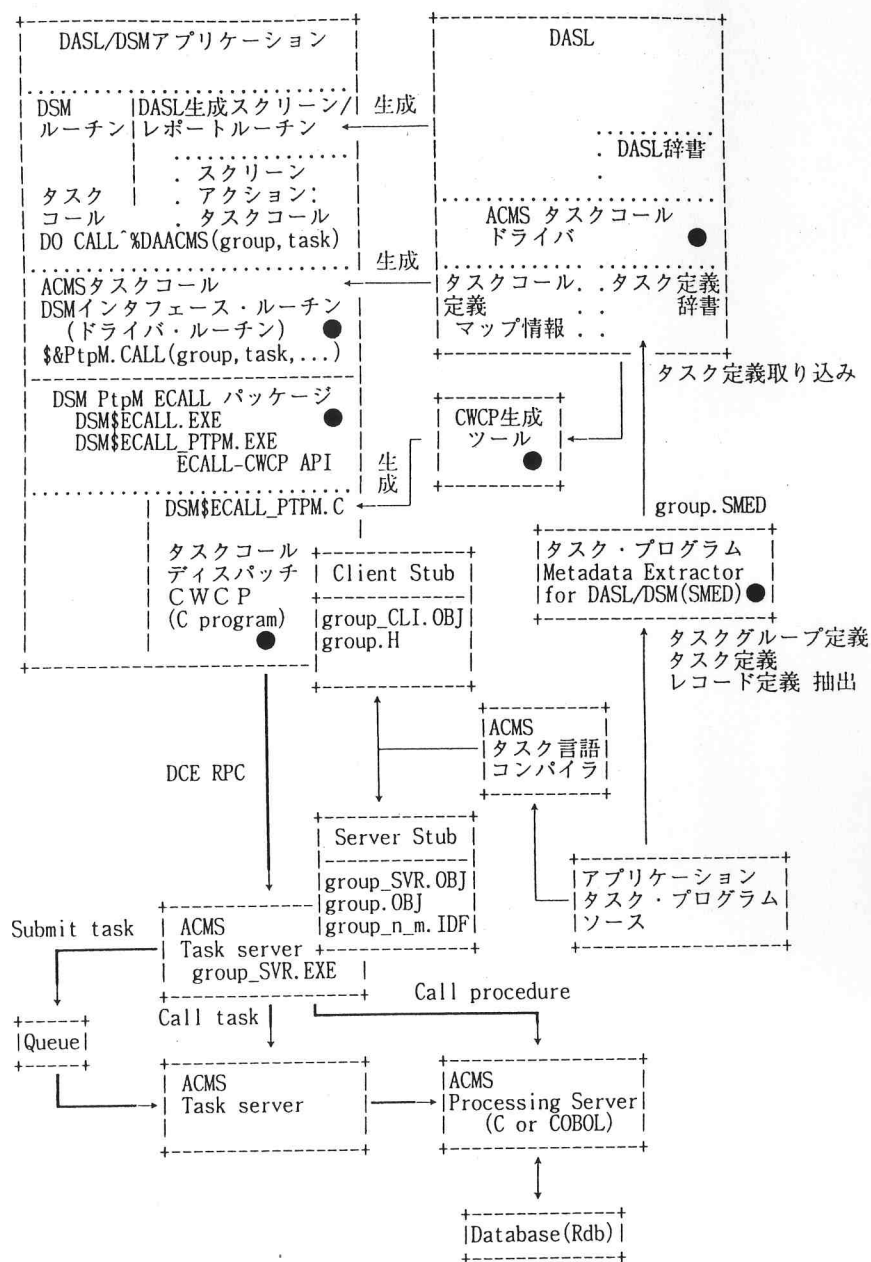


図1 DASL for ACMSの全体構造

6 おわりに

M と M 外の世界との統合上の試みとして、4 G L アプローチによる事例を紹介しました。人間がある一定時間にタイプライン出来る量が限られている限り、プログラムの生産性を上げるには、少ないコマンドで多くのことが出来る必要があります。ひとつは、既にあるプログラムの再利用(サブルーチン)であり、ひとつは、上位抽象コマンドが多くのことを実行してくれる 4 G L アプローチです。この 4 G L アプローチの中で、M 以外のシステム／ツールを統合利用する今回の試みは、M の世界を広げるひとつの方向だと考えます。この種の試みは M の純血性を損なうかも知れません。しかし、なんでも自前でやるにはそのパワーが必要で、M はそれができるか、それが賢明なのか問われています。M は他との共存の中で、その特異性とテクノロジーを広め、深めて行くべきと考えます。そして、その時こそ、M の真価が問われる時だと思えます。

文字検索による日本病理剖検輯報日本語データベースシステム Information Retrieval system for the *Bouken Shuho* by Japanese Word Keys

馬場謙介¹、黄聰乾¹、藍沢繁雄²

Kensuke Baba, Soken Koh, Shigeo Aizawa

Since 1958, briefly described autopsy documentation through Japan are published annually in the Annual of Pathological Autopsy Cases in Japan (*Bouken Shuho*), of which presswork has been assisted by a small computer since 1989. The work file which was used in that presswork and written in Japanese characters for the publication is transplanted into an authors' personal computer (PC-H98/NEC) within the cases autopsied in 1989. Analyzing the printing format of the file by the control characters in it, the Japanese data were segmented into institution autopsied, into each case, into each item, and into each findings. Correction of miss-spellings (Table 1a), normalization of Japanese characters (Table 1b), and elimination of adjective (*exp*:slight, marked etc. in Japanese), therapeutic (*exp*:postoperative, after irradiation etc.) and quantitative (*exp*:1000ml, 2cm etc.) descriptions (Table 2a,b,c) were performed interactively and/or automatically with specially designed editing system (coded by SP-MUMPS/SUMIDEN). Concurrently this system produces four kinds of files; full document file, simplified document file, natural to code word dictionary file (main dictionary) and logical dictionary files to be used to enhance data in document files. Though this work authors found that the simplified document file and the main dictionary (like Table 3a,b) are helpful for document retrieve of the autopsy cases. However, also we experienced that, even if main dictionary have no target key words, we can find out retrieve logic having higher hit rate than that of classical numerical code based autopsy database (Table 4a,b). It is emphasized to establish the more general way to design the retrieve logic and the tools to help the designing as the conclusion of this study.

(Keywords: Autopsy, Japanese, Retrieve, MUMPS)

1 はじめに

日本病理剖検輯報の電算写植機に入力するためのワープロデータ(1989年度剖検分)を半自動的に所見(疾患)単位に分解することに成功し、これを従来の疾患コードキーによる検索システム(1-4)とリンクすることによってもとのドキュメントを電子的に提供できるようになった。しかし、従来のコード化の作業が、(1)「コード化の精度が高くないこと」、(2)「コード化の対象が疾病に偏重していること」から、従来のコード化に依存しないシステムが望まれる。そこで、文字情報をキーとする剖検輯報掲載症例を検索することを試行し、成功したので報告する。

1 国立埼玉病院臨床研究部

〒351-01 埼玉県和光市諏訪2-1

TEL 048-462-1101

2 慈恵医大病理

1 Dept. Clinical Research, National Hospital, Saitama,

2-1 Suwa, Wako, Saitama 351-01, Japan

TEL 048-462-1101

2 材料と方法

データの移植と症例項目所見の分離:

1989年次剖検分の日本病理剖検輯報の電算植字に用いたフロッピーディスクのデータを解読して、著者のパーソナルコンピュータ(H98, 1G光ディスク/NEC)に移植した。移植した印刷書式のデータをSP-MUMPS(国際標準/日本語標準/住友電工)を用いた自作のプログラムにより、施設別・症例別に分解し、不要な印刷制御文字などを削除した。更に、各症例を項目(剖検番号・住所・年齢・性・職業・臨床診断・出所・剖検所見・備考)に分解し、同様に不要な印刷制御文字などを削除した。つぎに、剖検所見は、その記載規約に従って、主病変、副所見、死因を識別し、その夫々を疾病単位に分解した上で、データベースに格納した。各症例のデータは、項目間を「\$」、病理所見間を「^」で区切って(256バイト未満の)可変長の構造にした。

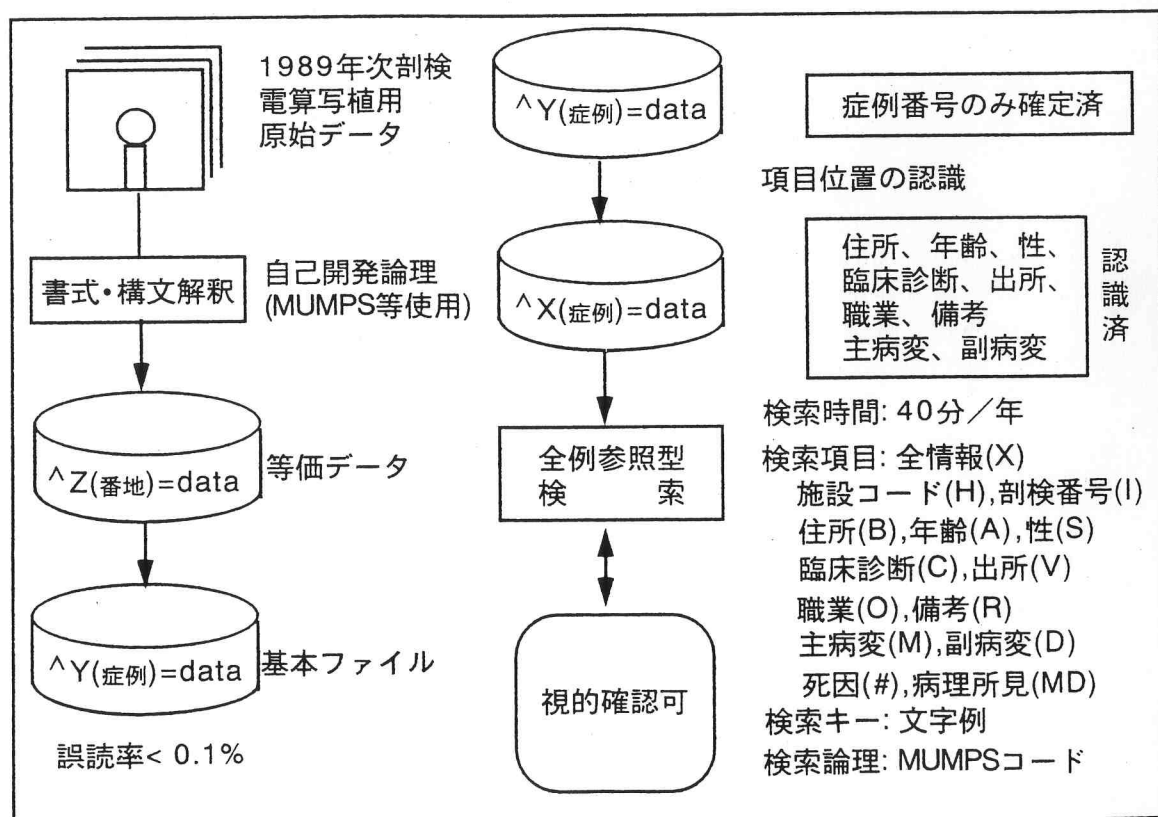


Fig.1 Flow of procedure

Table 1a Miss-spelling (Exp.)

多発性脳梗→多発性脳梗塞	
多発性脳塞→多発性脳梗塞	
嵌屯→嵌頓→嵌頓	
総胆管→総胆管	末梢→末梢
心衰炎→心囊炎	硬塞→梗塞
為膜性→偽膜性	窒息→窒息
栗粒性→栗粒性	表弱→衰弱
機質化→器質化	内蔵→内臓
硬塞→梗塞	腫張→腫脹
懐疸→壊疸	透折→透析
未熟→未熟	表弱→衰弱

Table 1b Normalization (Exp.)

そううつ病→そう鬱病→そう鬱病→躁鬱病	
びまん性→び慢性→瀰漫性→瀰漫性	
タンパク→タン白→たん白→蛋白	
るいそう→るい瘦→羸瘦→瘠瘦	頸→頸
クモマク→くもまく→蜘蛛膜	鬱→鬱
ギ膜性→ぎ膜性→偽膜性	總→総
胆のう結石→胆嚢結石	眞→真
うつ血→うっ血→鬱血	ゆ着→癒着
胆汁うっ滞→胆汁鬱滞	のう腫→嚢腫
骨粗そう症→骨粗鬆症	乳ビ→乳び→乳糜
びらん→び爛→糜爛	胆のう炎→胆嚢炎

文字の変換と修飾語の排除:

自動的にコード化するには、高度の「用語シソーラス」と高度の「解説文法論理」を要するが、そのいずれも現存しないし、間違え易い文字の正誤語彙辞書もない。そこで、まず、使用文字を Table 1a に例示したごとく統一した。この正誤語彙辞書は、ほとんど試行錯誤や偶然の発見を蓄積したものである。この文字変換の論理の多くは単純で、この変換は全自動的に変換し、以後の処理では元の(誤った)文字を問い返せないようにした。

独立概念の分離と情報の簡素化等:

第一に、病変の程度などを修飾する語句を削除して(Table 2a)、検索向きに表現を簡素化した。つぎに配慮したことは、例えば、「右」は「右胸心」等があるので一律の変換は避けることである。Table 2a の最下段に示した並列記載の処理は面倒であった。例えば「胃,食道潰瘍」を「胃潰瘍」と「食道潰瘍」の二つの所見にする処理がそれである。更に、因果関係の記載の分離は困難を極めた。次に、「術後肝炎」と「胃癌術後」の「術後」の意義

の相違にも配慮した。同様に、「移植後」、「蘇生器使用后」等にも配慮した(Table 2b)。最後に、Table 2cの如く論理的に数値等のデータの排除を配慮した。

但し、病期の情報、「疑」等の精度の情報、「局所解剖」等の情報については、今回は配慮しなかった。

Table 2a Eliminated Adjectives (Exp.)

種別	処理	例
程度	自動	高度の、高度な、(高度)、(高度)、著名、中等度の、軽度の、軽微な
分布	注1	弥漫性、び漫性、びまん性、散在性、多発性、個在性、限局性
側性	注2	左側、右側、両、左、右、両側
並列	対話	および、及び、及、ならびに、並びに、並、と、兼、「・」、「-」

注1: 多発性硬化症等を配慮に入れる。

注2: 左室肥大・右胸心等を配慮に入れる。

Table 2b Therapeutic Words (Exp.)

種別	実例
手術	切除術後、摘出術後、摘除術後、形成術後、術後、術後再発
置換術	大動脈置換術後、人工血管置換、僧帽弁置換、人工弁置換
人工臓器	ペースメーカー装着、人工心臓装着、調律器装着、蘇生器装着
放射療法	放射線治療後、放治後、放治後の状態、部分照射後、照射後
化学療法	化療後、化療後の状態、化療後再発、治療後、治療急性転化
免疫療法	免疫療法後、免疫療法後再発
移植	骨髓移植後、骨髓移植、移植術後
合併治療	放射線化療後、術後化療、術後放治

Table 2c Essence of Logic to Eliminate Quantitative Adjectives (Example)

摘 要	消去論理の要点 (MUMPS言語により記述)
経 過(1)	1N.N."全経過"."半"."年"."半".N."ケ"."ケ"."月"."半".N."日"."の"."状態"
経 過(2)	1N.N."半"."年"."半".N."ケ"."ケ"."月"."半".N."日"."経過"."の"."状態"
計 量 値(1)	."重量"."約".1N.N1"g"."r"
計 量 値(2)	."右"."左"."両"."側"."約".1N.N1"L"."."
計 量 値(3)	."右"."左"."両"."側"."約".1N.N1"I"."."
計 量 値(4)	."右"."左"."両"."側"."約".1N.N1"ml"."."
計 量 値(5)	."右"."左"."両"."側"."約".1N.N1"mL"."."
計 量 値(6)	."右"."左"."両"."側"."約".1N.N1"cc"."."
計 量 値(7)	."右"."左"."両"."側"."約".1N.N1"輻"."."
計 量 値(8)	."大きさ"."直"."半"."周"."径"."長"."約".1N.N1"cm"
数 量	(数個、数ケ、数ヶ、数コ、多数、小数、少数)
死後時間(1)	"死後"."約".1N.N"時間"."の"."状態"
死後時間(2)	"死亡"."約".1N.N"日"."約"."約".1N.N"時間"."約".1N.N"分"."後"."の"."状態"
胎 齢(1)	"生後"."約".1N.N."日"."約".1N.N."時間"."約".1N.N."分"."の"."状態"
胎 齢(2)	"胎齢"."約".1N.N."ケ"."ケ"."月"."約".1N.N."週"."約".1N.N."日"
胎 齢(3)	"胎令"."約".1N.N."ケ"."ケ"."月"."約".1N.N."週"."約".1N.N."日"
胎 齢(4)	"在胎"."約".1N.N."ケ"."ケ"."月"."約".1N.N."週"."約".1N.N."日"

インターラクティブな処理:

修飾語の排除、独立概念の分離、情報の簡素化は、程度の差こそあれインターラクティブな処理(対話処理)を伴った。この処理には、特殊な機能を付加した編集システムを考案⁵⁾して使用した。

データベースとこの編集システムを結合して、インターラクティブに

問題の情報(語句)が、

1. 語句辞書に存在すれば、'処理終り'
2. 語句辞書に存在しなければ、論理を見直し
語句辞書に存在すれば、'処理終り'
3. それでも、語句辞書に存在しなければ、
語句辞書を更新

を繰り返して、Simplified Database を生成するとともに、「語句辞書」と「論理辞書」を生成更新した。この行程を経て、より処理能力の高い「語句辞書」と「論理辞書」を暫時作りあげた。

Table 3a Brain Infarction (Examples)

右視床梗塞・右小脳梗塞・右側頭葉出血性梗塞・右大脳梗塞・右大脳半球梗塞・右脳梗塞・急性脳梗塞・旧脳梗塞・橋の梗塞・橋梗塞・橋底梗塞・橋底部梗塞・(局)橋梗塞・(局)脳梗塞・広範脳梗塞・出血性脳梗塞・左後頭葉出血性梗塞・左大脳梗塞・左内包梗塞・左脳梗塞・左被殻梗塞・視床梗塞・小型脳梗塞・小脳梗塞・小脳梗塞部・小脳出血性梗塞・小脳陳旧性梗塞・新旧脳梗塞・新鮮脳梗塞・新鮮脳梗塞症・前頭葉梗塞・巣状脳梗塞・側頭葉梗塞・多発性出血性脳梗塞・多発性陳旧性脳梗塞・多発性脳梗塞・多発性脳梗塞症・多発性脳硬塞・多発陳旧性脳梗塞・多発脳梗塞・大脳橋梗塞・大脳梗塞・大脳出血性梗塞・大脳小梗塞・大脳陳旧性梗塞・陳旧性小脳梗塞・中脳梗塞・陳旧性出血性脳梗塞・脳梗塞疑・陳旧性多発性脳梗塞・陳旧性多発脳梗塞・陳旧性脳梗塞・陳旧性脳梗塞症・陳旧性脳硬塞・陳旧脳梗塞・内包梗塞・脳の出血性梗塞・脳幹梗塞・脳幹部梗塞・脳橋部梗塞・脳梗塞・脳梗塞開頭・脳梗塞後遺症・脳梗塞症・脳梗塞巣・脳硬塞・脳出血梗塞・脳出血性梗塞・脳小梗塞・脳小梗塞巣・脳小梗塞多発・脳巣状梗塞・脳多発梗塞・脳多発性梗塞・脳内陳旧性小梗塞巣・脳皮質梗塞・脳微小梗塞・被殻梗塞・微小脳梗塞・微小脳硬塞・貧血性脳梗塞・両側脳梗塞

多発性脳梗塞*のように誤植と確信できるものは、「語句辞書」に含めて吸収した。

Table 3b Amyloidosis (Examples)

2次性アミロイドーシス・アミロイドーシス・アミロイドーシス・アミロイド症・家族性アミロイド症・家族性アミロイドーシス・家族性アミロイド症・家族性全身性アミロイドーシス・原発性アミロイドーシス・原発性アミロイド症・原発性全身性アミロイドーシス・続発性アミロイドーシス・続発性アミロイドーシス・続発性アミロイド症・続発性全身性アミロイドーシス・続発性全身性アミロイド症・二次性アミロイドーシス・二次性アミロイドーシス・アミロイド沈着・アミロイド沈着症・全身性アミロイド沈着・全身性アミロイド沈着症・類澱粉症・2次性類澱粉症・類澱粉症沈着症・家族性類澱粉症・家族性類澱粉症沈着症・原発性全身性類澱粉症沈着症・原発性全身性類澱粉症・続発性全身性類澱粉症・続発性全身性類澱粉沈着症

3 結 果

実際の1989年次1年分のデータで生成された「病名辞書」を例示すると、Table 3a,b の如くであった。Table 3a の表題の「脳梗塞」が、この表に掲げた用語の上位概念である。表題の「脳梗塞」は、又(数字ではないが定義された)コードであり、これに対して表中の用語ならびに「脳梗塞」は、「脳梗塞」を表わす自然語コードとも云える。

1989年次1年のデータから、目的の例えば、「脳梗塞」とその下位概念の疾患を持つ症例を拾い上げるのに要した時間は、22分であった。「脳梗塞」や「アミロイド症」は辞書を持つが、辞書を持たない疾患を検索する場合には、検索論理を記述しなければならない。検索論理は検索の目的によって様々であるので、実際に目的の結果が得るまでの時間は一定しないが、2時間以上を必要とする検索は減多になかった。

Table 4a Hit Ratio (Thyroid Gland Malignancy)

Method	Hit
Classical method (numerical code database)	690(99.8%)
Present study with no secondary retrieve	651(94.2%)
Present study with secondary retrieve	691(100%)

(690は正しくは、690疾患)

Table 4b Secondary Retrieve (malignant neoplasms of the thyroid gland)

甲状腺乳頭癌	papillary carcinoma, thyroid	10 cases
甲状腺乳頭状癌	papillary carcinoma, thyroid	5 cases
甲状腺潜伏癌	latent carcinoma, thyroid	4 cases
甲状腺濾胞腺癌	follicular adenocarcinoma, thyroid	3 cases
甲状腺濾胞状腺癌	follicular adenocarcinoma, thyroid	2 cases
甲状腺Latent癌	latent carcinoma, thyroid	2 cases
甲状腺左葉乳頭状腺癌	papillary ca., thyroid, lt lobe	1 case
甲状腺乳頭腺癌	papillary adenocarcinoma, thyroid	1 case
甲状腺(右葉、乳頭状癌)	thyroid cancer (rt lobe, pap. ca.)	1 case
甲状腺ろ胞状腺癌	follicular adenocarcinoma	1 case
甲状腺硬化癌	scirrhous carcinoma, thyroid	1 case
甲状腺潜伏性硬癌	latent scirrhous carcinoma	1 case
甲状腺潜在性微小癌	latent minor carcinoma, thyroid	1 case
甲状腺ラレント癌	latent cancer, thyroid	1 case
甲状腺扁平上皮癌	squamous cell carcinoma, thyroid	1 case
甲状腺腫瘍(原発悪性リンパ腫)	thyroid neoplasm (lymphoma, prim.)	1 case
甲状腺原発免疫芽球性リンパ節症	IBL, thyroid primary	1 case
甲状腺好酸性細胞癌	eosinophilic cell ca., thyroid	1 case
甲状腺左葉の乳頭癌	pap. ca. of thyroid left lobe	1 case
乳頭状腺癌(甲状腺)	papillary adenocarcinoma (thyroid)	1 case
total		40 case (20 different words)

辞書を用いない検索の例として、「甲状腺悪性腫瘍」の検索方法と検索結果を述べる。甲状腺悪性腫瘍の検索の目的で次の論理を仮に設定して検索してみた。即ち、検索論理を

「主病変・所見に「甲状腺癌」、「甲状腺腺癌」、「甲状腺乳頭状腺癌」、「ろ胞状腺癌」、「甲状腺潜在癌」、「甲状腺不顕性癌」、「甲状腺微小癌」、「悪性リンパ腫(甲状腺)」、「甲状腺悪性リンパ腫」、「甲状腺髄様癌」、「甲状腺原発悪性リンパ腫」のいずれかを含む」

として、仮に検索したところ、651例を得た。しかし、この検索論理で目的の「甲状腺の悪性腫瘍」が全部拾われたとは限らない。そこで、検索論理の正しさを検討する目的で、上記651例以外で「甲状腺」を含む症例を拾い上げ(599例)、その中から「甲状腺の悪性腫瘍」を目で拾い上げたところ(補助検索)、コード化による従来の方法にまさる検索ができた(Table 4a)。補助検索で新たに「甲状腺悪性腫瘍」の下位語彙に新たに20語が登録された(Table 4b)。

でき上がったデータベースの大きさは、全体で、約20Mバイトであった(内訳Table 5 参照)。ID(施設・解剖番号)をキーに任意の1症例の全文章を取得するに要する時間は、平均 1/1000 秒未満であった。検索論理のコーディングに要する時間は、検索目的によって様々であったが、テストランを含めて平均して、2時間程度であった。また、テストランにあたっては、従来の所謂「剖検輯報データベース」による検索件数のが参考データとして有益であった。

Table 5 Specification of the Database

Size of Database	40 M-byte
Full Document File	(12.6 MB)
Simplified File	(4.4 MB)
Dictionary File	(1.7 MB)
Logic File	(0.7 MB)
Work, Free Area etc	(20.6 MB)
Document File Access Time*	<1msec

*: Random access by ID.

4 考 察

日本病理学会では、剖検所見をコード化して電子計算機を用いて剖検診断を集計したうえで¹⁾、日本病理剖検輯報の巻末にその結果を掲載してきた。この電子計算機による集計処理中に生成された作業ファイルは、「病理剖検輯報データベース」と呼ばれ¹⁻⁴⁾、疾病の疫学研究に利用されてきた⁶⁻¹⁶⁾。しかし、(1)検索キーがコード化されていること、(2)原文の参照が困難であること等々の欠点があった。そのため剖検輯報が持つ情報を充分には活用できなかった。

本方法は、上記(2)については、対象症例の全情報を文字として(ほとんどプリンターの速度で)印刷できる程の優れた性能をもっている。本法は(病理剖検データベースにはない)臨床診断・職業等の情報を持つので、これらを含む研究ができる。しかも、磁気化されているので、文字データを検索論理に組み込むこともできるし、即座に印刷することもできる。さらに、データが(箇条書ながら)文章で表現されているので(文脈から)、単語自身にない意味を引き抜くこともできる。例えば、コード化による従来の病理剖検データベースでは不可能であった「胃癌で死ななかった胃癌」の検索に成功し、既に報告した¹⁷⁾。

剖検輯報に用いられる用語のシソーラスを作ることにより、コード化に頼らず且つ一般性のある検索システムができる。しかし、一般性は多少欠くもののシソーラスが完成していなくても、(甲状腺悪性腫瘍の処理で例示した如く適当な補助検索を探すことによって)高い検索率で検索することができることがわかった。

現在、本システムが持つデータは、1989年次剖検分の約38531件に過ぎない。今後、症例を追加して行きたい。データの追加によってデータ量が増えたときには、現在の処理速度ではやや不十分と予想されるので、今後は超大量データの検索に配慮が必要であろう。電算植字が導入されていなかった時代の剖検輯報の印刷

文字を光学文字読取装置(OCR)を介して電子計算機に取り込む試みがはじめられているが、経済的で且つ実用的な方法はみつかっていない18)。

5 む す び

シソーラス・語彙辞書があるに越したことはないが、シソーラスがない用語についても検索できることが判った。これが、今回の研究で最も意義のあることである。つぎに、検索の論理を設計して試行している内に、(二次的に)シソーラスが豊かになることを指摘したい。これは、「シソーラスなしに文章情報は検索できない」と考えた従来の考えを改めさせる事実として注目に値する。従って、現在まだ、検索論理設計の一般的・普遍的な方法が確立されていないが、より普遍的な検索論理設計とその道筋の体系化が肝要であろう。この体系化と並んで、(インターラクティブな処理に必要な)操作環境、検索処理環境、論理設計環境の統合的な整備も今後の開発の中心課題となろう。

文 献

1. Kimura,S & Baba,K: Frequency tabulation of pathological findings more than 23,000 autopsy cases of annular collection through Japan. MUG QUARTERY 126-133, 1979.
2. Baba,K & Aizawa,S: Nationwide autopsy registration over 30 years. Autopsy in Epidemiology and Medical Research Ed. E Riboli & M Delendi, Lyon International Agency for Research on Cancer pp. 235-244, IARC 1991
3. Urano,Y & Baba,K et al: Annual of pathological autopsy cases in Japan --- Computarization of autopsy data from 1974 to 1979 and their statistical study. Acta Pathol Jpn 19: 23-46, 1982.
4. 馬場謙介: 剖検輯報データベース --- 集計のための作業ファイルについて. 病理と臨床6: 103-110, 1988
5. 馬場謙介: %XRead スモールコンセプト (Extended Read Small Concept (%XR)). Mumps 17: 45-51, 1991 (Japanese with English Abstract).
6. 馬場謙介, 藍沢茂雄, 浦野順文: 日本病理剖検輯報症例の「疾病級・級年次・年齢階級・性別頻度」. 厚生省特定疾患難病の疫学研究班昭和62年度業績報告書. 148-149, 1987.
7. 馬場謙介: 日本病理剖検輯報に掲載された川崎病ならびにその疫学研究に必要な統計母数. 厚生省特定疾患難病の疫学研究班昭和57年度業績報告書. 180-198, 1982.
8. 藍沢茂雄, 浦野順文, 馬場謙介: 剖検輯報によるSLE症例検索. 厚生省特定疾患難病の疫学研究班昭和57年度業績報告書. 176-179, 1982.
9. 藍沢茂雄, 浦野順文, 馬場謙介: 剖検輯報によるSLE症例検索(2). 厚生省特定疾患難病の疫学研究班昭和58年度業績報告書. 271-277, 1983.
10. 浦野順文, 藍沢茂雄, 馬場謙介ら: 剖検からみた老年化傾向と難病. 厚生省特定疾患難病の疫学研究班昭和58年度業績報告書. 251-262, 1983.
11. Baba,K: Indications of increase of occupational pleural mesothelioma in Japan. 5: 3-15, 1983.
12. 馬場謙介, 実藤隼人, 吉村健清: 日本病理剖検輯報のがん疫学への活用. 職業がん---疫学的アプローチ. 篠原出版, 倉恒国徳編 175-190, 1984.
13. 馬場謙介: 脳卒中病型の推移に関する臨床・疫学的研究. 厚生省循環器病委託研究費による研究報告書集(藤島班) 197, 1991.
14. 清水弘之, 馬場謙介ら: 難病疫学における日本病理剖検輯報の疫学的利用について. 厚生省特定疾患の疫学研究班(柳川班)平成3年度研究業績集 132-135, 1992.
15. 馬場謙介: 日本病理剖検輯報の記述情報のデータ化と期待される効果. 厚生省特定疾患の疫学研究班(柳川班) 平成3年度研究業績集. 141-144, 1992.
16. 藍沢茂, 清水弘之ら: 日本剖検輯報を用いた難病悪性腫瘍合併率の検討. 厚生省特定疾患の疫学研(柳川班) 平成3年度研究業績集. 1992.

52 文字検索による日本病理剖検輯報日本語データベースシステム

17. 稲葉裕,馬場謙介ら: 剖検輯報における潜伏胃癌の検討.厚生省特定疾患の疫学研究班(久光班) 平成4年度研究業績集. 1994(印刷中).
18. Baba,K: Enhancement of OCR data of the autopsy annual (*Bouken Shuho*). Mumps 18(Suppl): 134-135, 1991.

異巾文字列処理 Principles to Handle Proportional Character String

馬場謙介
Kensuke Baba

Japanese version of MUMPS provided two string functions to know the width of the full string and character position of the string included within the physical width, but no string functions to get the operated string. So, for example, when we get extracted part of the string, we recognize the position of last character of the field first, and then extracted the string part using the position previously recognized. Even simplest example above, the original string should be written twice as mentioned. This work will show the most smart styles of the combination use of the \$ZPOSITION function and the \$EXTRACT function. In this paper, four external functions: \$\$Extract^%X to return the string part extracted, \$\$Find^%XX to find the string position of left-most character of the i+1th field, \$\$Piece^%XX to return the string part of ith field and \$\$Justify^%XX to return the string part extracted are precisely discussed. Deference between \$\$E^%XX and \$\$J^%XX also discussed. The extrinsic specifications of above \$\$functions are suggestive to be the specifications of the new \$functions of the next Japanese version of M-language.

(Keywords: proportional, string function, \$ZPOSITION, \$ZWIDTH, \$\$E^%XX, \$\$F^%XX, \$\$P^%XX, \$\$J^%XX)

1 はじめに

現(日本語)M言語は、LetterとTypeとから成る文字列に対し、単独で実際に文字列を取得できる文字列処理関数を持たない。(\$ZPOSITION関数は'知る'関数で、'得る'関数ではない)。従って、例えば、物理長で指定した部分を抽出するには、'知る'関数である\$ZPOSITION関数と'得る'関数である\$EXTRACT関数とを組み合わせないと目的が果たせない。しかも、少し複雑になると、\$ZPOSITION関数と\$EXTRACT関数の単純な組み合わせでは対処できなくなる。一般に、Proportional Characterの文字列処理と文字フォントとは不可分の関係にある。従って、使用する文字フォンの字巾辞書を定義しない限り、Proportional Characterの文字列処理は語れない。そこで、今回は、Proportional Characterの内、最も簡単なLETTER字巾:TYPE字巾=2:1の場合に限って考察する。この考察は、LetterとTypeとから成る文字列の文字列処理に適合した単独の(得る)文字列処理関数の仕様に示唆を与える。

国立埼玉病院臨床研究部
〒351-01 埼玉県和光市諏訪2-1
TEL 048-462-1101

Dept. Clinical Research, National Hospital, Saitama,
2-1 Suwa, Wako, Saitama 351-01, Japan
TEL 048-462-1101

2 抽出の問題点といろいろな抽出

抽出範囲と物理位置・長で指定すると、いつもLETTER文字の直前、直後が指定されるとは限らない。指定がLETTER文字の直前後でない(不合理は指定)があり得ることを考えると、少なくとも以下に述べる四つの場合を考慮しておく必要がある。以下その四つの場合の処理を項に分けて論ずる。

範囲を物理位置で指定した抽出：

文字列Sを表示した場合の物理位置aから物理位置bの文字列部分を抽出する一般式を、

$$\text{\$EXTRACT(S,\$ZPOSITION(S,1,a)+.5,\$ZPOSITION(S,1,b)+.5)}$$

としておくと、aやbが、Letterを中途半端に指定しても(不合理指定があっても)、不都合は一応回避できる。しかし、厳密には、aが不合理指定になった場合、そのまゝ表示すると、文字列は指定より半角分左にシフトした出力になってしまっている。また、結果の物理長は、指定の物理長の±1半角巾の間を変動する不都合も生じる。

表示巾を固定した抽出：

文字列Sを表示した場合の物理位置aから物理巾wだけ抽出する一般式は、

$$\text{\$E(\$E(S,\$ZP(S,a)+.5,1023)),1,\$ZP(\$E(S,\$ZP(S,a)+.5,1023)),w)+.5)}$$

で得られるが、この場合でも、結果の物理長は指定の物理長より1半角巾長くなることがある(長くしたくないのなら下線部分を削除)。ここでもシフトの問題が起こる。普通は、表示巾を固定しても、シフトは、問題とならないし、右端の超過もあまり問題にならない。

左端を調整した表示文字列の抽出：

上述の抽出の仕方で普通は問題ないが、表示画面上で複数行にわたり横スクロールしようすると、表示文字列が左シフトしていたのでは上下に不揃いになって不都合である。不揃いを避けるためには、

$$\text{\$S(\$ZP(S,X)\#1=0:"",1:F)_ \$E(S,\$ZP(S,X)+1.5,1023)}$$

とすると、目的の表示文字列を得るのに便利である。ここで、Fは表示巾がTYPE文字と同じ任意の文字(補填文字)である。但し、Xの定義は、(前項のaとは異なり)、Xより左をカットする物理位置とする。

両端を調整した表示文字列の抽出：

上記の方法を発展させて、両端を調整するには、仮想関数を用いる方が便利である。

$$\text{\$BOTH^ \%CUT(String,X,x,F,f)}$$

そして、この仮想関数の内容は、


```

BOTH(S, X, x, F, f) d      q F_S_f
      .s:$zp(S, X)#1=0 F="" s:$zp(S, x)#1=0 f=""
      .s S=$E(S, $ZP(S, X)+1.5, $ZP(S, x))

```

である。ここで、

S: もとの文字列
 X: 左端の切り捨てる位置(この位置の文字は切れ捨てられる)
 x: 右端の位置(この位置の文字は切れ捨てられない)
 F: Xが不合理指定の場合の充填文字(列)
 f: xが不合理指定の場合の充填文字(列)

である。

Table: STRING {\$\$}FUNCTIONS to Know & to Get

Type Only	Type & Letter
FUNCTIONS to Know Characteristic of String	
\$Length	\$ZWidth
\$Find	\$\$F^%XX
	\$ZPosition
FUNCTIONS to Get Treated 'Real' String	
\$Extract	\$\$E^%XX
\$Piece	\$\$P^%XX
\$Justify	\$\$J^%XX

X-Yフィールドへの表示:

ここでは、「文字列Sを一定の長さの行に分割して、X-Yフィールドに表示すること」を考える。前の議論では、「文字の過長の問題」、「シフトの問題」は厳密に議論しなかったが、X-Yフィールドに表示するとなると、この問題を厳密に扱っておかなければならない。何故なら、前行の処理結果が次行に伝播するからである。このような「伝播」をうまく扱うには、(標準言語体系で\$FIND関数が\$PIECE関数の補助関数として定義されているように)、補助関数を定義して、これを使うのが常道の一つである。そして、nで指定した行が終わったつぎの(論理)文字位置を返す次の関数を用意する。

```

$$PACK^%FIND(String,w,n)

```

ここで、wはフィールドの物理横長である。この関数はワープロがそうであるように各行の始まりを左詰めにすることを想定した仮想関数である。この仮想関数の中身は、

```

PACK(STRING,w,n) n j, f s f=1 f j=1:1:n d      q f
      .s f=f+$l($e(STRING,1,$zp(STRING,w))), STRING=$e(STRING,f,1023)

```

である。この仮想関数(外部関数)を使うと、求める n 番目の行の文字列は

$\$E(\text{String}, \$\$PACK^{\wedge}\%FIND(\text{String}, w, n-1, \$\$PACK^{\wedge}\%FIND(\text{String}, w, n)-1))$

となる。更に、上の値を返す仮想関数として

$\$\$PACK^{\wedge}\%PIECE(\text{String}, w, n, \text{Filler})$

を用意しておくとなお更に便利であろう。ここで、Fillerは、右端が中途半端に切れたときの'詰めもの'補填文字で、簡単には、半角のスペースを指定するが、

$\$C(27)_{-}[m>_{-}\$C(27)_{-}[m_{-}$

のように凝ってよい。

(ワープロ風ではなく)、両端を調整した場合のX-Yフィルードへの表示文字列についても同様に補助関数として $\$\$PACK^{\wedge}\%FIND$ を用意して定義するのが賢明である。

Table: $\$\$PACK^{\wedge}\%PIEC(\%, 6, n)$ の返値(例)

%	12345 1 2 x 3 123456A 1 B 2 CD	
n=1	12345	巾=5 字数=5
n=2	1 2 x	巾=5 字数=3
n=3	3 1234	巾=5 字数=5
n=4	56A 1	巾=5 字数=4
n=5	B 2 CD	巾=5 字数=4

3 言語要素を念頭に置いた単純化

外部関数として求められる仕様と言語要素としての関数に求められる仕様とは、同じでよいとは限らない。なぜなら、前者には、'直接の使用の便利さ'が後者に比べてより強く要求され、後者には、'言語として整合性'が前者に比べてより強く要求されるからである。ここでは、言語要素に組み込むべき機能としての意味を強調して、よりスリムにした仕様を持つ外部関数を以下に示す。

```
%X ;eXtended string functions for letter-type-mixed string (c)kb 1993.06.06
E(% , a, b, j) ;EXTRACT
q: '$d(%)' "" q: '$d(a)' "" ;illegal
g: '$d(b) LtBorder' ;Left Border
s: '$d(j) j=0' ;default
g: -j=2 LtBehind ;Left Behind
g: -j=1 LtBorder ;Left Border
q: +j=0 $e(% , $zp(% , a)+1.9, $zp(% , b)) ;Net String
q: +j=1 $s($zp(% , b)#1=0: "", 1: $e(% , $zp(% , b)+.9)) ;Right Border
```

```

q:+j=2 $e(%,$zp(%,$b)+1.9,$l(%))
q "" ;-----
E(%,$a,$b,$j) n t ;EFFICIENT EXTRACT
q:'$d(%)" " q:'$d(a)" "
g:'$d(b) LtBorder
s:'$d(j) j=0
g:-j=2 LtBehind
g:-j=1 LtBorder
s t=$e(%,$zp(%,$a)+1.9,$l(%))
q:+j=0 $e(t,1,$zp(t,$b-a))
q:+j=1 $s($zp(t,$b-a)#1=0:" ",1:$e(t,$zp(t,$b-a)+.9))
q:+j=2 $e(t,$zp(t,$b-a)+1.9,$l(t))
q "" ;-----
LtBehind q $e(%,$l,$zp(%,$a))
LtBorder q $s($zp(%,$a)#1=0:" ",1:$e(%,$zp(%,$a)+.9))
;=====
F(%,$w,$n) n j,$f,$t ;FIND EFFICIENT NEXT FIRST POSITION
q:'$d(%)" " q:'$d(w)" "
s:'$d(n) n=1
s n+=n,$w+=w s f=1,$t=%
f j=1:1:n s t=$e(%,$f,$l(%)) s f=$zp(t,$w)¥1+f
q f
;=====
P(%,$w,$n) ;PIECE OF nth w-WIDTH EFFICIENT FIELD
q:'$d(%)" " q:'$d(w)" "
s:'$d(n) n=1
q $e(%,$$F(%,$w,$n-1),$F(%,$w,$n-1))
;=====

```

ここで、各行ラベルの機能は、

- E jの指定に従い、物理位置の文字または物理範囲の文字列を返す
- EE jの指定に従い、物理位置の文字または物理範囲の文字列を返す
- F 指定した序数の物理範囲の次の論理文字列位置を返す
- P 指定した序数の物理範囲に入る文字列を返す

である。EとEEの相違点は、前者が機械的に切取抽出するのに対し、後者は端に隙間が空いたときその隙間を利用することにある。また、受渡し変数の意味は次の通りである。

- % (処理の対象とする)主文字列
- a 始点の物理位置
- b 終点の物理位置
- w 抽出領域の物理巾
- n 領域の序数
- j 返値の種類(下参照)

jの値によって、返値は次のようになる。

- j=-2 左切断位置より左の文字列(不合理切断文字を含まない)
- j=-1 左切断の不合理文字(不合理でないときはヌル)
- j=0 指定領域の文字列(不合理切断文字を含まない)
- j=1 右切断の不合理文字(不合理でないときはヌル)
- j=2 右切断位置より右の文字列(不合理切断文字を含まない)

\$ZPOSITION関数の第3引数を活かして使うためには、第3引数の値を%ZPに定義して、上記ルーチンの\$ZP関数の第3引数として%ZPを追加することによって、上記ルーチンは、Letter巾>Type巾=2:1でない場合に簡単に拡張できる。

処理の種類の増加に対処するために、jによって処理の種類を指定するようにした。この方針は、「言語要素の数は少なく、夫々の機能は多様に」のM言語の設計の根本精神に叶っている。そして、jの値の-2,-1,0,1,2の意味とその内容との連想性も決して低くない。

4 考 察

M言語は、文字列処理に優れているが、LetterとTypeが混在する文字列に対しては、'知る'関数として\$ZPOSITIONと\$ZWIDTHが用意されているだけで、'得る'関数は、用意されていない。そのため、論じてきたように'知る'関数の他に'得る'関数である\$EXTRACT関数を併用しなければならない。LetterとTypeが混在する文字列の処理は、日本語特有の(正確には物理巾の異なる文字を混用する民族の)問題である。議論を深めて、便利な仕様の'得る'関数を確立したい。ここで、'言語要素としての関数'の仕様を意識して示した機能仕様が、次期の文字列関数の仕様の検討の土台になることを望む。 任意の文字巾を文字から成る文字列のProportional表示/印刷は言語レベルで処理すべきか、アプリケーション・レベルの処理に委ねるかは、議論を要するところである。ここでは、この議論を避けて、LetterとTypeが混在する文字列を考察の対象にした。

文 献

1. Baba, K et al: M言語. オーム社, 東京, in press, 1994.

SQLによるDSMデータベースへのアクセス

Access to DSM database using SQL

辰己岳欣

Takeyoshi Tatsumi

SQL を用いて DSM データベースの検索や更新を行なうためのインターフェースドライバを開発中です。これによって既存の RDB 検索ツールから DSM データベースをアクセスしたり、また、クライアント/サーバアプリケーションの構築が容易に行なえます。

(キーワード: DSM DASL SQL RDB クライアント/サーバ)

We have been developing an interface driver with SQL to retrieve and update DSM database. Using this driver enables us both to access DSM database by way of public retrieval tools for RDB and to easily develop client/server application.

(Keywords: DSM DASL SQL RDB Client/Server)

1 はじめに

弊社では、RDB for OpenVMS やリレーショナルデータベース以外のデータベースに対して、弊社標準の Digital Standard Relational Interface を用いた SQL (Structured Query Language) アクセスを行なうための一連の製品群を計画しています。この計画に基づき、RDBaccess for RMS, RDBaccess for ORACLE 等が現在製品化されています。標準の SQL コマンドによる DSM への検索、更新を可能にする、「DB Gateway for DSM」はこの製品群の一部として位置付けられます。DB Gateway for DSM は、RDB の世界で提供される標準 SQL ツールを DSM データベースに対して使用することを可能にします。

2 DB Gateway for DSM の概要

DB Gateway for DSM は、テーブルのメタ情報を管理する「メタデータドライバ」と実際にデータアクセスを行なう「データドライバ」からなります。DASL (DSM application software library) 検索ドライバを用いてエミュレートされた RDB テーブルから、メタデータドライバが、フィールド長、フィールドタイプなどの情報を取得し、それに基づいてデータドライバが実際にグローバルからデータを RDB レコードとして検索、更新します。

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 06-202-1419

Digital Equipment Corporation Japan Digital Consulting

Nichimen Building, 2-2-2, Nakanoshima, Kita-ku, Osaka 530,

Japan

TEL 06-222-9211 FAX 06-202-1419

必要なデータの要求は、これらのドライバの上位レベルにあるNon SQL Database Serverが行ないます。図1は、データの流れを示しています。

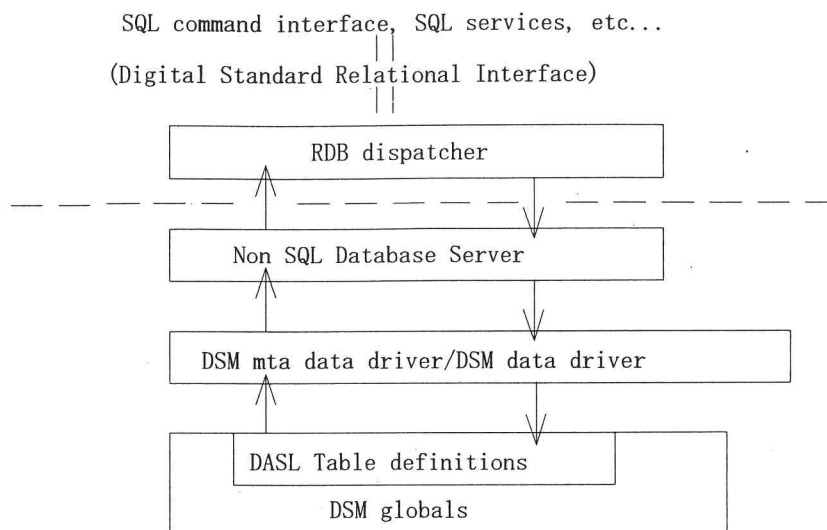


図1 データの流れ

(クライアント)

- Work stations
- Personnel Computers (DECpc, etc...)

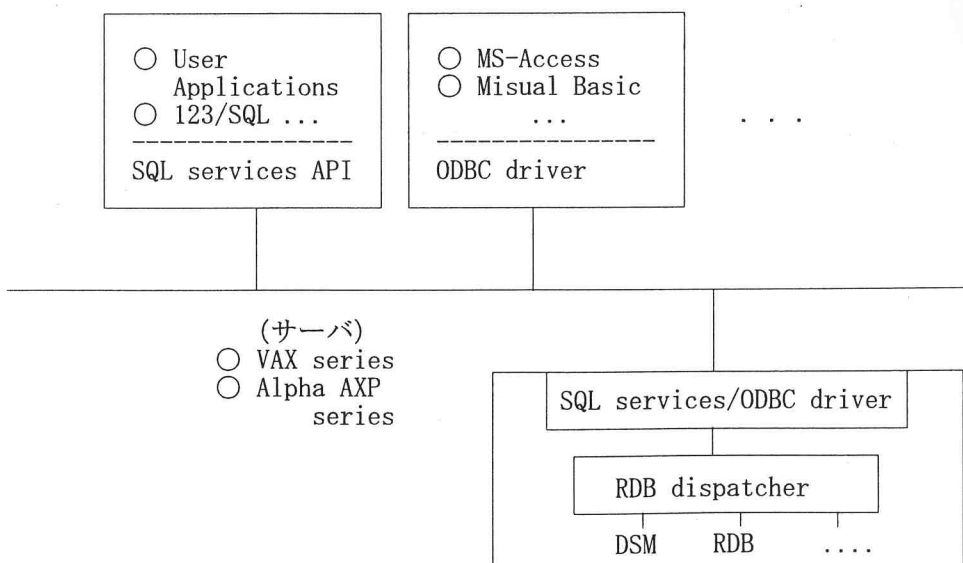


図2 クライアント/サーバモデル

3 クライアント／サーバモデルの構築

Digital Standard Relational Interfaceを用いたものとしては、通常の SQL コマンドインターフェースの他に、クライアント／サーバアプリケーション構築用のAPIを含むSQL servicesがあります。SQL servicesを用いれば、DSMをサーバデータベースとするクライアント／サーバ型のアプリケーションが構築できる他、クライアント側で用意されている既存のSQL services対応のソフトウェアが利用できるようになります。

また、将来的には、米国 Microsoft社がデータベースアクセスの標準インターフェースとして提唱している ODBCドライバを提供する予定ですので、図2に示されるように、ODBCに準拠したMS-Access, Visual Basic等のクライアントツールからDSMデータベースにアクセスできるようになります。

4 まとめ

DB Gateway for DSM を用いれば、エンドユーザは、データベースの種類(RDB,DSM,...)を意識することなくデータ処理が行なえます。また、既存のRDB用の検索ツールや開発ツールが使用できるので、アプリケーション開発の幅が飛躍的に拡大します。

THE UNIVERSITY OF CHICAGO
LIBRARY

1964

最新DSMの特徴と今後のプラン

Some features of the latest DSM and its future plan

佐藤 比呂志
Hiroshi Sato

日本語DSM for OpenVMS バージョン6.2が発表されました。バージョン6.2よりAlpha AXPシステム上でもDSMの稼動が可能となります。最新DSMの特徴と今後のプラン、その中でも特にMWAPI(Window ApplicationInterface)の実装について説明します。

(キーワード: DSM Alpha AXP MWAPI)

We released Japanese DSM for OpenVMS Version 6.2. This version allows us to run DSM on the Alpha AXP system. This article describes some features of the new version , and also our plans on future DSM products especially M Window Application Programming Interface (MWAPI).

(Keywords: DSM Alpha AXP MWAPI)

1 最新DSMの特徴

- ・ 弊社の64ビットRISCベースAlpha AXP システムで稼動する最初のDSM
- ・ 1990年度TYPE A言語拡張仕様の採用
- ・ XMOTIF外部呼び出し関数組み込み
- ・ 新内部漢字コードの採用
- ・ VAX DSMとの相違点とVAXシステムからの移行について
- ・ その他の機能強化点

2 今後のDSMプラン

DSM システムをサーバクライアントモデルに適合させるべく様々な製品が計画されています。

- ・ SQLaccesss for DSM
- ・ MWAPI (M language Window Application Interface)
- ・ 新しいプラットフォーム上でのDSM

日本ディ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 06-202-1419

Digital Equipment Corporation Japan Digital Consulting
Nichimen Building, 2-2-2, Nakanoshima, Kita-ku, Osaka 530,
Japan
TEL 06-222-9211 FAX 06-202-1419

DSM for OSF/1 AXP

DSM for WNT

DSM for Windows 4.0

- ・新しいDSM用ネットワークプロトコルDDP+・RPC（リモートプロシジャコール）機構の組み込み

3 M WAPI の概要

今後、Mでのウィンドウプログラミングを可能とするM WAPI(Window Application Programming interface)について、DSM バージョン6.2 での先行的実装例で説明します。

ssvn(System Structured Variable Names)の使用

Window オブジェクトの階層／属性等を表現する為に`^$WINDOW ssvn`を使用します。ssvn を利用することによりWindow オブジェクトの複雑な階層構造／属性情報を柔軟に管理できます。

このWAPIを利用して書かれたサンプルアプリケーションM言語ソースプログラム、WINDOW構造を定義するグローバルの一部を添付します。

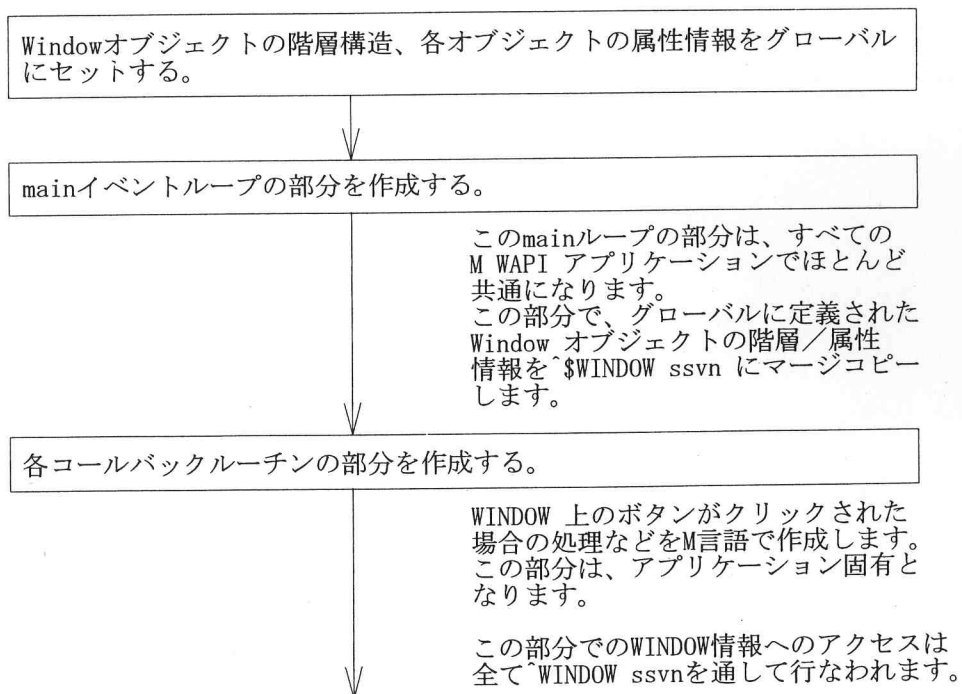


図1 M WAPIでのプログラミング作業の流れ

mainルーチンの部分と一部のコールバックルーチンのリストです。

```

BURGER
BURGER ;PJ; Fast Food Application demo

GO ; main functin
D INIT
M ^$W("W1")=^BURGER("W1") ;load window info
D ^iESTART ;      enter event loop
K ^$W
Q

INIT ; initialize
K
K ^$W
Q

;***** Callback Subroutines *****

CHECK(REASON,AMOUNT) ; checkbox callback
S AMOUNT=$G(AMOUNT)
S GADGET=$P(REASON," ",1),REASON=$P(REASON," ",2)
W !,"You're in CHECK^BURGER; reason=",REASON," , gadget=",GADGET," , amount=",AMOUNT
I AMOUNT D TOTAL(AMOUNT)
Q

DOC(REASON)- ; Document callback
S GADGET=$P(REASON," ",1),REASON=$P(REASON," ",2)
W !,"You're in DOC^BURGER; reason=",REASON," , gadget=",GADGET
Q

LIST(REASON) ; list callback
S GADGET=$P(REASON," ",1),REASON=$P(REASON," ",2)
W !,"You're in LIST^BURGER; reason=",REASON," , gadget=",GADGET
I REASON="SELECT" D
. ;non-standard, to be replaced per spec
. S ITEM=^iWEVENT($J,"CHOICE")
. S TEXT=^$W("W1","G","LIST","CHOICE",ITEM)
. S COST=$P(TEXT," ",1)
. I $D(^$W("W1","G","LIST","VALUE",ITEM)) S SIGN=1
. E S SIGN=-1
. ;end of non-standard code
. S TOTAL=$G(TOTAL)+(COST*SIGN)
. S ^$W("W1","G","TOTAL","VALUE")=$J(TOTAL,0,2)
. S ^$W("W1","G","SB1","VALUE")=TOTAL*100
Q

LISTB(REASON) ; Listbutton callback
S GADGET=$P(REASON," ",1),REASON=$P(REASON," ",2)
W !,"You're in LISTB^BURGER; reason=",REASON," , gadget=",GADGET
Q

LISTE(REASON) ; list entry box callback
S GADGET=$P(REASON," ",1),REASON=$P(REASON," ",2)
W !,"You're in LISTE^BURGER; reason=",REASON," , gadget=",GADGET
Q

```

Window構造を定義するグローバルの一部です。

```

^BURGER
^BURGER("W1") = Test Window Specification
^BURGER("W1","EVENT","CLICK") = WINDOW^BURGER("W1 CLICK")
^BURGER("W1","EVENT","FKEYDOWN") = WINDOW^BURGER("W1 FKEYDOWN")
^BURGER("W1","EVENT","FOCUS") = WINDOW^BURGER("W1 FOCUS")
^BURGER("W1","EVENT","PDOWN") = WINDOW^BURGER("W1 PDOWN")
^BURGER("W1","EVENT","PDRAG") = WINDOW^BURGER("W1 PDRAG")
^BURGER("W1","EVENT","PUP") = WINDOW^BURGER("W1 PUP")
^BURGER("W1","EVENT","UNFOCUS") = WINDOW^BURGER("W1 UNFOCUS")
^BURGER("W1","G","CHECK1","ACTIVE") = 1
^BURGER("W1","G","CHECK1","EVENT","CHANGE") = CHECK^BURGER("CHECK1 CHANGE")
^BURGER("W1","G","CHECK1","EVENT","FOCUS") = CHECK^BURGER("CHECK1 FOCUS")
^BURGER("W1","G","CHECK1","EVENT","HELP") = CHECK^BURGER("CHECK1 HELP")
^BURGER("W1","G","CHECK1","EVENT","SELECT") = CHECK^BURGER("CHECK1 SELECT",.5)
^BURGER("W1","G","CHECK1","EVENT","UNFOCUS") = CHECK^BURGER("CHECK1 UNFOCUS")
^BURGER("W1","G","CHECK1","POS") = 305,135
^BURGER("W1","G","CHECK1","TFFACE") = M, FIXED
^BURGER("W1","G","CHECK1","TFSTYLE") = BOLD
^BURGER("W1","G","CHECK1","TITLE") = Fry .50
^BURGER("W1","G","CHECK1","TYPE") = CHECK
^BURGER("W1","G","CHECK1","VALUE") = 0
^BURGER("W1","G","CHECK1","VISIBLE") = 1
^BURGER("W1","G","CHECK2","EVENT","FOCUS") = CHECK^BURGER("CHECK2 FOCUS")
^BURGER("W1","G","CHECK2","EVENT","HELP") = CHECK^BURGER("CHECK2 HELP")
^BURGER("W1","G","CHECK2","EVENT","SELECT") = CHECK^BURGER("CHECK2 SELECT",1)
^BURGER("W1","G","CHECK2","POS") = 305,170
^BURGER("W1","G","CHECK2","TFFACE") = M, FIXED
^BURGER("W1","G","CHECK2","TFSTYLE") = BOLD
^BURGER("W1","G","CHECK2","TITLE") = Burger 1.00
^BURGER("W1","G","CHECK2","TYPE") = CHECK
^BURGER("W1","G","CHECK3","EVENT","FOCUS") = CHECK^BURGER("CHECK3 FOCUS")
^BURGER("W1","G","CHECK3","EVENT","HELP") = CHECK^BURGER("CHECK3 HELP")
^BURGER("W1","G","CHECK3","EVENT","SELECT") = CHECK^BURGER("CHECK3 SELECT",.5)
^BURGER("W1","G","CHECK3","POS") = 305,205
^BURGER("W1","G","CHECK3","TFFACE") = M, FIXED
^BURGER("W1","G","CHECK3","TFSTYLE") = BOLD
^BURGER("W1","G","CHECK3","TITLE") = Hotdog .50
^BURGER("W1","G","CHECK3","TYPE") = CHECK
^BURGER("W1","G","DOC1","ACTIVE") = 1
^BURGER("W1","G","DOC1","EVENT","FKEYDOWN") = DOC^BURGER("DOC1 FKEYDOWN")
^BURGER("W1","G","DOC1","EVENT","FKEYUP") = DOC^BURGER("DOC1 FKEYUP")
^BURGER("W1","G","DOC1","EVENT","FOCUS") = DOC^BURGER("DOC1 FOCUS")
^BURGER("W1","G","DOC1","EVENT","HELP") = DOC^BURGER("DOC1 HELP")
^BURGER("W1","G","DOC1","EVENT","KEYDOWN") = DOC^BURGER("DOC1 KEYDOWN")
^BURGER("W1","G","DOC1","EVENT","KEYUP") = DOC^BURGER("DOC1 KEYUP")
^BURGER("W1","G","DOC1","EVENT","UNFOCUS") = DOC^BURGER("DOC1 UNFOCUS")
^BURGER("W1","G","DOC1","FRAMED") = 0
^BURGER("W1","G","DOC1","POS") = 300,290
^BURGER("W1","G","DOC1","SIZE") = 14,3,CHAR
^BURGER("W1","G","DOC1","TFFACE") = M,SERIF
^BURGER("W1","G","DOC1","TFSTYLE") = ITALIC
^BURGER("W1","G","DOC1","TITLE") = INSTRUCTIONS doc. title:
^BURGER("W1","G","DOC1","TYPE") = DOCUMENT
^BURGER("W1","G","DOC1","VISIBLE") = 1

```

第20回日本エム・テクノロジー学会大会をふりかえって

第20回日本エム・テクノロジー学会大会 大会長

山本 和子

第20回大会は平成5年9月17日(金)から9月19日(日)にかけて出雲市の市民会館で開催されました。本学会はこれまで主として大阪、東京、名古屋近辺を中心に開催されています。出雲市は初めての試みでもあり、はたしてどれだけご参加下さるか心配しておりましたところ、多数の方々のご参加と絶大なるご支援・ご指導を賜り盛会のうちに無事終了することができ感謝いたしております。

Mumps がM-Technologyと改称されましたので、日本Mumps学会も今回から日本エム・テクノロジー学会と改称されることになりました。これを期に、Mも新しい未来へ向けて飛び立ちたいという願いを込めて、今大会のメインテーマは「飛翔」としました(表1)。

大会プログラムは、プログラム委員会(委員長:木村 一元先生)に作成をお願いしました。表2に掲げておりますように、特別講演2題、シンポジウム3題、一般演題25題、その他実演セッション、ワークショップ、イブニングセッション、海外活動報告、M言語講習会等々非常に盛り沢山で、定められた時間内に盛り込むのに苦労されたことと思います。

その内容について一部ご紹介させていただきますと、まず特別講演第1日目は、Prof. Walters が「国際化されたM言語の姿」と題して、米国で開発された英語版M言語が世界各国で用いられるようになり、各国語版M言語が生まれ、国際化され標準化された経過と今後の展望並びに日本の責任について講演されています。詳細については学会誌「Mumps」に投稿していただきましたのでお読み下さい。最後の言葉「The future is in your hands」が印象的です。

第2日目の特別講演はDumas氏が「ダウンサイジングとLANの利用」と題して、米国テキサス州VA (Veterans Administration)病院情報システムのダウンサイジングの実例を紹介されました。同氏は長年にわたってM言語によるシステム開発に従事され、現在VA病院の経営情報部長をされています。それだけに同氏の話には重みがあり、多くの示唆を得たと思われます。

本学会での大きな話題の一つは、M言語ユーザーの願いの一つであるM言語のJIS化をどのように実現させるかということでしょう。シンポジウムの第1はJIS化にご尽力いただいている大櫛陽一先生を中心に「プログラム言語のJIS化について」でありました。JIS化に向けて作業がかなり進んでいるとのことで、作業担当者の苦労話も飛び出したりして、熱きディスカッションがありました。一日も早くJIS化が実現することを祈ります。

本学会でのもう一つの話は、なんといっても実演セッションで魅力的な顔を出しているマルチウインドウやカラグラフィックなどのGUIではなかったでしょうか。ひと昔前のM言語ユーザーから見れば格段の差があります。シンポジウム第2は「M言語装備の現状と展望」と題して、各メーカーにM言語にどのような機能が追加されつつあるかを話していただきました。GUI関連の機能を充実すべく計画中とのことで、将来が楽しみです。

シンポジウム第3は「M言語教育」でありました。M言語ユーザーを増やす意味においても、教育は大切であると思われます。教育に携わっている現場からの意見が寄せられました。

ワークショップは「ソフトの標準化・・・システムの共有化、プログラムの共同開発は可能か」というテーマでした。私たち大学医学部附属病院情報システム担当者として、少ないソフト開発費と天文学的過大な要望に苦しめられている者にとっては、同じ業務内容のソフトを各大学病院が別々に開発するのではなく、ソフトを標準化して共同利用できないものかと一抹の期待を抱くわけですが、不発に終わりました。なかなか難しいテーマですが、今後こつこつと積み重ねて行きたいと思っています。

一般発表にも、上記にあげた話題が満ち満ちていて、さまざまな創意・工夫がされていました。大いに参考になりましたし、M言語の未来にも期待を大きくした次第です。

表1 第20回エム・テクノロジー学会大会

◆メインテーマ：飛翔——未来のエム・テクノロジー——

- ◇会期： 平成5年9月17日（金）～9月19日（日）
 ◇場所： 出雲市民会館（島根県出雲市塩冶有原町2-15）
 TEL（0853）24-1212
 ◇参加： 1)大会参加 96名（会員64名、一般32名）
 2)M言語講習 初級 5名（会員5名）
 上級 8名（会員3名、一般5名）
 3)展示参加他 11社
-

【プログラム委員会】

- 委員長 木村一元（独協医科大学）
 委員 嶋 芳成（日本ダイナシシステム株式会社）
 委員 上戸 隆（住友電気工業株式会社）
 委員 山下芳範（福井医科大学）
 委員 笹川紀夫（島根医科大学）
-

表2 第20回日本エム・テクノロジー学会大会プログラム

特 別 講 演	A会場 (301)
1. 「国際化されたM言語の姿」 Features for the International M Language 演 者: Prof. Richard F. Walters 司 会: 若井一郎 (マンプスシステム研究所)	【9月18日 (土) 14:00~15:00】
2. 「ダウンサイジングとLANの利用」 演 者: Mr. John R. Dumas 司 会: 鎌江伊三夫 (島根医科大学)	【9月19日 (日) 11:15~12:00】
シンポジウム	A会場 (301)
1. 「プログラム言語のJIS化について」 座 長: 大櫛陽一 (東海大学) : 小松義彦 (日本デジタルイクイップメント)	【9月18日 (土) 15:00~16:00】
2. 「M言語装備の現状と展望」 座 長: 山下芳範 (福井医科大学) : 木村一元 (独協医科大学)	【9月18日 (土) 16:00~17:00】
3. 「M言語教育」 座 長: 河村徹郎 (鈴鹿医療科学技術大学) : 笹川紀夫 (島根医科大学)	【9月19日 (日) 14:00~15:00】
ワークショップ	A会場 (301)
「ソフトの標準化 ... システムの共有化、プログラムの共同開発は可能か」 座 長: 本多正幸 (千葉大学) : 林 恭平 (京都府立医科大学)	【9月18日 (土) 17:00~17:30】
日本MTA総会	A会場 (301)
司 会: 日本MTA会長 河村徹郎 (鈴鹿医療科学技術大学)	【9月18日 (土) 13:00~13:45】
海外活動報告	A会場 (301)
「北米MTA大会報告」 「MDC (M開発委員会) 報告」 座 長: 河村徹郎 (鈴鹿医療科学技術大学) 報告者: 山下芳範 (福井医科大学)	【9月18日 (土) 13:45~14:00】
イブニングセッション	A会場 (301)
「これならわかるオブジェクト指向」 ー 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? ー オーガナイザー: 今泉幸雄 (SASインスティテュートジャパン)	【9月17日 (金) 18:30~20:00】

M言語講習会	A会場 (301)
1. M言語初級講習会	【9月17日 (金) 13:30~16:00】
講師：菊楽純子、伊藤 徹 (住友電気システムズ)	
2. M言語上級講習会	【9月17日 (金) 16:00~18:30】
講師：嶋 芳成、鈴木利明 (日本ダイナシステム)	
実演セッション紹介	A会場 (301)
司 会：今井 敏雄	【9月18日 (土) 9:30~10:00】
(日本デジタルイクイップメント)	
実演セッション	C会場(304,305)
	【9月18日(土)10:00~9月19日(日)15:30】

- ・千葉大学医学部附属病院医療情報部
- ・筑波大学臨床医学系
- ・日本デジタルイクイップメント株式会社
- ・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
- ・システム技研株式会社

一般演題 第1日目 【9月18日(土)】		A会場(301)
◎一般セッションI： テクニカルセッション		【10:00~11:00】
座長：嶋 芳成 (日本ダイナシステム)		
：上戸 隆 (住友電気工業)		
1-1	「パソコンM言語と他言語インターフェイス (C言語、アセンブラ言語による外部モジュールの開発)」	
	○沢田 潔、木下 元一、浅井 廣、岸 真司、新谷 彬	
	名古屋第二赤十字病院医療情報管理部	
1-2	「M言語よりのPC版SASシステムの利用」	
	○田久 浩志、今泉 幸雄*	
	東邦大学医学部病院管理研究室	
	*SASインスティテュートジャパン	
1-3	「MS-Windows によるインテリジェント端末の試み」 (実演)	
	岡田 好一 筑波大学臨床医学系	
1-4	「当検査センターにおけるM言語を用いた検査サブシステムの開発経過」	
	岩城 健一 (株)フジモトバイオメディカルラボラトリーズ	
1-5	「汎用機とM言語システムの接続」	
	阿部 敏明 (株)大阪血清微生物研究所システム部	

◎一般セッションⅡ： 画面制御

【11:00～12:00】

座長：岡田好一（筑波大学）

：松本重雄（システム技研）

2-1 「MS-WindowsとM言語による電子カルテシステムの開発」

○嶋 芳成、小嶋 興二*、谷掛 駿介**、喜多野三夫***、
岡田 怜****、笹原 茂****

日本ダイナシステム株式会社

*小嶋診療所

**谷掛整形外科診療所

***喜多野西大寺診療所

****株式会社リコー

2-2 「MWAPI (M言語WINDOWS-API)による電子カルテの試み」

○山下 芳範、山本 和子*、須藤 正克

福井医科大学医学情報センター

*島根医科大学医療情報学講座

2-3 「MS-Windowsにおける電子教科書の試作」

○永田 守秀、山下 芳範*、岡田 好一**、高橋 隆

京都大学医学部附属病院医療情報部

*福井医科大学医学情報センター

**筑波大学臨床医学系

2-4 「マルチウインドウ・カラーグラフィックス環境での会話型画面設計ツール
Screen maker」 (実演)

里村 洋一、山崎 俊司、○鈴木 隆弘

千葉大学医学部附属病院医療情報部

一般演題 第2日目 【9月19日(日)】 A会場(301)

◎一般セッションⅢ： 医学におけるMUMPS I

【9:00～9:48】

座長：田久浩志（東邦大学）

：沢田 潔（名古屋第二赤十字病院）

3-1 「汎用シェルと医療向けシェルの知識表現と推論機構」

○今泉 幸雄、大嶋 陽一*

SASインスティテュートジャパン

*東海大学医学部病院管理学教室

3-2 「自然語情報をデータベースに取込むための構造言語学の応用」

○マルシオ・ビチキ・ド・アマラル、里村 洋一

千葉大学医学部附属病院医療情報部

3-3 「M言語とMedical Decision Making」

○鎌江 伊三夫、笹川 紀夫、林 振健、山本 和子

島根医科大学医療情報学講座

3-4 「個人病院に於ける病院情報システムの活用」

ーレセプト専用機からMUMPSマシンに変更してー

○土屋 喬義、土屋 恭子、木村 一元*

土屋小児病院、獨協医科大学第一小児科

*獨協医科大学総研ME室

◎一般セッションIV: 医学におけるMUMPS 2

【9:48~10:36】

座長: 小森 優 (京都大学)

: 今泉幸雄 (SASインスティテュートジャパン)

4-1 「千葉大学病院医療情報システム(system-CHIBA)におけるシステム運用」

○本多正幸、里村洋一、鈴木隆弘、山崎俊司、高林克日己、赤井ユキコ*

、石井 晃**、中村裕義**、野口 昇**、北田光一**、清水 宗***

千葉大学医学部附属病院医療情報部

*千葉大学医学部附属病院看護部

**千葉大学医学部附属病院薬剤部

***住友電工システムズ株式会社

4-2 「中央検査部トータルシステムの再構築」

○中尾 満、森田 寛二、藤井 厚男、東畠 正満、翼 典之、

花坂 志郎*、平山 清和*

大阪市立大学病院中央検査部

*住友電工システムズ株式会社

4-3 「未熟児部および産科データベースの構築とその結合」

○田中 吾朗、渡辺 博*、木村 一元**

獨協医科大学第2小児科

*獨協医科大学産婦人科

**獨協医科大学総研ME室

4-4 「日本病理剖検輯報の日本語データベース」

◎一般セッションV: 医学におけるMUMPS 3

【10:36~11:12】

座長: 中尾 満 (大阪市立大学病院)

: 永田守秀 (京都大学病院)

5-1 「健診システムにおける病名と家族歴の登録サブシステムの開発」

○大櫛 陽一、笹川 紀夫*、高橋 正宏**、原 寿夫***

東海大学医学部医学情報部

*島根医科大学医療情報学講座

**郡山市健康振興財団

***郡山市医師会

5-2 「選択必修科目履修申告システムの開発」

○斉藤 成広、松木 三徳、菊地 良平、大櫛 陽一*

東海大学伊勢原学務課

* 東海大学医学部医学情報部

5-3 「自然語病名の使用頻度の分布について」

○林 振健、鎌江伊三夫、山本和子、笹川紀夫、山下芳範*、須藤正克*

島根医科大学医療情報学講座

* 福井医科大学医学情報センター

◎一般セッションⅥ: MUMPSの新技術

【13:00~14:00】

座長: 馬場謙介 (国立埼玉病院)

: 藤江 昭 (住友電工システムズ)

6-1 「SQLによるDSMデータベースへのアクセス」

辰己 岳欣 日本デジタライゼーション西日本第一統合システム部

6-2 「U-MUMPSクロスシステムジャーナルによるミラーシステムの構築」

○吉村 貴由、木村 隆、伊藤 徹、吉中位知郎、藤江 昭

住友電工システムズ株式会社応用システム事業部

6-3 「MとTPモニターとの統合の試み」

○今井 敏雄、佐藤 真美

日本デジタライゼーション西日本第一統合システム部

6-4 「最新DSMの特徴と今後のプラン」(実演)

佐藤 比呂志 日本デジタライゼーション西日本第一統合システム部

6-5 「逆\$ORDER関数の効用ー本院の入院診療録管理システムへの応用を例に」

○馬場 謙介、石名田洋一、鈴木 孝始*

国立埼玉病院臨床研究部

* 国立埼玉病院医事課

「日本エム・テクノロジー学会」ご入会のご案内

日本エム・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化推進
- 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」ことになります。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：京都1-57513
 口入者名：日本エム・テクノロジー学会

イ. 銀行口座 口座：三和銀行 聖護院(しょうごいん) 支店
 口座番号：普通口座 445-170772
 名義人：日本エム・テクノロジー学会
 高橋 隆 (たかはし たかし)

「日本エム・テクノロジー学会」規約

第一章 総 則

第1条 本会は日本エム・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、庶務財務担当幹事の属する住所に置く。

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) 学術集会、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) 日本エム・テクノロジー学会大会長(以下「大会長」という)	1名
3) 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
JIS・ISO担当	1名
4) 会計監事	1名
5) 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。

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、年次日本エム・テクノロジー学会大会を総括する。
- 3) 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。
- 4) 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。
- 5) 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。
- 6) 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。
- 7) 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。
- 8) 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。
- 9) M言語標準化担当幹事は、M言語の標準化を図る。
- 10) 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。
- 11) 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。
- 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、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委員会を召集する。
- 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 (日本エム・テクノロジー学会大会)

- 1) 本会は年1回以上の日本エム・テクノロジー学会大会を開催する。

第六章 資産および会計

第20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21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22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23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24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25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26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27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28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29条 本会の略称を日本エム・テクノロジー学会(MTA-Japan)という。

第30条 本会の入会費、年会費は別に定めるものとする。

第31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32条

- 1) 本規約は1997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。

「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。

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

〒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内

雑誌Mumps編集委員会宛

Tel:0853-23-2111（内線2941）FAX:0853-25-2764 NIFTY-serveID:HAH00247（山本）

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。別冊の料金は別に定めます。別冊の部数は校正稿提出時に申し出いただければ、50部単位で増刷いたします。

大櫛 陽一（東海大学医学部）

大谷 元彦（藤田学園保健衛生大学医学部）

木村 一元（独協医科大学医学部）

河村 徹郎（鈴鹿医療科学技術大学医用
工学部）

里村 洋一（千葉大学医学部）

高橋 隆（京都大学医学部）

林 恭平（京都府立医科大学医学部）

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

〒693 島根県出雲市塩冶町89-1

島根医科大学医療情報学講座内

日本MUMPS学会 雑誌編集委員会宛

Tel: 0853-23-2111 (内線2941)

FAX: 0853-25-2764

NIFTY-serve ID: HAH00247 (山本)

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19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。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19巻

1994年7月31日発行

発行者 日本エム・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工工学部医用情報工学科
 Tel:0593-83-9666
 FAX:0593-83-9666

編集者 日本エム・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)

事務局 日本エム・テクノロジー学会事務局
 庶務財務担当幹事 高橋 隆
 〒606-01 京都市左京区聖護院川原町54
 京大病院 医療情報部内
 Tel 075-751-3211
 Fax 075-751-3076

印刷所 宏和印刷株式会社
 〒544 大阪市生野区生野西1丁目1番30号
 Tel:06-716-2975 FAX:06-716-5286

DSMはいま . . .

1995年1月3日、DSMについて米国InterSystems社と米国ディジタル イクイップメント社(DEC)は技術提携および販売提携いたしました。

この発表に伴い、DECはDSMの所有権をInterSystems社へ譲渡し、DSM技術者／管理者が InterSystems社へ移籍し、開発が継続して進められています。

また、今後DECは InterSystems社からDSMの供給を受け続けVAR(Value Added Reseller)として販売、サポートし続けます。

日本語に強い日本語DSMの供給

国内では、従来どおり日本語DSMは製品／製品サポート共に日本DECが独占供給いたします。

ダウンサイジング化とオープン化への対応

DECは、今後ハードウェアは Alpha AXP システムを中核に置き、ソフトウェアはDSMをはじめとするソフトウェアベンダー供給製品を最も効率よく融合させていくSIビジネスの強化に力点を傾けていきます。

DECのオープン・クライアント／サーバ・システムとSI技術、そして InterSystems 社のDSM製品、GUI関連製品の組み合わせにより、まさに**M言語**が得意とする情報システムのダウンサイジング化の実現に向けて邁進していきます。

両者の技術を融合し、本格的なダウンサイジング化とオープン化への対応を目指す、DSM SIベンダーにご期待ください。

日本ディジタル イクイップメント株式会社

日本語DSM、最新のM言語技術に関するご質問は、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。

東京：03-5992-7054

大阪：06-222-9211

※DSMは、米国InterSystems社の商標です。

日本DEC

日本 ディジタル イクイップメント株式会社
〒167 東京都杉並区上荻1-2-1 ☎ 03(5349)7111

CONTENTS

■ Editorial

We can save computing environment in the 21st century.....Yoshikazu Okada 1

■ Original Articles

Internationalization of M: Progress and Responsibilities.....Richard F. Walters 3

Reconstruct of Total Laboratory System..... 15

Mitsuru Nakao, Kanji Morita, Atsuo Fujii, Masamitsu Higashihata, Shiro Hanasaka,
Kiyokazu Hirayama

Knowledge representation and inference engine in general expert shell and medical expert
shell.....Yukio Imaizumi, Yoichi Ogushi 23

M Language and PC-SAS.....Hiroshi Takyu, Yukio Imaizumi 31

An Approach to Integrating M with a TP monitor.....Toshio Imai, Mami Sato 37

Information Retrieval system for the *Bouken Shuho* by Japanese Word Keys..... 43

Kensuke Baba, Soken Koh, Shigeo Aizawa

Principles to Handle Proportional Character String.....Kensuke Baba 53

■ Articles

Access to DSM database using SQL.....Takeyoshi Tatsumi 59

Some features of the latest DSM and its future plan.....Hiroshi Sato 63

■ Report: 20th MUMPS Conference

On 20th MUMPS Conference.....Kazuko Yamamoto 67

Information for MUG..... 75

Instruction for Authors..... 83

Editor's Postscripts..... 87