

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

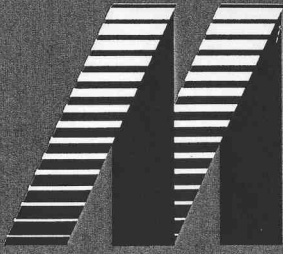
東海大学医学部・基礎医学系

大櫛陽一

Tel: 0463-93-1121 ext. 2140

Fax: 0463-96-4301

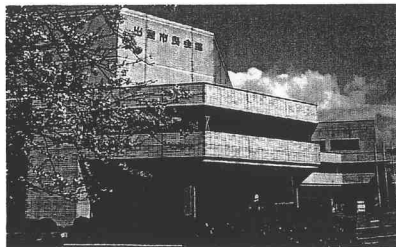
Email: youichi@keyaki.cc.u-tokai.ac.jp



*Technology
Association
Japan*

Proceedings '99

M Technology Association of Japan



The 26 th Conference
August 26-29, 1999
Izumo, Shimane

K
147
K
K



第26回日本Mテクノロジー学会大会

論文集

1999年8月26日～8月29日

島根県出雲市



目次

● 巻頭言	M-1
● 大会概要	M-3
● 第 26 回日本 M テクノロジー学会大会プログラム	M-7
● 論文集	
招待講演 医療データ共有をいかに実現するか——交換規約開発の現状——	招-1
会長講演 病院情報システム——これまでとこれから——	講-1
1) MSM Server/Workstation による健診システムの開発	1-1
2) 病診連携ベッド空床数照会システムの開発経験	2-1
3) 遠隔診断支援システムの構築	3-1
4) Cache'-Web 連携機能を用いた「地域医療情報ネットワークシステム」の開発について	4-1
5) M 言語の西暦 2000 年問題	5-1
6) M Web-Link による授業出席管理システムの開発	6-1
7) 大域変数の階層構造と日本語文切断のアルゴリズム	7-1
8) 診療データベースからのデータマイニング	8-1
9) CORBAMED の活動と応用例	9-1
10) sumiACCEL/21 (仮称) における電子カルテ機能について	10-1
11) 標準化を目指す新患者検査データベースの設計と再構築	11-1
12) 電子カルテ EChart	12-1
13) 国際標準化の動向と電子カルテ	13-1
14) MML と MERIT-9 の動向と応用	
15) 電子保存通達と電子カルテ	15-1
16) C 言語を用いた Cache' から DLL サブルーチンを呼び出す M 関数の作成	16-1
17) 開発支援ツール VB コード・ジェネレータとドキュメント・ジェネレータの紹介	17-1
18) M 言語用のコンパイラ・コンパイラの作成	18-1
19) JAVA による M 言語の装備	19-1
20) Java の最新動向	20-1
21) Cache' の最新動向	



情報の共有と標準化

第26回日本Mテクノロジー学会大会

大会長 山本 和子

1975年5月30日に、京都市で第1回MUMPSグループ研究会が開催されてから、はや24年が過ぎました。処理速度や開発効率を誇った時代、MUMPSがMテクノロジーに名称変更された時代、M言語の標準化にエネルギーを注ぎ込んだ時代、そしてMが Cashe に統一された時代を迎えました。その間にコンピュータのダウンサイジングが進行し、様々なGUI (Graphical User Interface) と自動プログラミング技術が開発され、通信技術の発達によりネットワークの時代へと進んでおります。

1983年頃の病院情報システムは、医事システムのコンピュータのメモリーが1MB、ハードディスクが200MB x 2台の時代でした。それが現在ではパソコンでさえメモリー256MB、ハードディスク6GBを装備できる時代になりました。処理速度を云々する時代ではなくなったのかもしれませんが。そしてインターネットは想像もできなかった速さで普及しました。JavaやCORBAも普及しています。Mはどのように対応するのでしょうか。また、ネットワークで情報を共有するためには、情報の標準化が重要な役割を果たします。

そこで、第26回日本Mテクノロジー学会大会では「情報の共有と標準化」をメインテーマにしました。今回、招待講演として、医療情報学会の電子カルテ研究会で活躍され、診療情報交換規約を纏められた宮崎医科大学の吉原博幸教授をお招きできたことは大変うれしく思います。吉原先生の主たる研究テーマとともに、米国での事情など、よもやま話も聞けるのではと期待しています。

それから、プログラム委員長のご尽力で、医療情報学会の里村会長から国際標準化の最新動向と電子カルテのご講演や、M以外のJavaやCORBAなどの最新技術動向のお話を聞けることになりました。Mは医療を中心に発展したようにも見受けられますが、実際には医療以外にも利用されています。医療も、医療中心から保健、福祉、介護へと発展しております。一般演題では、Mを用いた先端的技術開発の演題をご応募いただきました。活発に議論していただき、先達の豊富な経験を吸収し、Mの明日への発展に寄与していただけることを期待しております。

大会概要

メインテーマ：情報の共有と標準化

大会会期：1999年8月26日（木）9時30分～17時30分

1999年8月27日（金）9時00分～16時30分

チュートリアル：1999年8月28日（土）9時30分～17時

Cache 概要、Cache プログラミング、Cache Object、Visual Basic の利用

1999年8月29日（日）9時30分～15時

その他の GUI ツールの利用（Access, Delphi, Java 等）、Cache WebLink

会場：島根県出雲市 出雲市民会館 三階301会議室（JR出雲市駅から徒歩20分）

島根医科大学第三研究棟四階電算演習室

招待講演：1999年8月26日（木）13時50分～14時40分

広域診療情報交換への取り組み

——ボストンで感じた日米の温度差——

宮崎医科大学附属病院医療情報部長 吉原 博幸 教授

懇親会：1999年8月26日（木）17時30分～

シェルブール二階（大会会場から徒歩1分）

参加についてのご案内

会費：大会参加費 5000円

懇親会費 6000円

チュートリアルのみ参加 3000円（含資料代）

大会参加者でチュートリアル参加者は資料代1000円

・参加の受付は当日のみです。事前登録は行いませんのでご了承ください。

・チュートリアルと病院見学参加者は準備の都合上、なるべく事前に事務局までお知らせ下さい。

受付：8月26日 9時～17時

8月27日 9時～15時

日本Mテクノロジー学会関連行事

日時：1999年8月25日（水）15時～17時

島根医科大学附属病院医療情報ネットワークシステム見学会

集合場所：島根医科大学第三研究棟四階電算演習室

（希望者は準備の都合がありますので8月15日までに事務局へお申し込み下さい）

日時：1999年8月25日（水）17時～20時

日本Mテクノロジー学会幹事・評議員会

島根医科大学第三研究棟三階医療情報学講座内図書集会室

日時：1999年8月26日（木）13時10分～13時40分

日本Mテクノロジー学会総会 出雲市民会館 三階会議室

組 織：

第26回日本Mテクノロジー学会大会

■大会長 山本和子 (島根医科大学医療情報学講座)

■プログラム委員会

プログラム委員長 本多正幸 (千葉大学医学部附属病院医療情報部)

プログラム委員 今泉幸雄 (ファイザー製薬(株)IT統括部)

岡田好一 (東海大学医学部医用工学情報系)

木村一元 (獨協医科大学医学情報センター)

嶋 芳成 (日本ダイナシステム(株))

田久浩志 (東邦大学医学部病院管理学教室)

■実行委員会

実行委員長 柳樂真佐実 (島根医科大学医療情報学講座)

実行委員 石垣恭子 (佐賀医科大学看護学科)

小倉常睦 (住友電工システムズ(株)医療情報システム事業部)

沢田 潔 (名古屋第二赤十字病院医療情報部)

向井まさみ (ニチメンデータシステム(株)技術部)

山下芳範 (福井医科大学医学情報センター)

日本Mテクノロジー学会関係ホームページ

URL : <http://www.shimane-med.ac.jp/japanese/medical/meeting/26mtaj/>

URL : <http://ww1.fukui-med.ac.jp/mta/>

URL : <http://www.dokkyomed.ac.jp/j-sj/mta/26mtaj/>

大会事務局： 〒693-8501

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

第26回日本Mテクノロジー学会大会事務局

FAX : 0853-20-2170

TEL : 0853-23-2111 内線5380

担 当 : 石丸裕子、加藤由美

E-mail : mta@shimane-med.ac.jp

会場までのアクセス

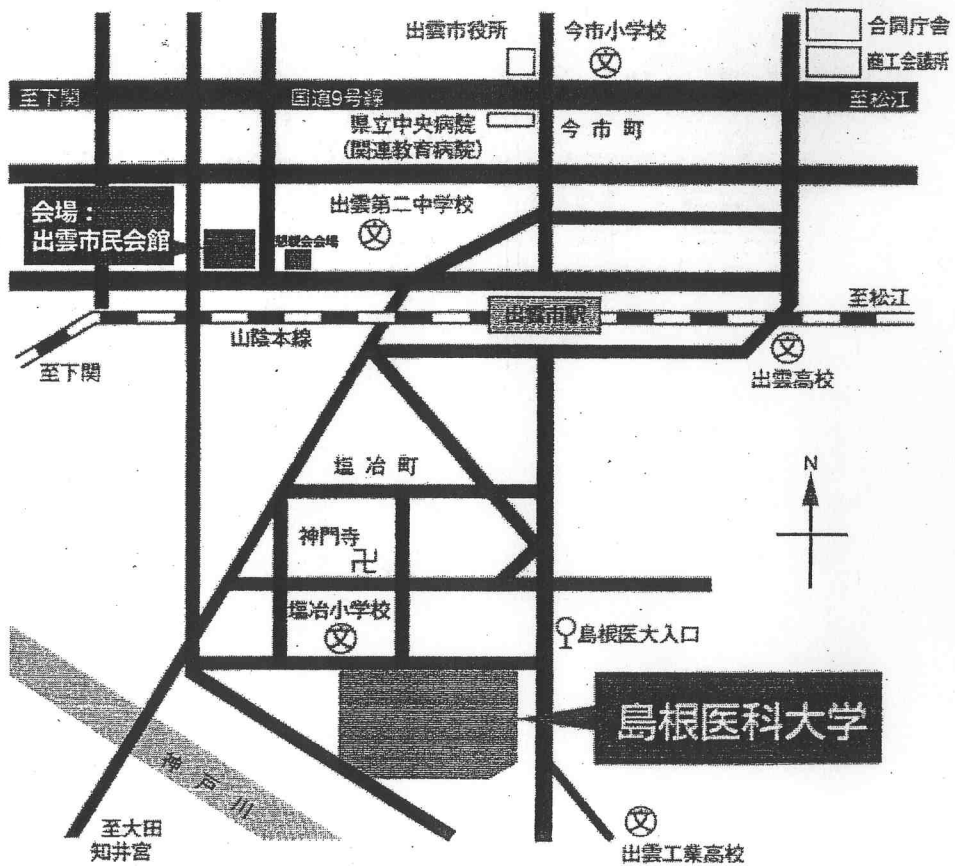
【出雲市民会館】

◇JR 出雲市駅(北口)から

- ・タクシーで3分
- ・徒歩20分
- ・バスは駅隣接のバスターミナルから「島根医大」または「出雲大社」「日御碕」行きに乗車し、最寄りのバス停下車。

◇出雲空港から

空港連絡バスで出雲市駅まで25分。

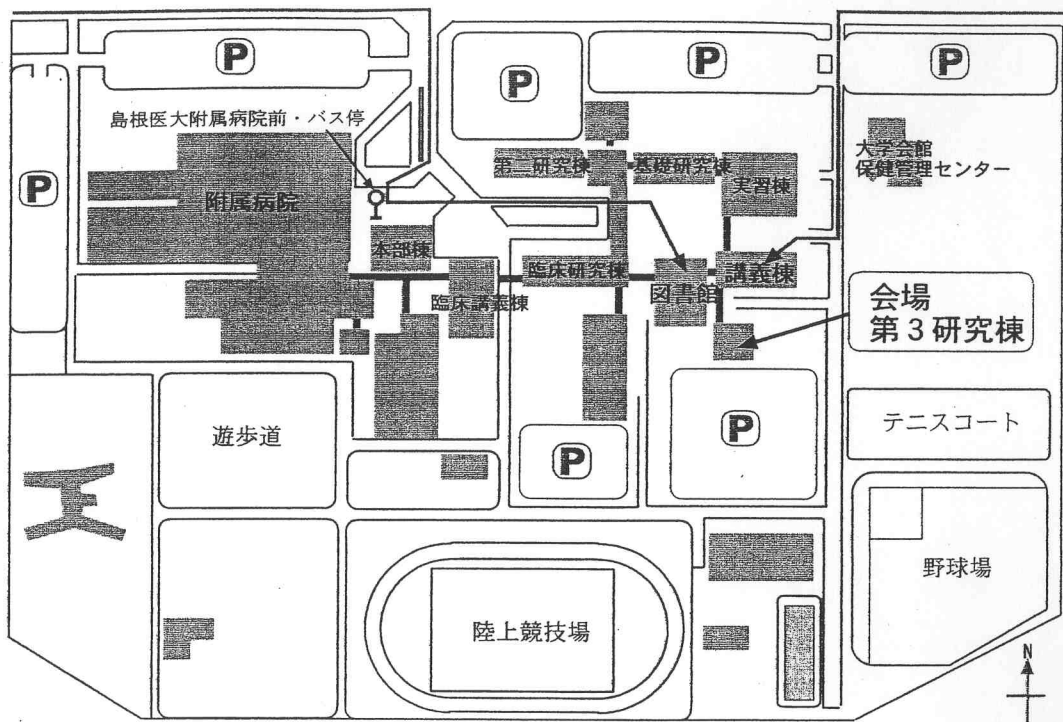


【島根医科大学】

◇JR 出雲市駅(南口)から

- ・タクシーで3分
- ・徒歩20分
- ・バスは駅隣接のバスターミナルから「島根医大」行き乗車、島根医大病院前下車。または「根波」、「出雲須佐」行きに乗車、島根医大入り口で下車、徒歩5分。

島根医科大学構内図



第26回日本Mテクノロジー学会大会プログラム

8月26日(木) 第1日目 出雲市民会館 三階 301会議室

開会の辞 9:30~9:40

山本和子(島根医科大学)

一般セッション I (医療情報システム I)

9:40~11:20

座長: 津本周作(島根医科大学)

- 1 MSM Server/Workstation による健診システムの開発
大櫛陽一(東海大学医学部医用工学情報学)
- 2 病診連携ベッド空床数照会システムの開発経験
沢田 潔 川田新一 木下元一 浅井 広 岸 真司
(名古屋第二赤十字病院医療情報部)
- 3 遠隔診断支援システムの構築
本多正幸 鈴木隆弘 高林克日己 津本優子 里村洋一 植松祐美子* 姜 琳**
(千葉大学医学部附属病院医療情報部 同看護部* 住友電工システムズ(株)**)
- 4 Cache'-Web 連携機能を用いた「地域医療情報ネットワークシステム」の開発について
向井まさみ* 山本和子** 中島裕生*
(ニチメンデータシステム株式会社技術部* 島根医科大学医療情報学**)

大会長講演

11:30~12:10

座長: 河村徹郎(鈴鹿医療科学大学)

病院情報システム——これまでとこれから——

山本和子(島根医科大学医療情報学)

<昼 食>

日本 M テクノロジー学会総会 13:10~13:40

日本 M テクノロジー学会会長 大櫛 陽一

招待講演

13:50~14:40

座長: 本多正幸(千葉大学)

医療データ共有をいかに実現するか——交換規約開発の現状——

吉原博幸(宮崎医科大学附属病院医療情報部)

一般セッション II (M テクとその応用)

14:50~16:05

座長: 岡田好一(東海大学)

- 5 M 言語の西暦 2000 年問題

鈴木利明 嶋 芳成(日本ダイナシステム株式会社)

- 6 M Web-Link による授業出席管理システムの開発
大櫛陽一 岡田好一 春木康男(東海大学医学部医用工学情報系)
- 7 大域変数の階層構造と日本語文切断のアルゴリズム
高橋 亘(関西福祉科学大学社会福祉学部)

企画セッション I (最新技術動向 I)

16:15~17:05

座長：山下芳範(福井医科大学)

- 8 診療データベースからのデータマイニング
津本周作 高林克日己* 山崎俊司**
(島根医科大学医療情報学 千葉大学医学部第二内科* 琉球大学医学部附属病院
医療情報部**)
- 9 CORBamed の活動と応用例
藤生尚光((株)ニコン SI 開発部)

8月27日(金) 第2日目 出雲市民会館 三階 301 会議室

一般セッション III (医療情報システム II)

9:00~10:15

座長：沢田 潔(名古屋第二赤十字病院)

- 10 sumiACCEL/21(仮称)における電子カルテ機能について
村上 英 木戸須美子 青木浩二 上戸 隆 藤江 昭(住友電工システムズ(株))
- 11 標準化を目指す新患者検査データベースの設計と再構築
姜 琳 本多正幸* 鈴木隆弘* 高林克日己* 岩橋俊樹 四方昭弘 里村洋一*
(住友電工システムズ(株) 千葉大学医学部附属病院医療情報部*)
- 12 電子カルテ EChart
嶋 芳成 山本 樹(日本ダイナシステム株式会社)

特別セッション(情報の共有と標準化)

10:25~12:00

座長：大櫛陽一(東海大学)

- 13 国際標準化の動向と電子カルテ
里村洋一(千葉大学医学部附属病院医療情報部)
- 14 MML と MERIT-9 の動向と応用
山下芳範(福井医科大学医学情報センター)
- 15 電子保存通達と電子カルテ
山本和子(島根医科大学医療情報学)

<昼食>

一般セッション IV (開発支援・M 利用技術)

13:00~14:40

座長：里村洋一(千葉大学)

- 16 C 言語を用いた Cache'から DLL サブルーチンを呼び出す M 関数の作成
岡田好一 大櫛陽一 大久保裕和*(東海大学医学部、株式会社ケンウッド*)

- 17 開発支援ツール VB コード・ジェネレータとドキュメント・ジェネレータの紹介
 広瀬清司(有限会社エム ビイ ビイ)
- 18 M 言語用のコンパイラ・コンパイラの作成
 鈴木利明 嶋 芳成(日本ダイナシステム株式会社)
- 19 JAVA による M 言語の装備
 内田達弘(名城大学理工学部)

企画セッションⅡ(最新技術動向Ⅱ)

14:50~16:10

座長: 木村一元(獨協医科大学)

- 20 Java の最新動向
 今泉幸雄(ファイザー製薬株式会社 IT 統括部)
- 21 Cache' の最新動向
 Joe Gallant(Objects Product Manager, InterSystems Corp.)

閉会の辞 16:10

山本和子(島根医科大学)

8月28日(土) 第3日目 島根医科大学 第三研究棟四階電算演習室

チュートリアル 1

9:30~12:00

- 1 Cache' の概要
- 2 Cache' プログラミング

チュートリアル 2

13:00~17:00

- 3 Cache' Objects
- 4 Visual Basic の利用

8月29日(日) 第4日目 島根医科大学 第三研究棟四階電算演習室

チュートリアル 3

9:30~12:00

- 5 その他の GUI ツールの利用(Access, Delphi, Java 等)

チュートリアル 4

13:00~15:00

- 6 Cache' WebLink



第26回日本Mテクノロジー学会

論 文 集

医療データ共有をいかに実現するか - 交換規約開発の現状 -

宮崎医科大学附属病院医療情報部

吉原博幸

抄録

1999年4月22日に出された厚生省通達により、一定の条件を満たせば診療録の電子保管が可能になった。これに伴って、今後電子カルテの普及が期待される。電子カルテが真の意味で社会に役立つためには、複数の医療機関にまたがった診療情報交換が可能になる必要がある。宮崎県では、その第一歩として、インターネット上に仮想LAN(VPN)を作り、その中に共同利用の診療情報データベースを構築中で、これを医療機関、患者、行政等が共有する予定である。同様の試みが各地で行われることを期待している。

キーワード：電子カルテ, MML, MERIT-9, XML, Name Spaces, SGML, HTML VPN

1 電子カルテが機能するための3つの問題

患者さんは、一生の内に少なくとも5から10の医療機関を受診する。その大部分は同じ医療圏で占められると考えられる。電子カルテが現実のものとして語られるようになった現在、少なくとも県域内での医療情報の流通は、医療機関にとっても患者さんにとっても意義のあることと考えられる。しかし、医療機関が独自にLANを構築し、電子カルテに貯えられたデータを安全に管理することは、経済的にも、技術的にも大変困難であると思われる。電子カルテの情報が広域で利用され、その価値を十二分に発揮するためには、以下の3つの問題をクリアする必要がある。

- 1) 異なる電子カルテシステム間で情報交換が可能であること
- 2) 患者さんのデータの所在を明らかに出来る仕組みが存在すること
- 3) 医療機関が簡単に使うことの出来るシステムが提供されること

以下、これらの3つの問題に対するソリューションについて述べる。

1.1 異なる電子カルテシステム間での情報交換の仕組み：MML (Medical Markup Language) [1-3]

データベースは、一種の表(テーブル)と考えられ、個々の施設では、独自の定義のテーブルを運用している。従って、たとえば、住所、氏名、病

名、、、の様な簡単なデータセットを交換する場合を考えても、施設間でデータの出現順序が違う可能性がある。これをそのまま転送すると、当然データの順序が入れ変わってしまう。それでは都合が悪いので、全てのシステムのデータベースを同じ定義にする方法が考えられるが、これは事実上不可能である。すでに全国で動いているシステムを全て書き直す必要があるからだ。そこで我々が提唱している方法がデータ交換のための標準手続き(MML: Medical Markup Language)である。この考え方は、個々のシステムを拘束せず、図1に示したように、他の施設とのデータ交換の際、MMLフォーマットに変換して送出する。受け取った側は、MMLフォーマット化されたデータを自分のシステムに合った形式(順序)に変換してデータベースに格納する。こうすることによって、施設毎の独自性を保ちながら、全国の医療機関とデータの交換が可能

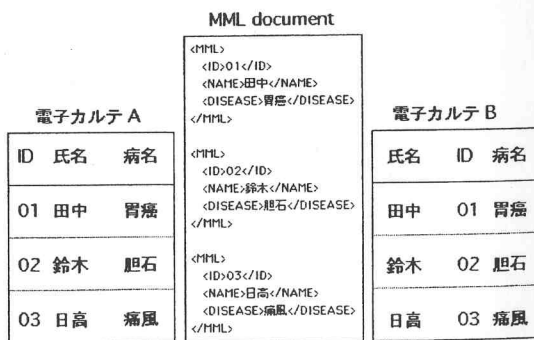


図1 MMLによる医療情報交換のしくみ

能になる。各々のシステムはベンダーが競争して作ることになるので競争の原理が働き、より使いやすいシステムが生まれる。

この方式のもう一つのメリットは、各システムが、標準フォーマットに対するインターフェイスを一つだけ作れば良いという点である。もし、個々のシステム間で交換のためのインターフェイスを作ることになると、膨大な数のインターフェイスを用意しなければならなくなり、これは事実上不可能なことである。

MMLは、WWW記述言語であるHTMLと同様の記述方法を用いている。一般的に、この様な記述方法は「タグ付言語」「マークアップ言語」と言われる。HTMLのタグは、<center>、<H1>、<p>（各々、文字をセンタリング、ヘッダ1、パラグラフ、の意味）などの様に、主として文字をレイアウトするための書式情報で成り立っている。医療情報をこのHTMLで送ることは簡単であるが、あくまでも「人間が見て解釈する」ことを前提とする。つまり、

```
<left>鈴木一郎</left><p>
<left>胃癌</left><p>
```

というHTML化された情報が相手に送られた場合、

```
鈴木一郎
胃癌
```

と左詰めで表示される。これを人間が見た場合は、氏名と病名ということが容易にわかる。しかし、コンピュータが理解して、正しく病名と氏名をデータベースに格納するためには、

```
<name>鈴木一郎</name>
<diagnosis>胃癌</diagnosis>
```

の様に、医学的な意味を明示するタグを用いたほうが処理しやすい。この例で使った<name>、<diagnosis>などのタグを使って、医療情報を伝える技術、これがMMLだ。私達は、カルテの論理的な構造を検討し、その中で使用される医学の意味を表わすタグを決め、タグ同士の階層構造を決めた。これにより、SGML/XML技術を用いて文書型定義（DTD: Document Type Definition）を作ることが出来る。MMLは、このDTDに他ならない。施設間でデータを交換する場合、これらを用いたSGML文書に変換してやり取りすれば、正しく医療情報を伝えることが出来る。

1.2 診療データの所在管理：GMHD(Global Electronic Health Record)[4,5]

地球上に存在する医療機関が何らかの方法で結ばれ、地球規模で分散した病歴データを、あたかも一冊のカルテの様にひとつのまとまりとして見る事が出来る（図2）。これがカルテの電子化の最大のメリットであると思われる。複数の医療機関に分散された患者情報を、いかにしてまとまった病歴として見せることができるかという問題を解決するために、受診歴データベース（以下GMHD: Global Medical History Database）を提案する。これは医療情報ディレトリサービスの一種とも言える。

これまで、ICカード、光カード等のメディアに病歴情報を記録する方法の試みがあるが、病歴情報は膨大であり、おのずからこれらのメディアには容量の点で限界がある。超大型コンピュータを使って、医療情報センターの様な集中型のデータベースを運用する考えもあるが、検索要求の集中による検索のレスポンス低下など、多くの問題があり、やはり分散型管理が現実的な選択であろう。では、地球上に分散されたデータを「一つのカルテ」として見るためにはどうするか？これを解決するためには、以下の2つの前提が必要となる。

前提1：受診歴の管理。

患者が、これまでどの医療機関を受診したかが判ること。これは、細かいレベルは必要でなく、病院単位で良い。いつ受診したかも必要ではない。したがって、持つべき情報としては、受診したことのある病院数、すなわち高々20レコード程度の情報である。この情報を受診歴データベースGMHDに保持する。

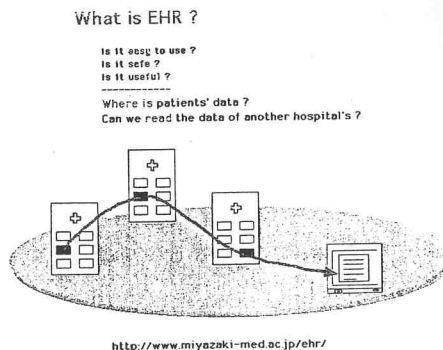


図2 分散記録されたデータのコントロール
個人情報に複数の医療機関に分散している。これをあたかも一つのカルテの様に見せる技術が必要である。これがないと電子カルテの価値は半減する。

前提2：システム間汎用インターフェイス

個々の電子カルテシステムが、システム間汎用インターフェイスを持つこと。つまり、他施設からの患者情報検索要求に対応出来るような共通のインターフェイスを装備する必要がある。具体的にはMMLなどを使ったインターフェイスである。

これらの前提が満たされると、ネットワークに接続されたシステムであればどの病院に行っても、自分の病歴は一つのまとまりとして参照することが出来る。

図3に示した如く、初めて受診する病院に行き、患者のIDを入力すると、

- 1) システムは、患者のデータが自分のシステムに存在するかどうかをチェック。
- 2) 自動的にGMHDに問い合わせ、その患者の病院受診歴を検索
- 3) 受診歴に従い、自動的に他の病院システムと会話し、データをMML形式で取得
- 4) 他施設から受け取った全てのデータをローカルシステムで表示

というプロセスを実行する。つまり、患者データは、見かけ上一つにまとまったものとして参照可能となる。図4にそのイメージを示したが、上記一連のプロセスはユーザーには意識されず、単に患者の病歴が表示されるだけであるが、よく見ると、病歴（ドキュメント）は、様々な医療機関で作成されたものであることがわかる。

1.3 小規模医療機関でも無理なく使うことの出来るシステム

これまで述べてきたように、異なる電子カルテシステム間での情報交換をMMLで実現し、分散した情報の所在管理をGMHDで行うと、理論上、広域電子カルテシステムが可能となる。しかし、個人開業

The concept of Global Medical History Database (GMHD)

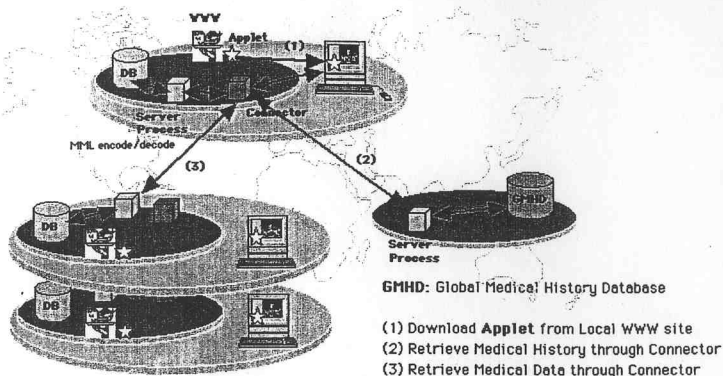


図3 GMHDの概念図



図4 広域対応の電子カルテのイメージ図

医などの様な小規模医療機関でLAN、電子カルテサーバー、MMLインターフェイス、データのバックアップなどを行うのは運用上、経済上非常に重い負担となる。そこで我々が提案し、プロトタイプを作成しているのが、共同利用型の医療情報システムである。このシステムは以下のコンポーネントから成る（図5）。

1) MMLインターフェイスを持つ共同利用データベース

図5に示したように、大規模医療機関や、小規模機関で発生した診療データは、夜間に自動的にMML文書に変換され共同利用データベース（以下DB）に送られる。DBにはMML文書を解析するMMLインターフェイスが装備されており、MML文

書を解析し診療データを抽出しデータベースに格納する。医院や病院から閲覧要求があった場合は、同様にMML文書を作成し相手に送付する。また、患者からの閲覧要求に対しては、HTML形式でデータを送信する。従って、患者は特殊な環境を用意する必要がなく、ごく普通のパソコンで十分である。データの媒体にMML文書を使うので、このデータベースを利用する機関の電子カルテシステムは、必ずしも同じデータベース構造を持つ必要はない(異なるベンダーの製品でも良い)。

MML文書を作成/解析するインターフェイスを持つだけで良い。

2) サービスゲートウェイ

主として患者からのサービスに應えるためのゲートウェイ。セキュリティチェックも行う。電子カルテ情報の閲覧のみならず、問診情報、メール、掲示板などのサービスを提供する。将来は、小規模機関が使うコンピュータアプリケーション等(電子カルテアプリケーション、医事計算アプリケーション、各種マスター類など)のデジタル資源を中央管理とし、バージョン変更などの際、Push技術を用いて夜間自動的にリモートでパソコン環境を整備することも考えられている。また将来、毎月のレセプト作成を代行するサービスも可能である。

3) クライアントコンピュータ

前項でも述べたように、小規模医療機関は各々好みの電子カルテシステムを導入し、MMLインターフェイスを装備する。患者はごく普通のパソコンにWWWブラウザをインストールしておくだけで良い。将来、Push技術を用いたアプリケーション配信サービスがスタートした場合は、MMLインターフェイスを実装する必要もなくなる。なぜなら、これらのサービスから配送されるアプリケーションは中央から見て既知のシステムであり、中央のDBとの間で正確に診療情報を交換出来るからである。

以上述べたパイロットシステムは、一見集中型に見える。しかし、診療データの発生源はあくまでも各医療機関であり、細かなデータ(特に医事関連やオーダー情報)は各医療機関に保持される。中央のDBに格納されるデータは、MMLで規定された比較

標準化された医療情報の利用

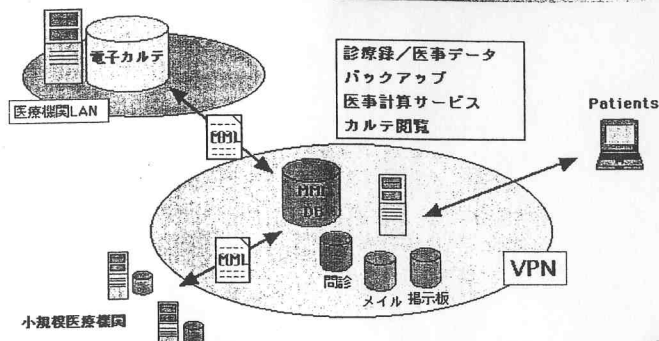


図5 共同利用型の医療情報データベースシステム

データベースへのデータ入出力は全てMML形式で行う。蓄えられた診療データは、セキュリティを保ちながら、医療機関、患者、行政などが利用する。インターネット上ではあるが、利用者から見ると閉鎖されたLAN(VPN)として安全に利用出来る。

的粒度の粗いものであり、診療録として最低限必要とされるデータを保存することになる。医事システム等の混在するアクティブな電子カルテ環境では、診療録の電子保存の前提である「改ざん防止」などはなかなか担保し難い。しかし、このような仕組みを使えば、これらの条件を担保することが可能となるだろう。診療録の原本を中央のバンクに預ける、と言った感覚である。

この「共同利用型診療録保存システム」は、恐らく各都道府県医師会単位で運用されることになると思われる。当然、特定の個人データが複数の都道府県に分散記録されるケースも出てこよう。その時はGMHDの様な医療情報ディレクトリサービスが効果を発揮するものと思われる。

2. SGML/XMLを使った情報交換規約の現状

昨年までは、MMLはSGMLで表現されていた。しかし、データベースとの親和性を強く意識したXMLの発展を受けて、MMLはSGMLからXMLへと変貌している。SGMLと比較して、XMLは次のような大きな特徴を持つ。

- 1) 終了タグが必須
- 2) Hyper Linkが可能
- 3) NameSpacesの概念

例外なく終了タグを持つことで、プログラム処理がシンプルになった。また、Hyper Linkの機能によって、URI指定でインターネット上のリソースの利用が可能になった。

特筆すべきはNameSpaces（名前空間）の機能である。これはプログラムのメインルーチンとサブルーチンの関係に酷似している。すなわち、一つのXMLインスタンス内で、複数のDTDの利用を可能とするものである（本年3月に、W3Cより、namespaces in XML version 1.0規格書が発表）。

医療情報交換規約としてのMML設計のジレンマとして、情報を詳細に定義すると、DTDが巨大となり管理が困難となるが、一方、定義を粗くすると記述力が不足しているとの批判を受ける。様々な専門分野を包括する医療情報の交換規格は、単一の構造では対応が難しい。そこで、MMLの最新バージョンはnameSpacesの考え方を取り入れ、MML本体の中から、必要に応じてsub-DTDの組み合わせを使うことが可能な設計とした（図6）。この方法により、部分的なバージョン変更が本体に影響を与えなくなったばかりでなく、経過記録部分などで使われるであろう、専門分野特有のDTDの作成の分業化が可能になった。今後は、各学会などの専門分野からDTDが提案され、インターネット上に専門別のDTDライブラリとして管理されることになると思われる。

Namespaceの具体的な使用法（CLINICAL-ENCOUNTER-SECTIONに外科手術記録を記載する場合）は次の通りである。

```

-----
XMLインスタンス（例）
-----
<CPR-MML>
省略
<CLINICAL-ENCOUNTER-SECTION
xmlns:surgery="http://www.miyazaki-
med.ac.jp/mml/DTD/surgery.dtd">
<RECORD-INFORMATION-SECTION>
省略
</RECORD-INFORMATION-SECTION>
<surgery:opdate>19990526</surgery:opdate >
<surgery:operator>鈴木一郎</surgery:operator>
<surgery:method>幽門側胃切除</surgery:method>
<surgery:openotes>
  気管内挿管全身麻酔下に、上腹部正中切開を
  行い開腹し、、、、（省略）
</surgery:openotes>
</CLINICAL-ENCOUNTER-SECTION>
省略
</CPR-MML>

```

surgery.dtdを使用したいセクションにnamespace宣言を行い、URIを記載する。同セクション内では、MMLのDTDとsurgeryのDTDの混在が可能である。

MML DTDに基くインスタンス

Surgery DTD に基くインスタンス

```

<CPR-MML>
<SECTION
xmlns:surgery="URI">
<surgery:opdate >19990526</opdate >
<surgery:operator> 宮崎義彦</operator>
<surgery:method> 幽門側胃切除</method>
<surgery:openotes >
  上腹部正中切開にて開腹し、
</openotes >
</SECTION>
</CPR-MML>

```

```

<opdate >19990526</opdate >
<operator> 宮崎義彦</operator>
<method> 幽門側胃切除</method>
<openotes >
  上腹部正中切開にて開腹し、
</openotes >

```



図6 NameSpacesの使用概念

手術記録用のDTD(surgery.dtd)を設計し、DTDリソースとして所在を公開しておく。 報銀行)

3. 個人情報管理サービス (情

本論文では、個人の医療情報を蓄積する共同利用型のデータベースについて述べた。近い将来、光ファイバー網の発達で各家庭が結ばれ (FTH: Fiber to The Home)、今では考えられない程の通信インフラが生まれると、本論文で述べた様な共同利用データベースはその守備範囲を拡大し、家庭のパソコンで発生したデータのバックアップサービスとして機能するようになるのではないかと。現在、どの家庭も家に現金を貯えることはまずない。大半を銀行に預けている。これと同じようなことが起こるのではないかと。医療機関を訪れた患者は、診療記録を自分の情報銀行口座に納入することを要求するようになるかも知れない。その口座には、単に医療記録だけではなく、健康記録、メール、写真、登記簿謄本など、生活に絡む全ての情報を蓄積し、その利用は所有者本人がコントロールするのである。病院は記録のコピーを期限付きで使わせてもらうということになるかも知れない。

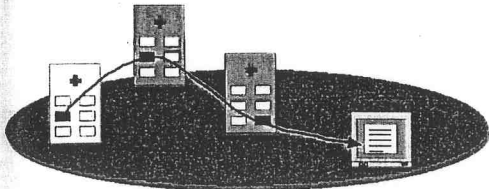
【参考情報】

- 1) 吉原博幸, 他: 医療情報の交換手順の標準化に関する研究, <http://www.miyazaki-med.ac.jp/medinfo/SGmeeting/document/kousei97MMLreport/>
- 2) 吉原博幸: 診療情報の電子化- 見えてきた電子カルテ-, <http://www.miyazaki-med.ac.jp/9807GYNweb/>
- 3) 日本医療情報学会電子カルテ研究会, MML Ver.1.0 β 3 / 規格書 シーガイアバージョン β 1, <http://www.miyazaki-med.ac.jp/medinfo/SGmeeting/document/SG97/02Ohe/ohe.html>
- 4) 吉原博幸, 他: Global Electronic Health Record, 分散医療データの一元管理を目指す, <http://www.miyazaki-med.ac.jp/medinfo/SGmeeting/document/SG97/24GMHD/GMHD.html> (和文)
- 5) Hiroyuki Yoshihara, et. al: The concept of Global Medical History Database (GMHD) and Its prototype, http://www.miyazaki-med.ac.jp/medinfo/SGmeeting/document/GMHD_Sympo97/GMHD.html (英文)
- 6) MMLワーキンググループ: MML DTDのコンポーネント化—Namespaces in XMLの活用—, <http://www.miyazaki-med.ac.jp/namespaces/>

What is EHR ?

- Is it easy to use ?
- Is it safe ?
- Is it useful ?

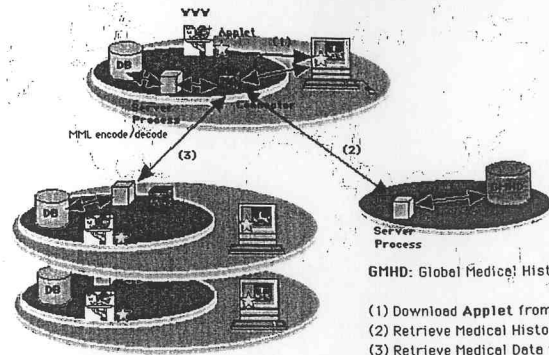
Where is patients' data ?
Can we read the data of another hospital's ?



<http://www.miyazaki-med.ac.jp/ehr/>

図2 分散記録されたデータのコントロール

The concept of Global Medical History Database (GMHD)



GMHD: Global Medical History Database

- (1) Download Applet from Local WWW site
- (2) Retrieve Medical History through Connector
- (3) Retrieve Medical Data through Connector

図3 GMHDの概念図

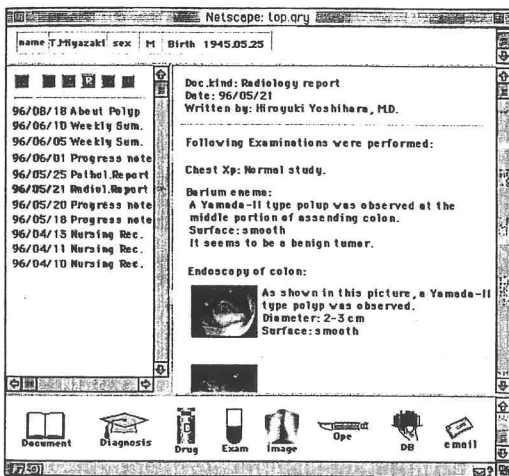


図4 広域対応の電子カルテのイメージ図

標準化された医療情報の利用

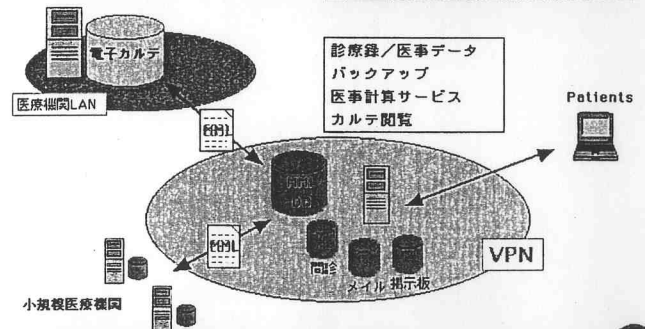


図5 共同利用型の医療情報データベースシステム
データベースへのデータ入出力は全てMML形式で行う。蓄えられた診療データは、セキュリティーを保ちながら、医療機関、患者、行政などが利用する。インターネット上ではあるが、利用者から見ると閉鎖されたLAN(VPN)として安全に利用出来る。

MML DTDに基くインスタンス

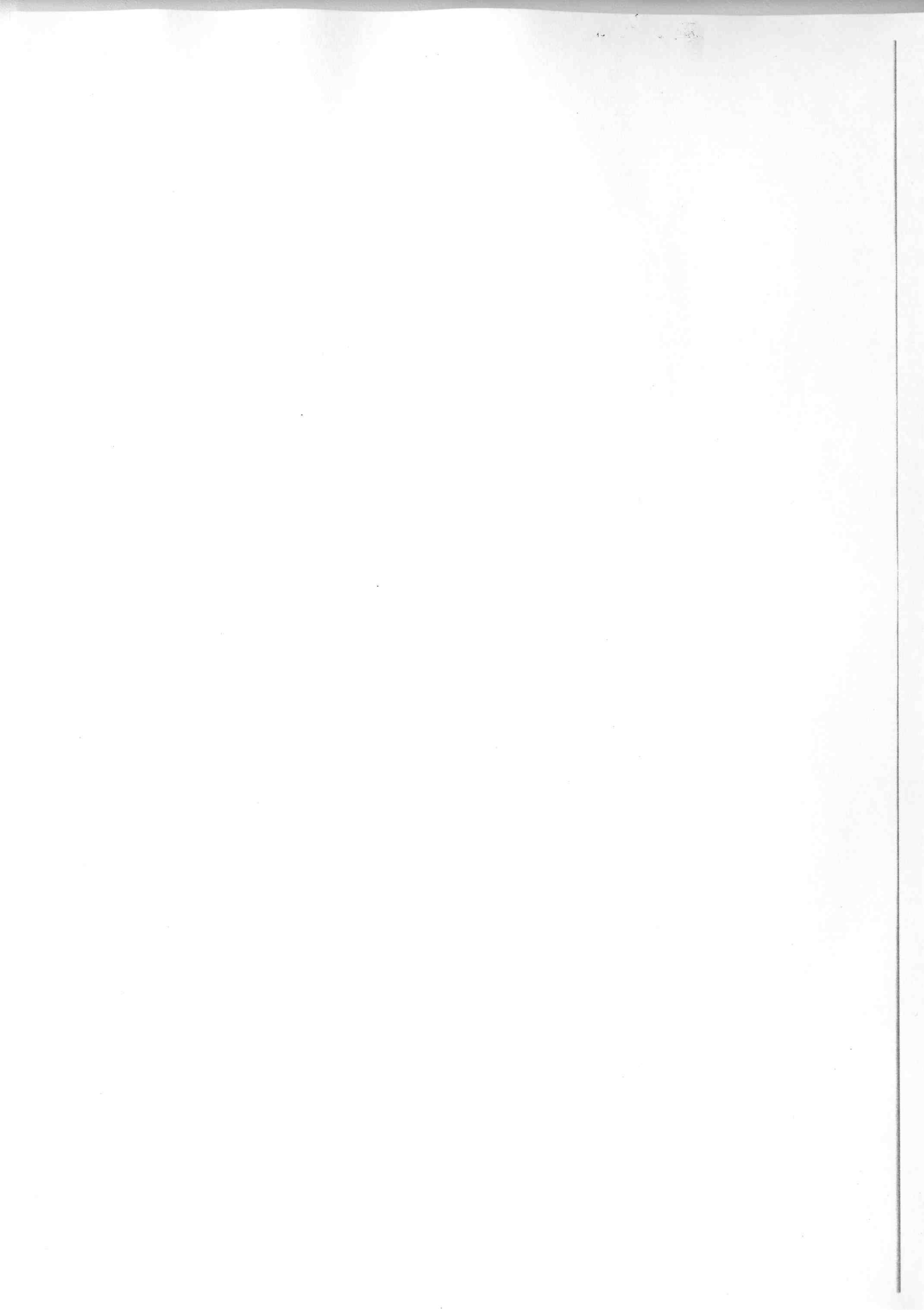
```
<CPR-MML>
<SECTION
xmlns:surgery="URI">
<surgery:opdate >19990526</opdate >
<surgery:operator > 宮崎義彦</operator>
<surgery:method> 幽門側胃切除</method>
<surgery:openotes >
  上腹部正中切開にて開腹し,
</openotes >
</SECTION>
</CPR-MML>
```

SurgeryDTD に基くインスタンス

```
<opdate >19990526</opdate >
<operator > 宮崎義彦</operator>
<method> 幽門側胃切除</method>
<openotes >
  上腹部正中切開にて開腹し,
</openotes >
```



図6 NameSpacesの使用概念



病院情報システム——これまでとこれから——

山本 和子

島根医科大学医学部医学科医療情報学

〒693-8501 島根県出雲市塩冶町 89-1

yam@shimane-med.ac.jp

1. はじめに

今年は医療情報学の発展に多大の貢献をされた元島根医大学長の平川先生がご逝去されましたので、昔のことを思い出す年になりました。古いことで申し訳ございませんが、Mとの出会いは1975年5月30日に京都市で開催された第1回MUMPSグループ研究会(平川先生会長)にたまたま出席したことから始まります。当時すでにMによる病院情報システムを導入しておられた京大病院、羽曳野病院、中京病院などの勇気ある決断に今更ながら敬意を表したいと思います。その年の9月に若井先生に誘われて米国病院情報システム視察団に参加して、米国を横断しながら病院情報システムを見学し、第4回米国MUMPSユーザーグループ大会に出席後、最後にマサチューセッツ総合病院(MGH)を訪問しました。丁度COSTAR(Computer-Stored Ambulatory Record)が完成した頃で、医師はチェックシートに記入し、それをオペレータが入力していました。端末の安さ、台数の多さ、手書きカルテが無かったことに驚き、医療の質の保障の時代に入ったという言葉が印象的でした。

2. Mの発展の時代

その後わが国においても、MUMPSを用いた病院情報システムの完成期に入ります。米国から帰国後、大阪医大でMによる入院カルテ管理システムを開発しました¹⁾。手書きカルテの貸し出し管理とともに、COSTARをすっかり真似てチェックシートによる退院時確定診断名の半自動コーディングと退院サマリ情報の入出力システムを作りました。プログラム開発を省力化するために、図1に示しましたように、データをパターン別に分けてデータ入力から参照・検索、統計解析ができる汎用プログラムパッケージを開発しました。

1983年に福井医大に赴任し、MUMPSによる病院情報システムを導入しました²⁾。User Orientedに開発するという事で開発規定を作り、職員にMを教育し、職員でプログラムを変更・作成できるようにしました。コンピュータによる理想郷を作るのが夢でした。病名コードより病名コードを探してコード入力していた時代でした。「記憶するという作業はコンピュータにまかせ、主体的に考える作業だけを人間がするようにシステムを設計する」のが基本方針で、随所に情報検索機能を付けました。病名自動コーディングシステムを開発しICDコードから病名を検索できるようにしました。また、正しい情報を入力するために、

入力時にデータチェックを掛けたり、ガイダンスを表示したりする機能を付けました。ところがこの「チェックを掛ける機能がある」ということが分かると、ありとあらゆるところにチェックを掛けてほしいという要求が出て、要求通りにしていると、まるで人工知能のような医事システムが完成しました。処理速度を気にしなければならない毎日でした。チェック機能に安心し、人間はただ入力機械のように何も考えずに作業するようになりました。挫折感だけが残りました。

医事システムから日々データを転送して病歴情報システムも開発しました。オーダーリングも始め、新しいことに次々挑戦して皆様に迷惑をかけた時代でした。

この頃にMUMPSはM言語と名称変更され、MUMPSユーザーズグループはMテクノロジー学会に名称変更されました。Mは標準化されISOになりました。大櫛先生を中心に作業班を作りJISにすることもできました。

3. ネットワークの時代へ

1992年8月に島根医大に赴任しました。この頃からコンピュータ環境は急激に変化し出しました。コンピュータのダウンサイジングが進行し、GUI(Graphical User Interface)と自動プログラミング技術が開発され、通信技術の発達によりネットワーク時代へと進んでいます。本学でも医療情報ネットワークシステムを開発しました³⁾。一方で、病歴への執着忘れ難く、本年になってようやく人並みの病歴情報システムを完成させました。この20年間で手がけた病歴情報システム一覧を表1に示します。これらの病歴情報に経過記録が加わって電子カルテデータベースとなります。外来・入院の経過記録と退院時記録は、地域医療機関との通信を考えて、図2のような形に統一しました。図1と比較して、チェックシートがテンプレートへと発展し、入力者が医師になりましたが、画像が入った以外、基本は殆ど変わっていません。端末側GUIはVisual Basicを用いた入力画面(図3)と、インターネットのWWWブラウザを用いた参照画面(図4)の共存となります。

MがCacheに統一され、WebとのLinkができました。本学の病院情報システムはまだU-MUMPSで稼動していますが、地域医療情報ネットワークシステムはCacheになりました。

4. 今後の動向

インターネットは急速に普及してグローバルネットワークの時代を迎えつつあります。機種に関係なく利用できるという意味において、端末側GUIはWebのブラウザに統一されて行くかも知れません。この時期に、厚生省から電子保存通達⁴⁾が出て電子カルテも急速に普及しそうな気配を見せています。本学でも地域医療機関で情報を共有するための地域医療情報ネットワークシステムを開発しました⁵⁾が、本システムは電子カルテと表裏一体となっています。開発してみて感じた事は、

①広域医療情報ネットワークは実現可能である。

②患者の治療を目的とした広域化対応電子カルテ⁶⁾と学術目的(患者識別情報消去)の症例データベースを構築することができる。

③症例データベースはこれまでのような簡単なものではなく、分散型検索キーデータベースを作成しておくだけで、総ての電子カルテ情報を引き出すことができるので、医学の発展並びに医療に大きく寄与すると期待できる(図5)。

④分散型ネットワークであるので負担は少なく維持管理は楽である。

というようなことが分かりました。ただし、それを実現するためには、

①漢字入力の負担が大きい。医学用語を入力するための辞書が必要である。

②Problem別のテンプレートが必要である。国レベルで統一しなければならない。

③1人1台の携帯端末が必要であり、携帯端末で病院情報システムを利用できるようにしなければならない。

などが上げられると思います。今後の発展を期待しております。

参考文献

- 1) 山本和子:大阪医科大学附属病院の入院病歴システム,第19回大阪大学BME研究会資料-病歴のコンピュータ処理の方法とその問題-,9-14,1981.
- 2) 山本和子他:福井医科大学総合医療情報システムの開発運用とその評価,医療情報学,9(1),21-31,1989.
- 3) 山本和子他:WAN環境に対応できる医療情報ネットワークシステムの構築,第16回医療情報学連合大会論文集,372-373,1996.
- 4) 松本義幸:診療録および診療に関する諸記録の電子保存について,医療情報学,Vol.19 Suppl. May 1999
- 5) 山本和子:SGML形式による患者情報の地域共有について,Seagaia Meeting '99,25-28,1999.
- 6) 吉原博幸:Global Electronic Health Record 分散医療データの一元管理を目指す,Seagaia Meeting '97,87-93,1997.

図1. 病歴データベースの概念

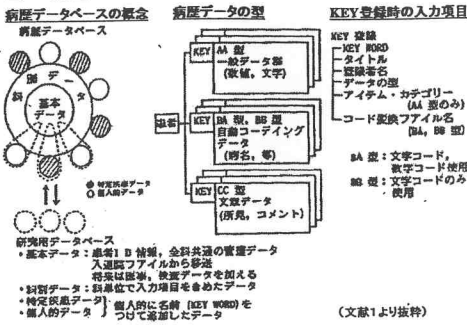


表1. 在籍時に開発した病歴情報システム一覧

項目	大阪医大 1979年	福井医大 1987年	鳥取医大 1999年
患者基本情報参照・検索	○	○	○
入院歴	○	○	○
外来受診歴	○	○	○
外来・入院病名参照・検索	○	○	○
退院時確定診断名検索	○	○	○
病理診断名検索	○	○	○
薬歴・検査歴参照・検索	○	○	○
診療行為参照・検索	○	○	○
退院サマリ参照・検索	○	○	○
病理診断報告書	○	○	○
放射線画像診断報告書	○	○	○
血液型・感染症・アレルギー	○	○	○
身長・体重	○	○	○
画像参照	○	○	○
地域医療・学外から参照	○	○	○

図2. 電子カルテデータベースの概念

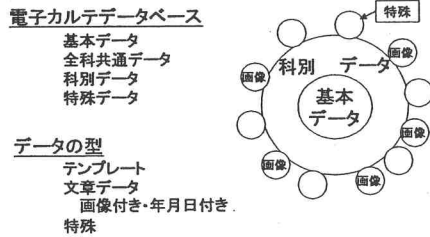
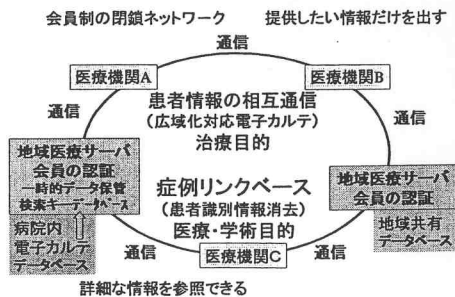


図3. 退院時サマリの入力画面例

図4. 退院時サマリの参照画面例

図5. 広域医療情報ネットワークの概念図



MSM Server/Workstationによる健診システムの開発

大櫛陽一

東海大学医学部医用工学情報学

〒259-1193 神奈川県伊勢原市望星台
TEL:0463-93-1121 ext. 2140 FAX:0463-96-4301
youichi@keyaki.cc.u-tokai.ac.jp

1. はじめに

MSM を使うと、M 言語だけでサーバ/クライアント・システムの構築、Windows-GUI アプリケーションの開発が可能である。古くからのM言語に馴染んだ開発者にはありがたい。各種ユーティリティがGUI化されており、デバッグ開発環境も充実しているため、短時間でのアプリケーションの開発が可能である。また、実行速度も極めて良好である。このため、開発時間が限られたアプリケーションや、実行時間が優先されるシステムには最適な道具である。なお MSM Server はサーバ側のシステムであり、MSM Workstation はクライアント側のシステムである。

2. DB機能

リモートDBのマウントと、ローカルDBのマウントが同時に可能であり、サーバのみにDBを持たせるシステム、サーバとローカルの両方にDBを持たせるシステム、ローカルDBのみのシステムなど多彩なシステム構成が可能である。また、リモートDBへのアクセス時間は非常に早く、サーバでのローカル処理の数倍程度の時間で処理できる。Mアプリケーションのコンパイル、Exe ファイルの作成が出来るため、クライアントでのアプリケーション起動はアイコンをクリックするだけでよい。このため携帯型アプリケーションなどのように、ローカルDBのみのシステムでは、DB始動/停止などの操作は不要であり、ユーザの操作性は極めて良い。また、MSM Workstation Professional では、クライアント台数に制限がないため経済的である。リモートDBへのアクセスは、M言語標準である Distributed Data Processing (DDP) を使って行う。例えば、システム名(SYS)が"YOZ"で、DBグループ名(UCI):Name Space が"NIS"にある、個人情報のファイル(^ZFKOJIN)の、個人番号(ID)が12345678のデータを、ローカル変数 X に取り出すには次のコーディングとなる。

```
SET SYS="YOZ", UCI="NIS", ID=12345678
```

```
SET X=^|UCI, SYS|ZFKOJIN (ID)
```

3. 開発環境

MSM Workstation では、次のビジュアルな開発環境が提供されている。

(1) File... Exe ファイルの作成など

New, Open, Save, Compile, Test, Make. EXE, Import, Export など

(2) Control... GUI 画面作成

ActiveX, Button, Checkbox, Combo Box, Date, Document, Drop-Down Combo, Drop-Down List, Generic, Group Box, Label, List Box, Long List, Tabbed Group, Picture, Radio Button, Scroll, Spin Button, SQL Data, Text Box

* Generic では、文字数値表示、グラフィック、画像などのすべてを描ける。

(3) Tool... M 言語各種ユーティリティ

Command Window (M 言語の直接実行画面)、Routine (ルーチン関係の各種ユーティリティ)、Global (グローバル関係の各種ユーティリティ)

(4) System... DB 関係ユーティリティ

Local Mount, Remote Mount, Unmount, Create, Expand, Validate

これらの機能は、Windows NT Server のリモートアクセス機能を使った遠隔接続でも使えるため、リモート保守ではルーチンやグローバルの修正に加えてテストランも可能である。

4. ベンチマークテスト結果

ベンチマークを実施した環境は次の通りである。

(1) サーバ

IBM-PC 365 Intel Pentium Pro 200MHz, 256KB キャッシュ, 192MB ECC メモリ
Windows NT 4.0 Server (Service Pack 4)

MSM Server 4.3.1/J グローバルキャッシュ・サイズ設定値 256KB (Default)

(2) クライアント

FMV-Desk Power SP Intel Pentium 133MHz, 64KB キャッシュ, 48MB メモリ
Windows 95

MSM Workstation 1.1.0A

(3) 通信方式

TCP/IP Ethernet 10Mbps

MSM DDP

ベンチマークの結果は次の通りである。

(1) グローバルのセット SET ^W(I)=""

- ・サーバ単独処理時間 0.031m 秒
- ・リモート処理時間 0.250m 秒... 約8倍

(2) グローバルの読み出し SET X=^W(I)

- ・サーバ単独処理時間 0.027m 秒
- ・リモート処理時間 0.110m 秒... 約4倍

(3) グローバルの\$ORDER SET ID=\$O(^ZFKOJIN(ID))

* 実個人基本情報 (14,000人、全容量29MB、1人平均57.5B)

- ・サーバ単独処理時間 1.5m 秒
- ・リモート処理時間 8.1m 秒... 約5倍

この結果は、ネットワーク・オーバーヘッドやサーバとクライアントの処理能力差を考慮すれば十分な処理スピードであろう。また、1台のサーバで数台のクライアントが同時処理をしても、クライアントから見た処理スピードが維持されることも示しており、通常の運用からすると1台のサーバに数10台の接続環境でも十分なサーバ処理能力が発揮されるものと思われる。

5. 健診システム

山形県西川町（人口約9000人）は、早くから保健医療福祉サービスの充実に努力しており、情報システムの導入にも熱心な町である。すでに本学会などで保健指導システムや予防接種関連システムの開発について報告をしている。(1)-(7) 今回は、MSM Server/Workstation を使って健診システムの再開発を行ったので報告する。

(1) 機器構成

- ・サーバ COMPAQ Proliant 1600 II/300 Pentium II 300 128MB ECC メモリ
アレイ・ディスク (RAID5) 4. 3GB × 3台
UPS 付き
Windows NT4. 0 Server (Service Pack 4)
MSM Server 4. 3. 1J
- ・クライアント FMV-BIBLO MMX-Pentium 200MHz 64MB メモリ
HDD 2. 1GB
Windows NT4. 0 Workstation (Service Pack 4)
MSM Workstation Professional 1. 1. 0A
現在の端末台数 9台

(2) Network

- ・サーバ/クライアント間 Ethernet/10BaseT TCP/IP による MSM-DDP 通信
- ・遠隔保守 WindowsNT Server - RAS MSM-Workstation からリモートマウント

(3) アプリケーションの概要

すでに開発済みの健康データベース、保健指導システム、保健指導記録登録システム、予防接種システムを統合し、次のアプリケーションを追加開発した。

(3. 1) 健診結果の入力

日帰りドック、1泊2日ドック、国保ヤング健診、共済健診、職員健診の5つの健診コース

(3. 2) 健診結果報告書の印字

(3. 3) 医療機関カルテ番号の登録と検索

(3. 4) 外部健診結果のFDからの入力

成人病検診、老人保健基本健診

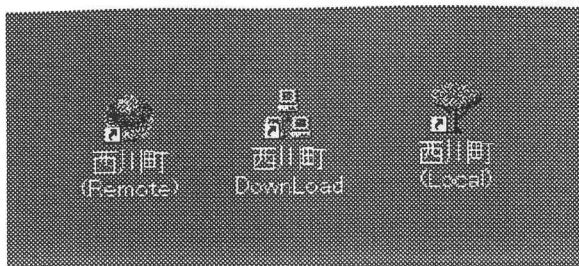
(3. 5) 各健診結果を老人保健基本健診判定基準で再判定する機能

(3. 6) 健康データベースへの登録

(3. 7) サーバから携帯端末へのデータ・ダウンロード

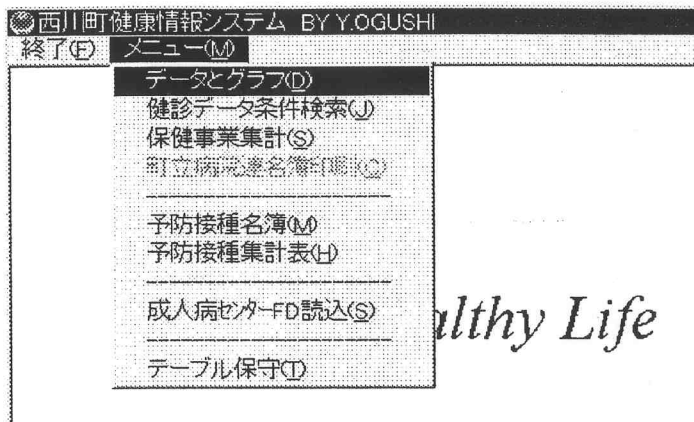
代表的な画面と帳票例を次に示し説明を行う。

(a) クライアント側のアイコン



通常は (Remote) アイコンを使って業務を行い、出張保健指導の前に DownLoad アイコンでデータのダウンロードを行い、(Local) アイコンでローカルDBを使った保健指導を実施する。1年分のダウンロード時間は約5分、1ヶ月分では約30秒と高速である。

(b) メニュー画面



このような Windows 画面は、`^$WINDOW` というシステム・グローバルをセットするだけで表示ができる。(MWAPI) "NI" という名前を付けたこの Window 画面表示ルーチンのコーディングは次の通りである。

`SET ^$W("NI", "SIZE")="640, 480, PIXEL" ...` 画面サイズをドット単位で設定

`SET ^$W("NI", "POS")="0, 0, PIXEL" ...` 画面左隅位置を指定

`SET ^$W("NI", "TYPE")="APPLICATION" ...` 画面の種類を指定

`SET ^$W("NI", "MODAL")="APPLICATION" ..` 画面の特性を指定

`SET ^$W("NI", "TITLE")="西川町健康情報システム" ...` タイトルバー表示

`SET ^$W("NI", "EVENT", "CLOSE")="EXIT^HM" ...` Window クローズ処理先

メニュー/サブメニューの内容や、画面中のオブジェクトも同様なコーディングで表示できる。

(c) 個人検索とファンクション・ボタンの画面

No	氏名	生年月日	歳性	町内会	住所
1	西川 太郎	S39.01.13	35 女	海味第二	海味395番地
2	西川 太郎	S15.06.26	53 男	海味第三	海味543番地3
3	西河 太郎	S01.01.13	73 男	梅沢第一	睦合丙55番地

個人検索キーとしては、姓名(ローマ字、カタカナ、漢字)、住民台帳番号、カルテ番号)のいずれも使用可能である。同一キーを持つ人のリストが表示される。該当の人を選択してから、ファンクションをクリックする。

(d) 健診結果入力メイン画面

ここではまず日付を入力する。該当日の健診受診歴があれば、健診区分が自動選択されて表示される。無ければ、区分の一覧から選択する。臨床検査技師、X線技師などが各健診結果を登録後、医師が承認登録を行い、結果報告書が印刷される。医師の承認登録が無ければ報告書の印刷が拒否される。また、医師の承認後は、健診結果の登録/修正が禁止される。

(e) 健診結果登録画面例

胃部X線検査結果登録 1999.06.24		
(E3)H11.02.25 西川 太郎(男)S15.06.26(58才) カル#:		
胃部チェックF No.	*胃部精検項目	
No.1	C 内視鏡	
No.2		
No.3		
胃部チェック部位1	胃部チェック部位2	胃部チェック部位3
食道	前大後小	前大後小
咽頭		
胃体部		
胃下部		
胃部チェック所見1	胃部チェック所見2	胃部チェック所見3
変形	壁硬化	
胃部X線所見病名1	胃部X線所見病名2	胃部X線所見病名3
A 食道憩室	C	0 異常なし
A 食道逆流	C 内臓逆位	0 内臓逆位
A 裂孔ヘルニア	C 食道変形	0 食道憩室
	C 食道ホリア疑い	
登録		

検体検査結果などの数値は Text Box、選択項目は List Box などを使って登録される。List box の選択項目リストは、サーバのテーブルを参照して表示している。クライアントに持たないため、保守性に優れており、遠隔保守が可能である。今回開発した特徴的なオブジェクトを紹介する。病名などのように大量のリストから選択する場合に、Combo Box で頭文字を入力すると、該当する項目に絞って再リスト表示するようにした。「胃部X線所見病名1～3」が、その例である。「胃部X線所見病名2」では、「C」を入力すると頭に「C」が付く項目に絞られている様子が見られる。

(f) ダウンロード画面

Down-Load of Personal Data

ダウンロード開始日
1999.03.19

開始

個人番号

前回のダウンロード日付が各クライアントのグローバルに記録されており、開始日付の規定値として表示される。開始ボタンをクリックすると、ダウンロードが開始される。動作モニターとして個人番号が表示される。実際には処理速度が速いため、個人番号の読みとりは困難であるが、正常動作の確認として使われている。ダウンロードが完了すると、その旨表示されると同時に、各クライアントのダウンロード日付が更新される。

(g) 健診結果報告書の印字例を次に示した。

検査結果報告書

白り帯のドック

SC:107 KC:1

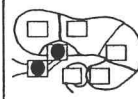
カルテNo. 123456
 受診日 H11.02.26
 氏名 西川 太郎 様 (男)
 生年月日 S01.01.13 (73 歳)
 住居 所 陸台西55番地
 電話番号 74-2159

要観察 [A]	疑陽性 精検不要 食老節室 切除胃
要注意 [B]	陳旧性胸腺所見 乳房皮膚の異常 低コレステロール血症
要精検 [C]	内痔核 中状腺腫大 胃外陰影
要治療 [D]	1. 股関節凝塊 高尿酸血症 貧血
治療継続 [D2]	心臓病
日常生活上の注意	偏食せず、穀分を多く食べて下さい。
連絡事項	全大腸内視鏡検査又は部分大腸内視鏡、注腸検査受診
視力 [B]	右 (矯正) 1.0 () 0 左 (矯正) 0.1 () B
聴力 [C]	右 所見なし 聴力低下 0 左 所見なし 聴力低下 C (1000Hz) (4000Hz)
色覚 [A]	色覚異常 A

診察 [O]	異常なし
身長計測 [O]	身長 172.0 cm 体重 68.0 kg 標準体重 66.0 kg 肥満度 2.0 %
血圧測定 [O]	最高血圧 130 mmHg 最低血圧 85 mmHg
尿一般 [C]	尿蛋白 (±) 尿糖 (+) 沈査 白血球 1~5 赤血球 6~10 円柱 異常なし その他
肝機能 [C]	GOT 10 IU/L GPT 10 IU/L γ-GTP 10 IU/L ALP 1 IU/L LDH 10 IU/L コリンエステラーゼ 10 IU/L 総ビリルビン 1.0 mg/dl 総蛋白 10 g/dl ZTT 10 KU A/G比
脂質 [B]	総コレステロール 100 mg/dl HDLコレステロール 10 mg/dl 中性脂肪 100 mg/dl 動脈硬化指数 9
腎機能 [D]	クレアチニン 1 mg/dl 尿素窒素 10 mg/dl 尿酸 10 mg/dl

心電図 [O]	右胸心
運動負荷心電図 [A]	疑陽性
眼底 [D]	右 正常 左 正常 両側網膜不能
血液 [D]	赤血球数 350 ×10 ⁹ /μl 血色素量 45 g/dl ヘマトクリット 15 % 白血球数 4300 ×10 ⁹ /μl 血小板数 30 ×10 ⁹ /μl
糖代謝 [D]	空腹血糖 (空腹時) 100 mg/dl HbA1c 10 %
肺機能 [O]	肺活量 4300 l %肺活量 105 % 1秒量 3000 l 1秒率 75 %
免疫血清 [C]	HBs抗原 (-) HCV抗体 (-) 梅毒血清反応 陰性 CRP定量 1 mg/dl
臓器超音波 [O]	肝臓 形態異常 膵臓 異常なし
その他 [C]	頭部CT検査 1:脳梗塞疑い 胸部CT検査 4:びまん性肺病変疑い

胃がん検診 [C]	胃部×線所見 食道癌 胃外陰影
大腸がん検診 [C]	便潜血反応1日 (-) 便潜血反応2日 (+)
内臓器 [D]	内痔核 内痔瘻所見 結腸ポリープ
胸部検診 [C]	胸腺×線所見 上左、上右 上中 慢性気管支炎 陳旧性胸腺所見
心臓 [C]	心臓比 50 %
眼底検診 [C]	II
乳がん検診 [C]	乳がん検診 乳房形態異常 乳房皮膚の異常 乳頭の異常 マンモグラフィー所見 良性乳腺腫瘍
骨粗鬆症 [C]	骨密度 : 45 g/cm ² 同年代比 : 95 %



○この次に検診を受けた後、医師の検診を受けてから、紛失しないように大切に保管して下さい。

6. まとめ

MSM Server/Workstation は従来からのM言語利用者には使いやすい開発環境である。Windows 画面作成、処理プログラム、通信などを標準 M 言語のみでコーディングでき、デバッグも分かり易いためである。これにより Windows アプリケーションの開発期間が非常に短縮される。また実行速度が早いため、ユーザにとって快適であり、経済的なシステム作りが可能である。遠隔保守も使いやすく、変更の多いシステムに適している。MSM Server/Workstation は現バージョンでも十分な機能を持っているが、今後とも引き続いて新しいプラットフォームへの適応などが行われることを期待する。

【参考文献】

1. 大櫛陽一、小林佐枝子、須貝昌博、奥山純二、土田伸、春木康男、岡田好一：西川町における携帯型保健指導システムの開発。第23回日本エム・テクノロジー学会大会論文集、68-71、1996。
2. 小林佐枝子、大櫛陽一、須貝昌博、奥山純二、土田伸、春木康男、岡田好一：個人指導と集団保健指導システムの開発。第16回医療情報学連合大会論文集、604-605、1996。
3. 大櫛陽一、奥山純二、飯野咲子、小林佐枝子、小川一博：市町村における予防接種業務のシステム化。Proceedings '97 of M Technology Association of JAPAN, 16-19, 1997.
4. 小林佐枝子、大櫛陽一、奥山純二、志田明子、渡辺敬子、安藤千恵：保健指導における情報システムの利用と評価。日本公衆衛生雑誌、44(10特別付録)、262、1997。
5. 奥山純二、大櫛陽一、小林佐枝子、志田明子、渡辺敬子、安藤千恵：山形県西川町健康情報システムの構築。日本公衆衛生雑誌、44(10特別付録)、362、1997。
6. 志田明子、大櫛陽一、佐藤勝男、小川一博、小林佐枝子、奥山純二、渡辺敬子、安藤千恵：保健指導における情報システムの活用。田尻町スキップセンター開設記念フォーラム'98抄録集、71-73、1998。
7. Ogushi Y, Haruki Y, Okada Y, Takahashi M, Shimizu M, Izumi Y, Watabe T, Kobayashi S, Okuyama J and Kurita Y : Development and Evaluation of Regional Health Database Systems. Proceedings of the Ninth World Congress on Medical Informatics. IOS Press, Amsterdam, 1998. pp1297-1300

病診連携ベッド空床数照会システムの開発経験

沢田 潔、川田 新一、木下 元一、浅井 広、岸 真司
名古屋第二赤十字病院 医療情報部
〒466-8650 名古屋市昭和区妙見町 2-9
TEL:052-832-1121(内線 3066) FAX:052-832-1130
e-mail : sawa@nagoya2.jrc.or.jp

1.はじめに

当院は、国の救命救急センターとして地域救急医療の中核を担いながら、1990年から地域の診療所との間で病診連携システムを運用している。病診連携システムにおける当院の主な役割は入院治療である。しかし、名古屋市内最多の救急患者を受け入れている当院は慢性的な満床状態にあり、病診連携登録医への入院不応需が頻発していた。このことは病診連携システムの充実発展に大きな障害となっていた。病診連携ベッド(以下連携ベッド)は、この登録医への入院不応需解消を目的に1997年6月から運用を開始し、登録医の緊急に入院が必要な患者を登録医の判断で入院させることのできる登録医専用病床である。今回、連携ベッドの空床数をインターネットにて登録医に提供することを目的としてシステムを開発した。

2.システム概要

①連携ベッド入退床登録

オーダエントリ入退院管理システム上で連携ベッド利用患者が入院された際に病棟にて入床登録を行う。また、一般病床へ転棟・転室の際には退床登録を行う。この入退床情報は下記の集計用サーバ対しM-DDPにて転送する機能をオーダエントリシステム主管ベンダとの仕様とした。

②空床数集計

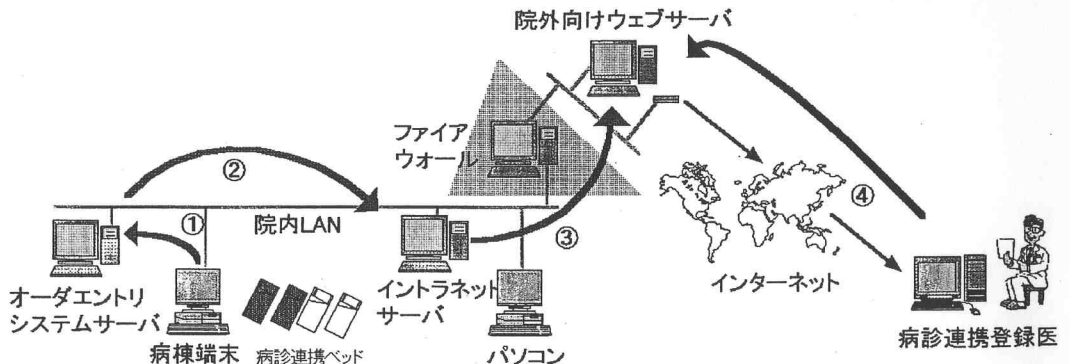
DDPにて転送された情報は、M言語システムが動作するイントラネットサーバでバックグラウンドJOBプログラムにて蓄積と集計を行い、下記の空床数提供サーバへFTP転送する。同時にこのサーバ内のウェブ機能によって、イントラネットにおいても空床数を閲覧できる仕組みとした。

③院外向け空床数提供

空床数情報ファイルは一般的なHTML形式で、院外向けウェブサーバに対して更新される。

④空床数の表示

サーバにはあらかじめユーザ名とパスワードを設定してあり、利用通知を受けた登録医にのみ空床数が閲覧できるアクセス制限を行っている(.htaccess)。



20% / 65%

3.結果

I. オーダエントリシステムとの DDP 通信

当初は入退院管理システム側から引数渡しリモート Do にて I/F を取る仕様であったが、実装が不完全であったので、イントラネット M サーバ側に対して連携ベッド入退床グローバルの DDP 書出しを行い、バックグラウンド Job にて書きこみがあつた時に取りこむ仕様に変更した。

II. 追加ハード・ソフト

オーダエントリシステムおよび既存のイントラネット・インターネットシステムの利用により追加購入したものはなかった。

III. テスト運用でのトラブル

転棟・転室後の退床登録が更新されないトラブルがあつたが、プログラムの変更と運用・操作マニュアルの改変により、トラブルは解消できた。

IV. アクセス制限

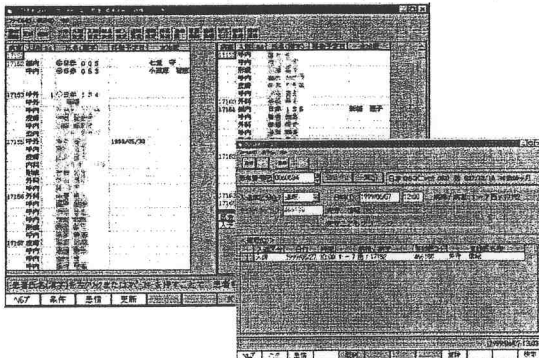
ウェブサーバのアクセス制限は正常動作が確認できた。

V. セキュリティ

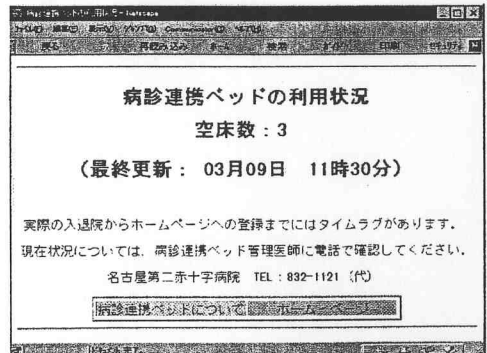
ウェブサーバへの空床数 HTML ファイル転送はファイアウォールを越えて行うので、院外向けウェブサーバに対して直接 FTP を行うのではなく、ファイアウォール経由の NFS を用いセキュリティに十分に配慮した。

4.画面構成

入床・退床登録画面(オーダエントリシステム)



空床数表示画面(インターネット)



5.まとめ

本システムの空床数は病棟で行われる入退床登録のタイミングで更新されるため、登録医から当院病診連携ベッド管理責任医師への利用申込の電話連絡を受けた時刻との間にタイムラグが発生するという制約があるものの、今後要求が高まると予想されるインターネットを利用した病院情報の提供に対する1つの回答になり得ると考える。

遠隔診断支援システムの構築

○本多正幸¹⁾、 鈴木隆弘¹⁾、 高林克日己¹⁾、 津本優子¹⁾、 里村洋一¹⁾
植松祐美子²⁾、 姜琳³⁾

- 1) 千葉大学医学部附属病院 医療情報部
- 2) " 看護部
- 3) 住友電工システムズ(株)

〒260-8677 千葉市中央区亥鼻 1-8-1
TEL : 043-226-2346 FAX : 043-226-2373
honda@ho.chiba-u.ac.jp

1. はじめに

千葉大学病院では、平成8年(1996年)より、system-CHIBA IIとして総合病院情報システムを稼動している。医事会計システムは富士通(株)のパッケージシステムにより運用しており、オーダーエントリなどの診療支援関係業務は、住友電工システムズの Accel を利用している。本院における患者 DB (患者基本情報、病名情報、検査結果データ、看護情報、各種オーダー情報、等々) は、M 言語により構築されており、患者基本情報や検査結果データなどは 1981 年から長期にわたり保存され、診療および研究に利用されている。

各種オーダーシステムの稼動を推進し、業務のシステム化の拡大を計っているところであるが、端末のレスポンス、院内ネットワークの脆弱性、サーバーマシンのパフォーマンスなどが問題として生じてきた。端末のレスポンスの高速化に関しては CPU の交換などを行いマイナーには対応作業を行ってきたものの、本質的な意味での改善にはなっていなかった。そこで、「遠隔診断支援システム」と銘打った昨年度の補正予算を契機にして、院内のネットワークの整備、サーバーマシンの強化を行った。端末についても数百台の単位での高速化(ハイパフォーマンス化)を実現した。これにより一応、今後の院内画像情報の流通、遠隔診断システムや地域医療連携の普及に伴う情報量の増大化に対応するためのインフラストラクチャーが整ったと考えている。

本院のネットワークは大きく分けて、診療支援系ネットワークと画像系ネットワークがあるが、これまで2台の ATM をその中核においてネットワークが構築されていた。今回の整備により、ギガビットイーサの交換機を2台導入し、これにより診療支援系のネットワークを構築した。2台の ATM は画像系ネットワークの中核として位置付けた。診療支援系ネットワークにおいてはレイヤ3スイッチ等を配し、端末からは 100Mbps の転送速度を実現し、今後頻度が高まるであろう画像データの検索にも対応できるように環境を整えた。もちろんオーダーエントリーなどの日常業務の高速レスポンス化にも対応した。画像系ネットワークにおいては、主に各種モダリティから発生する画像情報の収集を行える環境が整った。(参照 図1)

院内情報ネットワーク概念図 GigaBit II

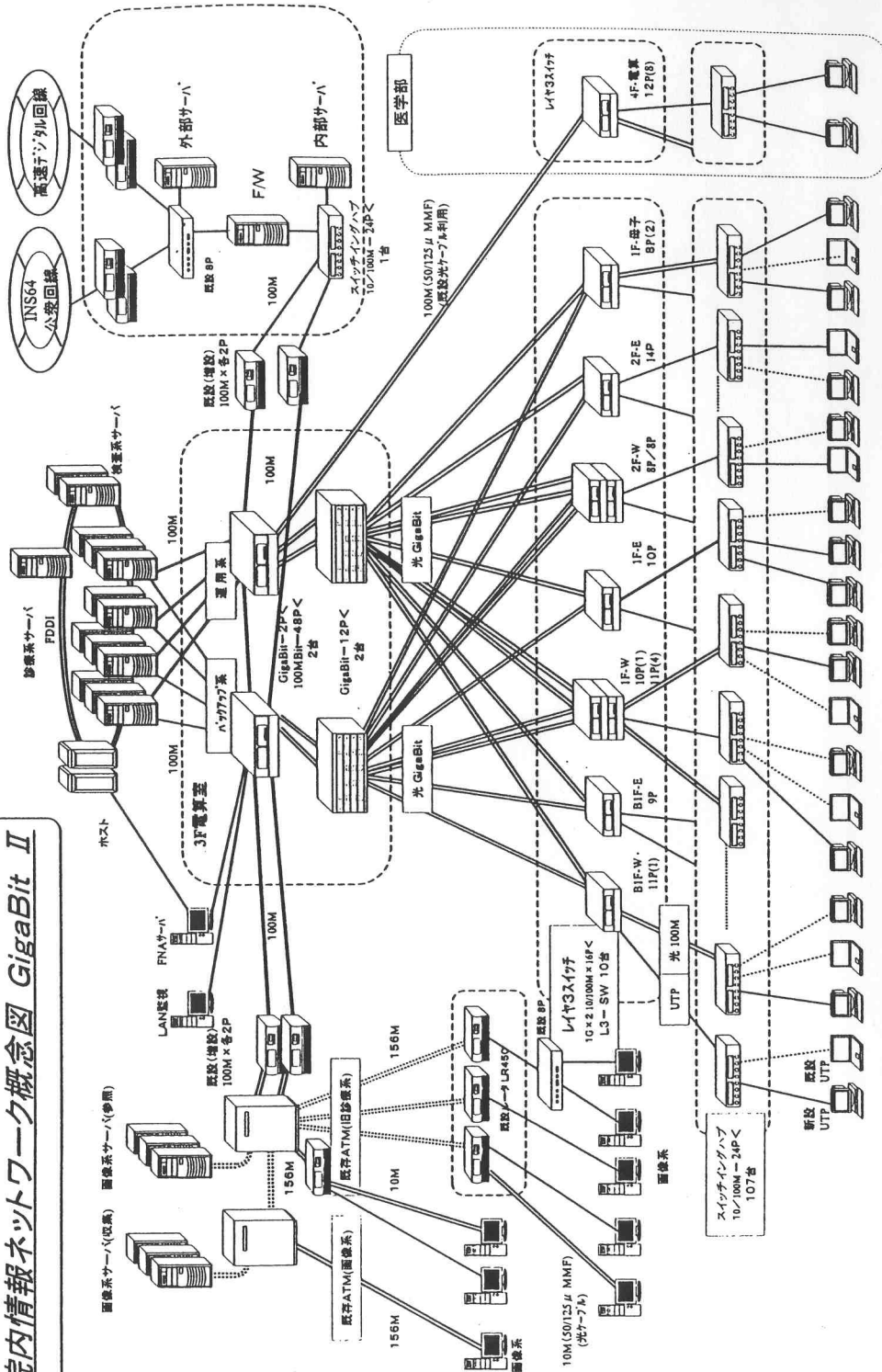


図1 ネットワーク概念図

・既存ネットワークを有効利用した帯域の拡張
 既存ハードの一部置き換えによる高速化

2. 遠隔診断支援システムの概要

遠隔診断支援システム (DxMM TeleMed、メダシス・ジャパン(株)) は、静止画、動画、白黒、カラーの画像の転送と共有、複数画像の同時表示と共有、DICOM 機器やスキャナーからの画像取り込み、TV 会議をしながらの遠隔画像診断等を行うことができるものである。さらに、一度遠隔病院へ転送し共有した画像は、パラメータ情報のみを送信することによって、部分拡大機能とその共有、ウインドウレベル共有、テレカーソルの共有が可能なのが本システムの特長となっていて、再度変更処理をした画像をそのまま送る必要がない。

具体的な連携医療機関の選定にあたっては、各診療科からのアンケートを下に、千葉県内 5 病院 (千葉県救急医療センター、君津中央病院、安房医師会、千葉県こども病院、成田赤十字病院)、栃木県内 1 病院 (塩谷総合病院) を選び、本院とは INS を経由して接続が行われる。今回は、システムのセット (ハードとソフト) の台数の制約があり 6 箇所としたが、今後連携病院を増やしていく予定である。本院から連携医療機関に情報が提供される場合は、院内で蓄積されている DICOM サーバー上の画像情報 (CT、MRI 等) を中心に連携医療機関にそれらの情報が INS 経由で転送され、情報の共有が行われる。また、TV 会議システムを通じた意見交換も同じタイミングで可能となっている。(参照 図 2～3)

3. システムの評価

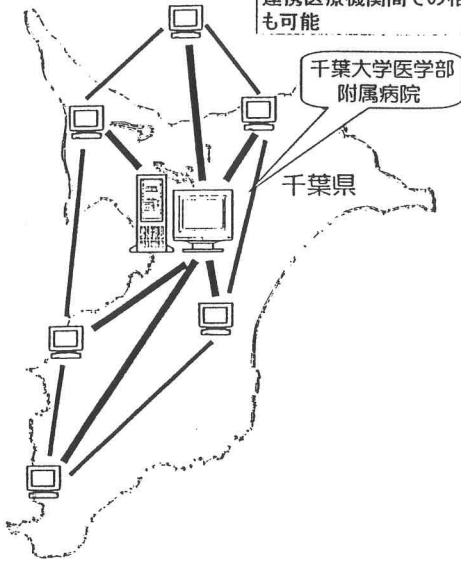
本院採用の遠隔診断支援システムは、これまで本院を皮切りに複数の病院で採用されているようであるが、これまでの仕様経験を踏まえて感想を述べてみる。まず第一に転送スピードの問題である。INS64 を 2 契約、つまり B チャンネル 4 本を束ねて 256KB での運用を推奨しているが、ルータの価格の問題もあり、過半数の病院では INS64 を 1 契約での運用となっている。千葉県こども病院を対象に 6 月 30 日に開通記念デモンストレーションが行われたが、あらかじめ共有する画像を転送しておき、デモの本番では部分拡大機能、ウインドウレベル共有、テレカーソル共有などにより問題なく行われた。しかし、リアルタイムの遠隔会議、遠隔カンファなどでは画像転送時間の問題は大きい。第二の問題は、第一の問題とも関係するが、音声途中でプツプツと切れて聞きにくいということである。本院では、ISDN のルータから院内回線を通して通信を行っていることにも原因があるようである。(名古屋大学では、直接 INS と結んでおり、その場合は本院で経験するほどの悪さではなかった。) この音声の問題については普通のアナログの電話回線を併用したほうがよいのではとの意見も出されており検討中である。第 3 の問題としては、2 地点の相互通信という仕様になっているが、3 地点以上の間での通信を行いたいという要望が出されている。TV 電話は別にして画像の共有という面では、近々多地点間での共有も可能になるということである。

今後、本院における院内画像情報の充実を図り、実際に遠隔診断支援システムの運用経験を重ねて、より使いやすいシステム、導入効果が期待できるシステムの構築を目指していきたい。

千葉大学医学部附属病院 遠隔診断支援システム

大学病院と各連携医療機関間は
ISDN(INS128) × 2にて接続
大学病院との遠隔診断だけでなく
連携医療機関間での相互遠隔診断
も可能

—連携医療機関側 基本システム—
(千葉大学医学部より貸し出し)



デジタルカメラ、デスクトップスキャナーから読み込まれた画像はDICOM
フォーマットで保存され、大学病院または各連携医療機関に送信されます

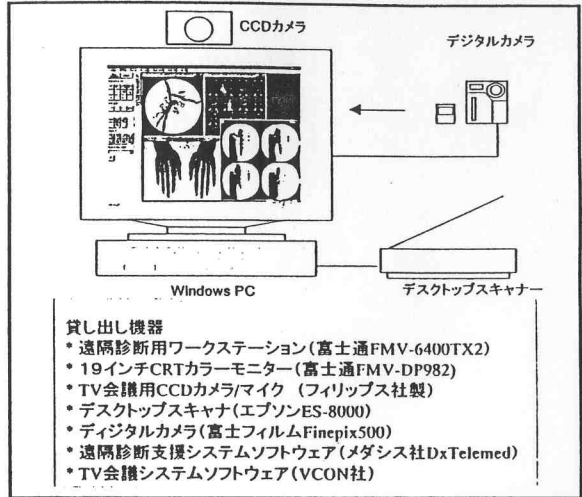


図2 遠隔診断支援システム 解説

千葉大学医学部附属病院 遠隔診断支援システム 大学病院内 システム構成図

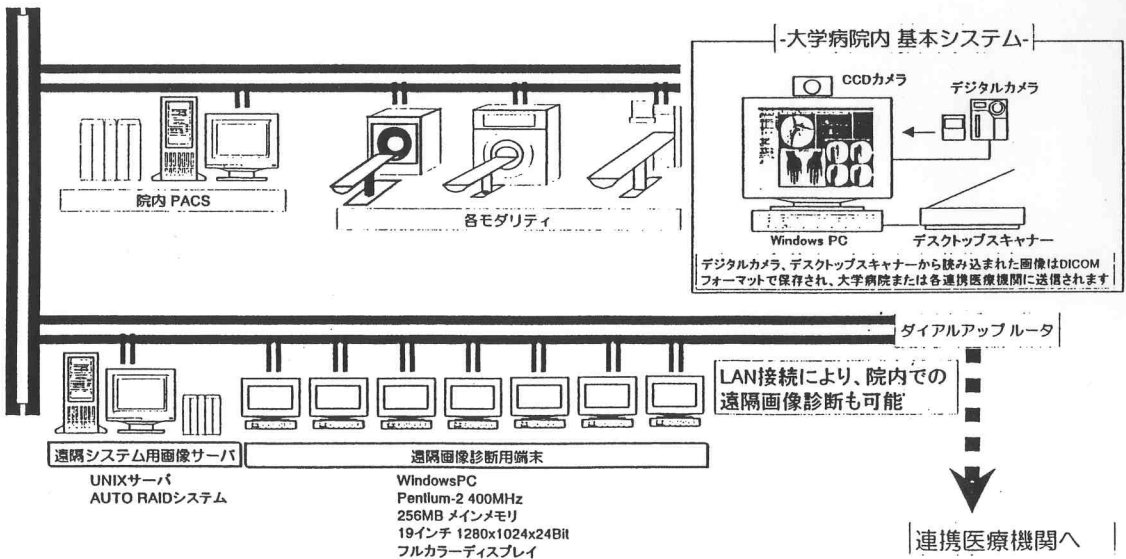


図3 遠隔診断支援システム 構成概念

Cache'-Web 連携機能を用いた 「地域医療情報ネットワークシステム」の開発について

○向井まさみ²⁾、山本 和子¹⁾、中島 裕生²⁾

1) 鳥根医科大学医療情報学講座

2) ニチメンデータシステム株式会社技術部

〒111-8520 東京都台東区柳橋2-19 秀和柳橋ビル

TEL: 03-3864-7699 FAX: 03-3864-7566

masami@nichimen-nds.co.jp

1. はじめに

“鳥根地域医療情報ネットワークシステム”は、地域医療機関間での患者情報の共有・相互利用を促進させ、入院・退院から社会復帰までのケア、在宅医療の充実等、地域医療の組織化を目的とし開発されたシステムである。

複数の医療機関（利用者）からのアクセスを受けるサーバクライアント間システムの構築という条件をふまえ、実現手段として、クライアント側には GUI として Web ブラウザを採用し、サーバ側を Web サーバを経由して各種データ(DB)にアクセスを行うシステム構成とした。

本稿では、このような構成を元に、鳥根地域医療情報ネットワークシステムのアプリケーション機能を、主に Cache の Weblink 機能を用いて開発したので報告する。

2. 地域医療情報ネットワークシステムの概要

地域医療情報ネットワークシステムのマシン構成およびサーバ役割を図1に示す。

本システムのアプリケーションは大きく分けて、以下の機能をもつ。

- ・ 文書情報交換機能
 - … 紹介状、情報提供依頼書、画像診断報告書等の各種依頼状の作成／参照機能と、その返信用文書作成・参照機能
- ・ DICOM 画像取得／転送機能
 - … DICOM 画像データの受信・転送・参照機能。受信は DICOM プロトコルで、鳥根医科大学内放射線部画像サーバとの通信を行う。
- ・ 病歴情報提供／参照機能
 - … 患者病歴情報の取得・提供・参照機能。鳥根医科大学内情報参照システムからの情報取得手段として HL7 を使用する。
- ・ 症例検索機能
 - … 症例情報の検索・参照機能。病名等をキーに用い、格納されている病歴情報を参照する。DICOM 画像の場合も撮影条件等で検索/参照する。
- ・ その他、システム管理機能 … 利用者の権限コントロール、マスタ登録機能

3. Weblink 機能の特徴とシステム実装時の考慮点

Cache-Weblink 機能の特徴は、Web サーバと DB サーバ間のセッションを保持する点であるといえよう。

開発者からみると、通常は hidden 変数や cookie を使用してページ間で受け渡す動的な情報を Weblink が保存/管理してくれるため、通常の関数間で引き渡す変数の感覚と同等に扱えるなどメリットがある。

反面、Web に表示する 1 ページ分の処理を 1 関数と同様に扱うため、目的ページへのアクセスを、どこからでもダイナミックに行うことは難しくなっている。つまり処理は必ず START ページとして定義したページから開始するように仕組みなければならない。また、セッションを有効に管理するためセッションのタイムアウト等の制限を設けざるを得なく、長時間アクセスがない場合は、セッションが切れてしまい、それまでの作業は保存されるが、再度開始ページからのセッションをはり直す必要がある。

4. 考察

開発当初採用していた Weblink 機能では、ブラウザの Back (戻る) ボタンと同期をとることができないなどの事象が認められたが、最終的に採用した Weblink では多くの点が改善されている。Web アプリケーションの実現手段として、優れている点と、まだ改善を要する機能があり、より一層のブラッシュアップを望むものである。

システムとしては、Web サーバを経由した構成にしているため、画像データのように他の Web インタフェースをもちサーバとの連携を取ることも可能となり、より拡張性のあるシステムとなっていると考える。

参考文献

- [1] 山本和子、牧野純、柳原真佐美：地域医療情報ネットワークシステムの構築について：第18回因幡情報学会大会論文集、1998
- [2] 中島、杉本、向井、鈴木：オブジェクト・リレーショナルモデルを採用した DICOM 準拠の医療画像データベース管理システム：第18回因幡情報学会大会論文集、1998

以上

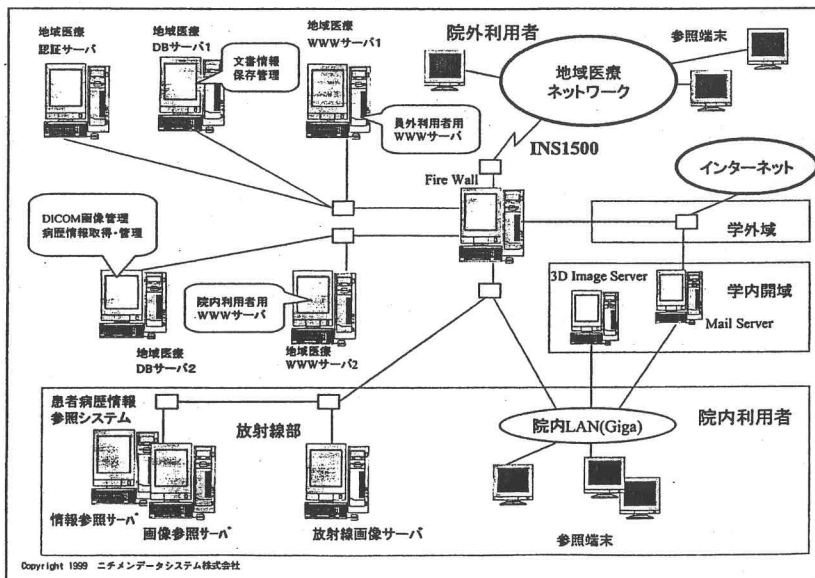


図1. 鳥根地域医療情報ネットワークシステム構成

* Tel developer * Weblink-TP
 * Weblink → SMTP → X-TC 4-2
 (%wlsmtpt?)

M言語の西暦 2000 年問題

○鈴木利明* 嶋 芳成*

*日本ダイナシステム株式会社

〒460-0007 愛知県名古屋市中区新栄 2 丁目 1-9 雲竜ビル東館 5F

Tel: 052-242-5441 Fax: 052-242-5984

E-mail: contact@jdynasys.co.jp

1. 西暦 2000 年問題とは

西暦 2000 年をまたがる日付データによってコンピュータシステムが期待しない動作をすることである。
(以下 Y2K 問題)

この問題の特徴をリストにする。

- A. 期限: 期限が決まっており延ばせない。
- B. 同時多発性: 世界中でこの問題が同時に起きる。
自己のシステムだけではない。
- C. 影響範囲: 日付の持つ汎用的な性格のため、ハードウェア、ファームウェア、OS、アプリケーション、組み込み装置など広範囲である。
M 言語装備、M アプリケーションだけに限らないすべての CPU を含むシステムに関係している。
- D. 検査範囲: 利用している検査機器、システム、ソフトウェアすべてに検査が必要である。
外部調達しているものも含める。
- D. 修正: 修正対象アプリケーションプログラムはほとんどのプログラムにわたる。
業務知識は不要である。処理ロジックは単純である。修正個所の発見、分析、修正検査にコストが掛かる。外部調達システムの場合対策不可能な場合もある。
- E. 被害規模: 被害額が膨大と予想される。長期にわたる可能性がある。
対策を怠った場合、基幹業務の停止も有り得る。

これら理由により事前対策が重要である。

西暦 2000 年問題対応リスト

PC 等ハードウェアの西暦 2000 年対応の試験

自社ソフトウェアの西暦 2000 年対応 (西暦 2000 年 2 月 29 日の動作確認)

他社ソフトウェアの西暦 2000 年対応の確認と対応

他社ハードウェアの 2000 年対応の確認と対策

他社検査機器等、ハードウェア機器、通信、ファイル等の西暦 2000 年対応

西暦 2000 年直前対策

オペレータ、利用者に対する事前教育 (現象の早期発見、連絡網)

西暦 2000 年正月の運用確認

データの確認

監査のやり方

SPSS
PLC
損益計算書

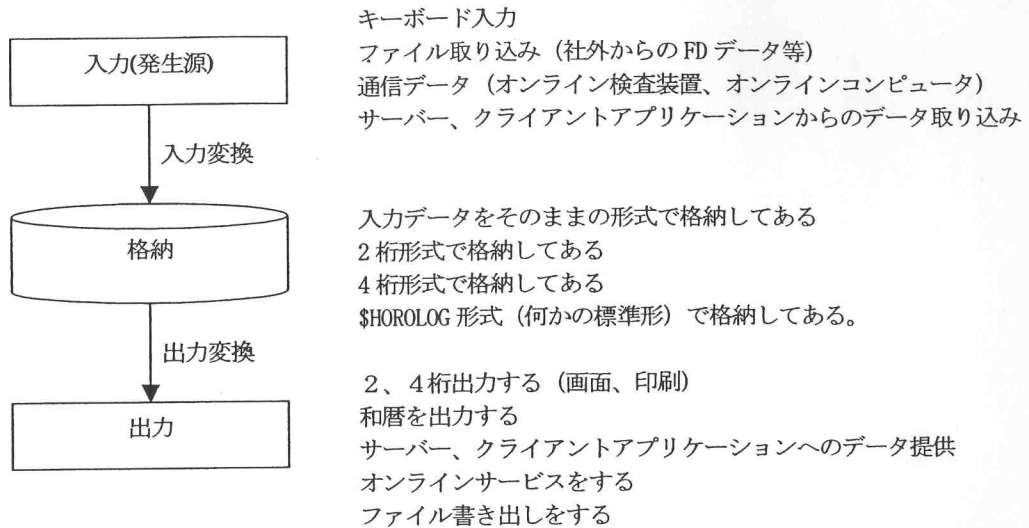
西暦 2000 年問題の分類

問題を起こす場所の分類

起こす場所は、4つに分けることができる。入力、格納、出力、論理計算である。

M 言語アプリケーションでは、入力、格納、出力、論理計算を別の問題として扱うことが多い。

入力および出力の自由度を高めるためにおよびアプリケーションの開発量を押さえるため、入力変換、出力変換をしている。



論理計算

前後の判定
期間の計算
日付による順序 (ソート)

しかしながら、M 言語ならば Y2K 問題を起こさないわけではない。問題を起こすのは西暦年号が 4 桁以外の桁数または固定桁数である。

入力は必ず西暦を 4 桁入力するシステムならば、入力場所で 2000 年問題はおきない。

しかしながら、必ず 4 桁入力することをユーザーが納得しないことも有り得る。また画面サイズの関係で 4 桁にできないこともある。入力エリアが 2 桁しかないときには、今後日付データは 4 桁に拡張しなければならない。2 桁入力した年号値が YEAR に代入されているとき、

SET YEAR=1900+YEAR または SET YEAR=19_YEAR

で拡張するシステムでは、2000 年を 1900 年として認識する (西暦 2000 年問題の発生)。

M 言語では、\$HOROLOG 形式で格納することが多い (これは Y2K 問題を起こさない)。二つの \$H を引き算すると経過日数を算出することができる格納形式である。また多くのユーティリティが存在している。しかし、それ以外の格納形式を取っているアプリケーションもある。年号を 2 桁で表わす形式ではオーバーフローを起こすので対策が必要となる。

出力は用途によって対策が変わる。印刷物では年号の桁数が変わるとコラム位置が変わり使い物にならないことがある。現在 2 桁出力のものは、内部形式が 4 桁になろうとも出力 2 桁を維持しなければならない。また同様に、通信プロトコル、ファイルでは受け取る他システムとの話し合いにあわせて設定を維持する必要がある。受け手が人間ならば 2 桁でも 4 桁でも解釈してくれるので致命的な現象は起きない。

アプリケーションの修正の基本的な対応

A. 2桁対応型

年号データを2桁のままにして対処する方法。年号データが変更できない場合に使用する。データファイルや検査装置からのデータが2桁で電送してくるような場合に使用する。プログラムだけの変更をする。一定期間後に再度手直しが必要である。随時変更ができる。具体的には、50～99は、1950～1999年、00～49の間は、2000～2049年とみなす。

(範囲は装備者によって異なる)

Caché の日付変換ユーティリティのウインドオプションを利用することも有効である。

B. 4桁対応型

年号データをすべて4桁で扱う方法。修正プログラムは9999年まで利用できる。プログラムとデータの一括変更が必要である。

C. 変則型

多くの場合、2桁対応の変形が多い。16進数表示に変えたり、数字の変わりにスペース文字を使用するなど。アプリケーション側での対応が個別に必要である。

計測、検査機器などにこの対応方法をとるものが多い。(末尾 Web サイト参照)

どの方式をとるにしても変更するならば、(\$HOROLOG 形式のような)標準の格納方式を決めて、入力変換出力変換と組み合わせるのがよい。この方式ならば変換量も少ないし、自由度も高い。

場所の発見方法

プログラムを調べる方法

入力、出力	\$H, \$HOROLOG、日付(変換)関数、日付ユーティリティ、キーボード入力、ファイル取り込み、通信取り込みプログラムをキーワードにして調べる。
日付計算	年号に使っているルーチン名、ラベル、変数名が分かっているときには、YEAR, Yなどをキーワードにして調べる。
格納	格納グローバル名が分かっているときには、それをキーワードにして調べる。格納する専用のルーチンを作成し、すべての格納はそれを経由する方法に変える方法もある。
ユーティリティ	自前の日付ユーティリティを用意している利用者は多い。そのユーティリティから調べること。

グローバル・データ中の西暦 2000 年問題

グローバル変数の中にも日付情報は格納されている。

格納方式はこの用途でいろいろある。各方法毎に西暦 2000 年に対して桁あふれ、論理計算ができるかを検証しておく必要がある。

検証

修正をした個所は、必ず動作検証をして、新たなバグを入れていないか、正しく修正されたかを検証しなければならない。

PC 環境における西暦 2000 年問題

ハードウェア

1996 年以前のハードウェアでは、Y2K 問題に対応していないものがある。

ハードウェアには、日付を管理するために常時電池駆動のカレンダークロックが動作している。

PC の電源を入れていなくてもこのカレンダークロックは動作している。しかし、古い PC では、西暦の下二桁のみカウントアップの対象としているものや、まったく西暦 2000 年を考慮していないものがある。

このタイプの PC では、1999 年 12 月 31 日から 2000 年 1 月 1 日に移行しない。

そのため、現在使用している PC が 2000 年 1 月 1 日にどう振る舞いをするかを調べ、管理する必要がある。このカレンダークロックの値を基準にして M 言語の \$HOROLOG の値を算出している。この値が狂うと正しい日付とならない。

対応 A. 西暦 2000 年に対応しない古い PC は破棄してしまう。

対応 B. DOS ならば、PC-DOS 2000 (IBM 社製) に変える。これは PC-DOS 7.0J の改定版でハードウェアが対応していなくても DOS レベルでカレンダークロックを補正する。

対応 C. 24 時間動作させていない PC ならば、2000 年の仕事はじめの日に、まず、日付を手動で再設定をする。ユーザー、管理者への教育を徹底する必要がある。

電池

カレンダークロックの電池の寿命は 4 年程度である。このため古い PC ではカレンダーを正常に維持できないものも存在する。電池を適当なものに取り替える必要がある。

OS

PC の動作環境である DOS, Windows95/98/NT, MacOS の西暦 2000 年対応版をインストールしておく必要がある。また情報を収集することも重要な仕事となる。

組み込み機器の西暦 2000 年問題

コンピュータが組み込まれている機器は PC だけではない。各種検査装置、ネットワーク装置、制御装置などにも含まれている。システムはこれらと通信をしながら動作しているものがほとんどである。検査機器等を使ってデータを取り込んでいる場合には、そのデータフォーマットがどう影響を受けるかを確認、対応を取っておかなければならない。これらの装置での西暦 2000 年問題がシステム全体に影響を及ぼすことがある。組み込み機器の特徴は、

1. メーカーが別々である。
2. 動作検証がし難い環境にあることが多い。(日付を変更できないことがある)
3. 問題が存在した場合に、メーカーで対応できる場合と、出来ない場合もある。対処方法が特殊な場合がある。
4. プログラムが ROM で固定しており変更できない場合もある。このような場合には、システム側が対処するしかない。

さらに、システムに接続しない機器でも、日付を利用している機器では西暦 2000 年に日付を設定し直したりしなければならぬ機器があると思われる(可能性)。これら機器でも 2000 年に利用を続けられるかどうかの確認が必要である。

DTM における西暦 2000 年問題

DTM の西暦 2000 年対応は V4.10 と V6.6 で行なう。それ以前の版ではジャーナルファイル名に障害がある。

障害を起すことのないユーティリティ

```
$HOROLOG, $$`%now, $$`%xdate($H, 1), $$`%xdate($H, 2), $$`%xdate($H, 3)
$$`%mdate, $$`%mdate($H, 0), $$`%mdate($H, 2), $$in`%date("T")
```

西暦 2000 年以降になると 2 桁が 4 桁に変わる日付ユーティリティ

```
$ZDATE, $$`%date, $$`%date($H), $$out`%date($H)
`%D, $$`%xdate, $$`%xdate($H, 0)
$$`%datecheck("T") → 58073_991231_12/31/99_Dec 31, 1999 (12/31/1999)
($c(22)→_) → 58074_20000101_1/1/2000_Jan 1, 2000 (1/1/2000)
```

戻り値は、2000 年になると桁数が 2 桁から 4 桁に変わる。この日付ユーティリティを使用したプログラムを出力で使用したプログラムでは、桁ずれを起す。また、日付の西暦年を取り出すとき、下記のようなプログラムでは正しい西暦年が取得できない。

```
例: set year=19_$p($ZDATE, "/", 3)
set year=19_$e($$`%date, 7, 8)
set year=1900+$E($p($$`%datecheck("T"), $c(22), 2), 1, 2)
set year=$p($$`%date("1/1/2002"), "/", 3)-$p($$`%date("1/1/1999"), "/", 3)
```

西暦 2000 年以降になっても 2 桁のままの日付ユーティリティ

```
$$`%DATE($H), $$`%mdate($H, 1)
この日付ユーティリティは出力仕様しても桁ずれはおきない。
しかし年数計算で使用すると誤計算をする。
```

```
誤例: set year=$p($$`%DATE(58074), "/", 3)-$p($$`%date(58073), "/", 3)
```

日付入力ユーティリティでは、2 種類の問題点を分けて考える必要がある。

A. ユーティリティに与える日付データ B. 戻り値

\$\$in`%date は、\$H 形式の値を返す。

修正の必要がない引数は、“12/31/1999”、“1/1/2000”、“2/29/2000”、“1/1/2001”

\$\$in`%date("12/31/99")	58073	12/31/1999 の\$H	修正必要
\$\$in`%date("1/1/00")	21550	1/1/1900 の\$H	修正必要
\$\$in`%date("2/29/00")	空文字列		使用不可
\$\$in`%date("1/1/01")	21915	1/1/1901 の\$H	修正必要
\$\$in`%date("1/1/0")	21550	1/1/1900 の\$H	修正必要

\$\$`%datecheck は、日付情報を複数の形式にして返す。\$c(22) は、_ で表現している。

入力	戻り値	コメント
\$\$`%datecheck("12/31/1999")	58073_991231_12/31/99_Dec 31, 1999A	
\$\$`%datecheck("1/1/2000")	58074_20000101_1/1/2000_Jan 1, 2000	A
\$\$`%datecheck("12/31/99")	58073_991231_12/31/99_Dec 31, 1999B	
\$\$`%datecheck("1/1/00")	21550_00101_1/1/0_Jan 1, 1900	B
\$\$`%datecheck("1/1/0")	0 不正入力扱い	C

A: 戻り値を\$c(22)で区切ったときの第1、第4ピースの使用は修正の必要はない。

第2、3ピースは、2000年以降は、2桁から4桁に桁数が変わるので戻り値の扱いに修正が必要である。

B: 入力日付データに2桁入力をするとき1900年代と認識する。

必ず4桁入力をするように変更しなければならない。

このため、戻り値は使用できないものとなる。

C: 年数に1桁を使用すると不正入力扱いになります。1桁入力にはできないようにしなければならない。

Caché における西暦 2000 年問題

InterSystems 社は Caché version 2.1 以降で確認している。しかし、すべてのユーザーアプリケーションの西暦 2000 年対応を保証するものではない。

日付変換ユーティリティ

\$ZDATE, \$ZD	引数(\$HOROLOG 形式)を日付形式に変換する
\$ZDATEH, \$ZDH	引数(日付形式)を\$HOROLOG 形式に変換する
\$ZDATETIME, \$ZDT	引数(\$HOROLOG 形式)を日付時刻形式に変換する
\$ZDATETIMEH, \$ZDTH	引数(日付時刻形式)を\$HOROLOG 形式に変換する

第二引数 (形式)	例	年号桁
0 DD Mnn {YY}YY	01 Jul 99	可変
1 MM/DD/{YY}YY	11/02/54	可変
2 DD Mnn {YY}YY	02 Nov 54	可変
3 YYYY-MM-DD	1954-11-02	4
4 DD/MM/{YY}YY	DD/MM/YYYY	可変
5 Mnn D, YYYY	Nov 2, 1954	4
6 Mnn D YYYY	Nov 2 1954	4
7 Mnn DD {YY}YY	Nov 02 54	可変
8 YYYYMMDD	19541102	4
9 Mnnnnn D, YYYY	July 1, 1997	4

これら関数にはその他にも複数の表示オプション、入力オプションが用意されている。

3, 5, 6, 8, 9 の形式の年号は常に 4 桁であるが、他の形式では 2 桁または 4 桁と変わる。

このため年号の桁数が変わるオプションを使ったアプリケーションでは西暦 2000 年に障害を起こすことがある。アプリケーションの変更が必要である。

特に西暦 2000 年問題に対応するために、年号の 2 桁ウィドウを指定するオプションを用意してある。

例：

1999 年 12 月 31 日での \$ZDATE(\$H) は、12/31/99 を返す。

2000 年 1 月 1 日での \$ZDATE(\$H) は、01/01/2000 を返す。

1999 年 12 月 31 日での \$ZDATE(\$H, 1, , 5, 40, 30) は、12/31/99 を返す。

2000 年 1 月 1 日での \$ZDATE(\$H, 1, , 5, 40, 30) は、01/01/00 を返す。

この例では、\$ZDATE(\$H) では、1999 年と 2000 年では年号の桁数が変わる。しかし、

\$ZDATE(\$H, 1, , 5, 40, 30) ならば、常に 2 桁にすることができる。

引数の意味は、第一引数の値を、今日を基準として 40 年前から 30 年後までの期間(ウィンド)は年号を 2 桁で取り扱うオプションである。その他に多数のオプションが用意されている。

日付ユーティリティ

%D	現在の日付を表示する
%DATE	日付を内部形式に変換する
%DO	内部日付を外部日付を表示する
%TIME	現在の時刻、曜日、日付を表示する

西暦 2000 年問題直前対策

事前の確認

冬場の流感と同様な対策が必要である。

	流感	西暦2000年問題
予防	体力増強	事前修正
直前対策	-	バックアップ
症状	発熱	ハード、入力、ソート等
隔離	人	アプリモジュール、DB
連絡先	病院	ソフト会社、作成者
早期発見	自覚症状	ユーザーの教育
早期治療	薬の投与	パッチ
長期治療	体力増強、免疫	4 桁化

まず、予防である。

風邪をひいたときにでる症状を事前に知っているのは大変重要である。ハードウェア、システムソフト、アプリケーションソフトなど項目別にリストの作成をお勧めする。

使用している機器の一覧表、各連絡先や、アプリケーションでは要員の連絡先、いつ連絡を取ることが出来るかを事前に調べとくと対処が早くなる。

問題をできるだけ小規模で押さえるためには早期発見、早期治療が重要である。そのためにはユーザーに対する教育、指導の徹底をする必要がある。

データ保全

壊れたデータベースからデータを正常に戻すのは困難である。このため壊れる前のデータ、データベースのバックアップを作成するのが一般的である。バックアップ装置、メディアを再検討してみる。各 M 言語装備にはバックアッププログラムが用意されている。

バックアップするデータベースサイズ、バックアップ容量、バックアップ時間を把握しなければならない。

1999 年 12 月 31 日現在のデータベースのバックアップ

西暦 2000 年対応を完全にしておいたとしても、大晦日にバックアップを作成するのは良い考えである。万が一にもデータベースが壊れたときには、頼みの綱になるのはこれだけであるからである。

縮態運転の用意

一部の機器に誤動作を発見したときには、その機器を切り離してシステム稼働を続ける必要がある。障害の程度にあわせて縮態運転の用意をしておくのがよい。

サービス停止に対する対策

致命的な障害が起きたときには、システムは停止せざるを得ない。そのときの対策を用意しておく。

年末年始スケジュール

1999年12月23日	木曜日	天皇誕生日
1999年12月24日	金曜日	
1999年12月25日	土曜日	
1999年12月26日	日曜日	
1999年12月27日	月曜日	
1999年12月28日	火曜日	
1999年12月29日	水曜日	
1999年12月30日	木曜日	
1999年12月31日	金曜日	
2000年1月1日	土曜日	
2000年1月2日	日曜日	
2000年1月3日	月曜日	
2000年1月4日	火曜日	仕事始め
2000年1月5日	水曜日	
2000年1月6日	木曜日	
2000年1月7日	金曜日	
2000年1月8日	土曜日	
2000年1月9日	日曜日	
2000年1月10日	月曜日	

一般企業ならば2000年1月の社外に対しての仕事開始日を遅くすることもひとつの方法である。棚卸しのようにシステムの社内検査をして、確認をとってから営業を始める。

西暦2000年問題サバイバルガイド

インフラ環境のサービス停止

西暦2000年問題は、自社アプリケーションの障害だけでなく、社会活動の活動基盤をなす電気、ガス、水道、下水、交通網、通信網の停止も可能性が残っている。交通機関などでは安全の確認されるまで利用を避けるのが得策である。電話では、停止はなくても輻輳を起こす可能性があるため複数のアクセス方法を確保するのがよい。トラックなどの輸送手段は被害を受けなくても各社の情報システムは影響を受けて物によっては流通量が少なくなることもありえる。

参考サイト

InterSystems 社 www.intersys.com

医療用具製造、輸入販売業者、外国製造承認業者を対象とした調査

www.mhw.go.jp/topics/c2000/tp0420-1_4.html

岩波ブックレット No. 451 コンピュータ 2000年問題 松井一郎著 ISBN4-00-003391-3 400円

M Web-Linkによる授業出席管理システムの開発

大櫛陽一、岡田好一、春木康男

東海大学医学部医用工学情報系

〒259-1193 神奈川県伊勢原市望星台

TEL:0463-93-1121 ext. 2140 FAX:0463-96-4301

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

1. はじめに

WWW ブラウザを GUI としたシステムは、ユーザ操作性の良さ、広域ネットワークシステムの構築性、保守のサーバ集中による効率化などの理由により広がりつつある。一方、CGI、処理プログラム、データベースなど他種類の製品を組み合わせる必要があり、開発やデバッグの難しさがあつた。今回、Cache の WebLink を使った授業出席管理システムを開発し、評価を行ったので報告する。

2. システムの開発目的

当教室で受け持つ授業は医学部の1年の必須2科目、3年～6年の選択3科目、修士課程の選択2科目、博士課程の選択2科目、健康科学部の看護学科の必須2科目、社会福祉学科の必須1科目、修士課程の必須1科目とあり、年間361時間あり、今後さらに増加が予定されている。授業の準備としてのテキスト作成、実習環境の整備、実習データの収集は大変な仕事であるが、その後の成績管理も大変である。出席管理、課題提出チェック、ペーパーテストの問題作成と採点などもかなりの作業量となる。出席管理は、成績を付けるためだけではなく、特に1年生では出席率向上や、問題学生の早期把握にも重要である。

(1) 現在は、次のような出席管理システムを開発して使用している。

- ・サーバ UNIX U-MUMPS
- ・クライアント Telnet 磁気カード・リーダー(キーボード割込)

このシステムでは、学生証が磁気化されており、ここに書かれた学生証番号を読みとってサーバに登録している。このシステムはすでに5年間使用されている。しかし、磁気カードリーダーの台数が限られているため、授業が連続する場合に学生の行列が出来るなどの問題が発生している。

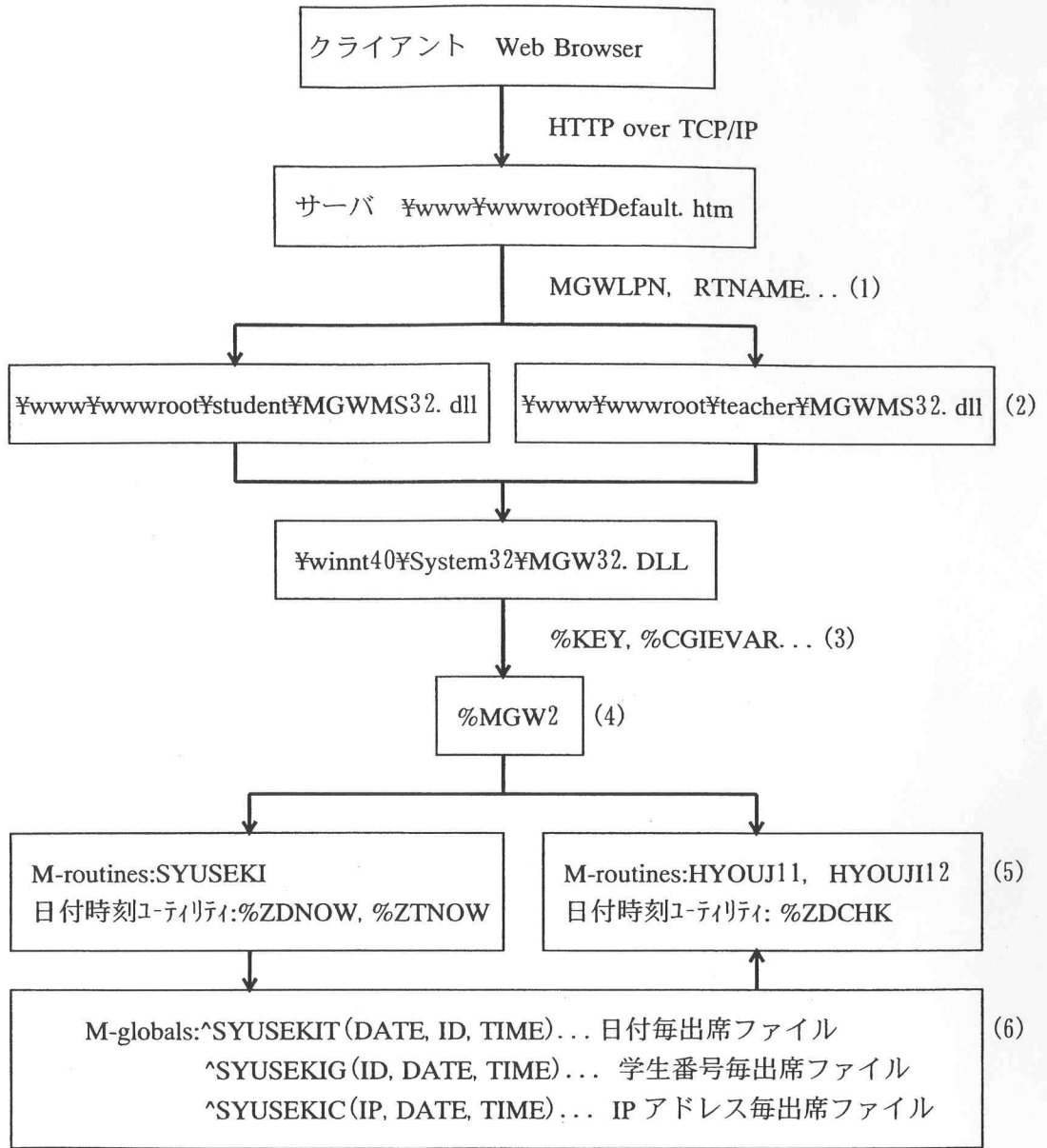
(2) 新システム

- ・サーバ WindowsNT Cache
- ・クライアント WWW ブラウザ

学生自身が自主的に出席登録をすること、全端末から登録出来るようにすること、他の先生での授業でも使えるようにすること、出席状態の確認を簡単化することなどが再開発の目的である。

3. WebLink とアプリケーションの構成

次に、WWW ブラウザからの要求に対応したアプリケーションの動きを示す。



特徴的な処理に付いて次に解説をする。

(1) MGWLPN は、WebLink に登録したコンフィギュレーションから一つを選択する。これは WebLink にリザーブされたキーで必須である。

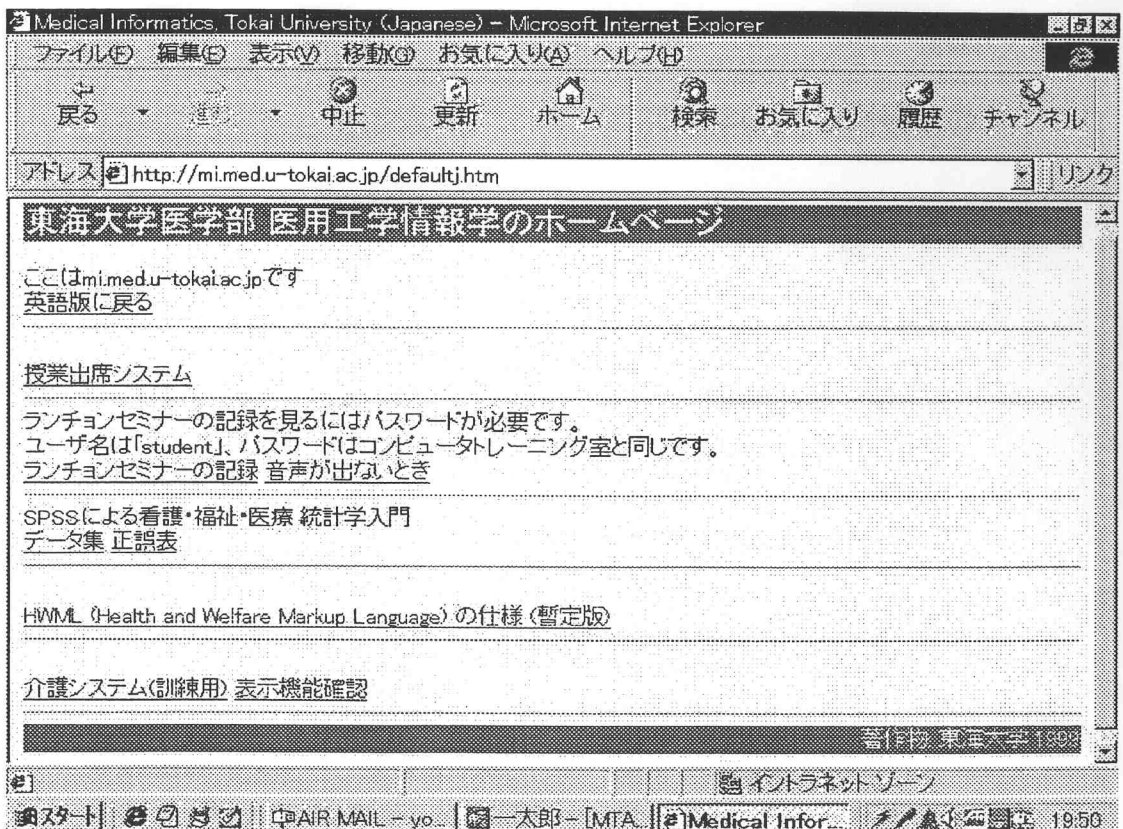
RTNAME は独自に使用しているキーで、ここでは呼ぶべきルーチン名として使用した。

* これらのキーは、%KEY ("MGWLPN"), %KWY ("RTNAME") などとして、以下に引き継がれる。

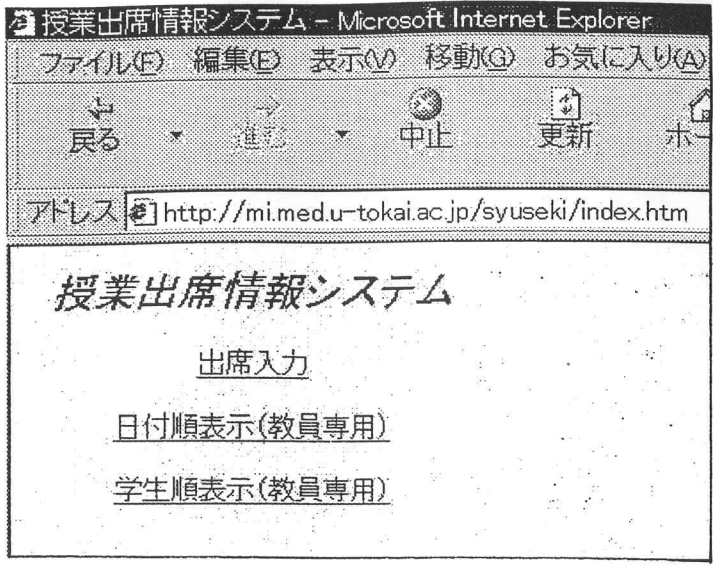
- (2) 学生用の出席登録はオープン利用としたが、教師用の出席一覧表示にはセキュリティを設定した。このため、MGWMS32.dll を2ヶ所に分けて、WindowsNT File System (NFS)の機能を利用してアクセス制限を行った。
- (3) アプリケーションに引き継がれる変数は、先の%KEY と%CGIEVAR がある。ここでは、出席登録端末をチェックするために、%CGIEVAR("REMOTE_ADDR")に IP アドレスがセットされてくることを利用した。
- (4) ここでは、%KEY("RTNAME")により、ルーチンを選択して間接実行している。次の一行だけのルーチンである。
- DO ^@%KEY("RTNAME") QUIT
- (5) アプリケーションプログラムであり、公開済みの各種ユーティリティ (参考文献1) から日付時刻関係を使用した。
- (6) 日付毎、学生番号毎のデータはリスト用に使用され、IP アドレス毎のデータは登録チェック用に使用される。

4. アプリケーション画面の例

当教室のホームページを次に示す。



ここで、「授業出席システム」をクリックすると次の画面になる。



(1) 出席入力画面

出席登録画面(SYUSEKI)

IP:150.7.171.86

学生番号:

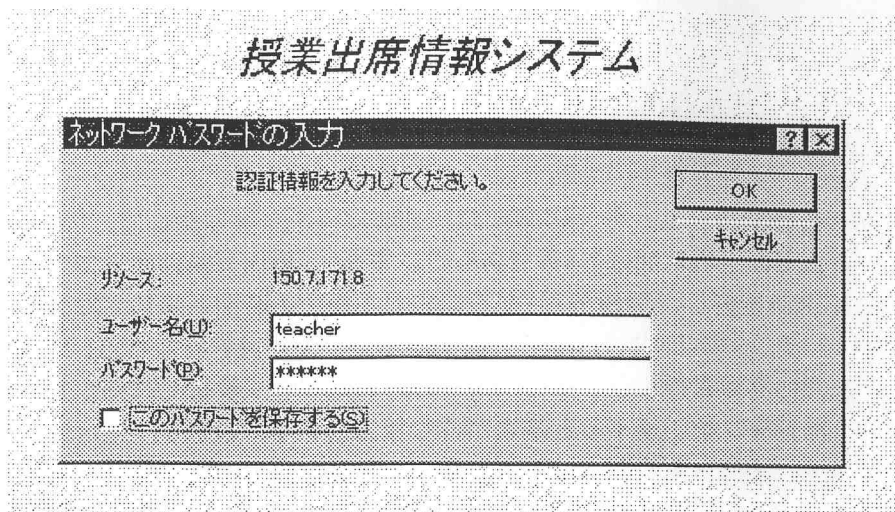
入力例 9JMD1001 9JMN2002 9JMA3003

[戻る](#)

ここで学生番号が入力されると、次のチェックが行われる。

- a) 入力文字のパターン照合
- b) 登録場所のチェック
- c) 時間帯のチェック
- d) 同一時間帯での複数回登録のチェック

(2) 教師用メニュー利用時のセキュリティ画面



これは、WindowsNT の NFS のセキュリティから表示される。

(2-1) 日付毎出席一覧表の表示

表示開始日付:	19990101
表示終了日付:	19990630
表示対象学科:	医学科 看護学科 社会福祉学科
<input type="button" value="表示"/>	

19990423	(1) 9JMM0012 10:43
_____	(2) 9JMM0013 10:43
_____	(3) 9JMM0014 10:44
_____	(4) 9JMM0015 10:43
_____	(5) 9JMM0018 10:44

開始日付と終了日付を入力した後、対象学科を選択して、表示ボタンをクリックすると、日付順、学生番号順に出席状態が表示される。

(2-2) 学生毎出席状況一覧表

表示開始学生番号:	9JMN0001
表示終了学生番号:	9JMW9999
<input type="button" value="表示"/>	

9JMN1234	(1) 19990417 18:09
_____	(2) 19990419 11:52
_____	(3) 19990421 17:25
_____	(4) 19990421 17:29
9JMW0001	(1) 19990421 17:30
9JMW1234	(1) 19990417 17:55
<input type="button" value="戻る"/>	

開始学生番号と終了学生番号を入力した後、表示ボタンをクリックすると、学生番号順、日付順に出席状態が表示される。

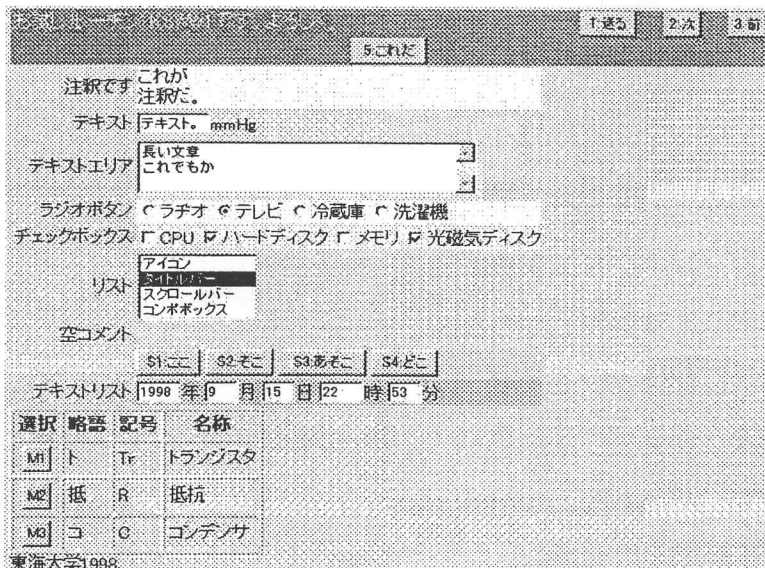
5. 評価結果

パフォーマンスの評価を次のシステム構成で行った。

- ・サーバ DEC pc XL592 Pentium-90MHz 128MB メモリ
Disk 8GB WindowsNT 圧縮形式
- ・クライアント NEC PC-9821Xn Pentium-100MHz 32MB
25台

「授業出席情報システム」の「出席入力」メニューをクリックして、「出席登録画面」が表示されるまでの平均時間は3.5秒であった。また、「出席登録画面」で学生番号を入力して「登録」ボタンを押し、「登録済み」が表示されるまでの時間も同じく3.5秒であった。Mの出席登録ルーチンである SYUSEKI の開始から終了までは0.01秒(\$ZH で測定)であった。また、一斉に登録を行うとライセンスで許されたジョブ数まで起動していき、ライセンスで許されたジョブ数に達すると端末が待ち状態となる。しかし、セッションが拒否されることはなく、端末台数の制限は無い。端末から見た処理時間のほとんどは、セッション接続処理（ログイン）であり、サーバ・アクセス毎にセッションが接続/切断されるためであろう。このような接続形態では頻繁なサーバ・アクセスを伴うアプリケーションでは無理があるが、授業出席登録のように短時間に集中したアクセスのアプリケーションでは少ないジョブ・ライセンスでも実用的であり、経済的なシステムとなる。

画面用部品として、次のコントロールが使用できる。



結論として、Cache の WebLink は大変使いやすい道具であり、コンパクトな WWW アプリケーションには最適であろう。今後、セッション時間の短縮、画面部品の種類の増加、ActiveX や Java Applet との連携などにより、応用が広がるものと思われる。

【参考文献】

1. 大櫛陽一、岡田好一編集：Mプログラミング入門、共立出版、東京、1996.

大域変数の階層構造と日本語文切断のアルゴリズム

高橋 亘

関西福祉科学大学社会福祉学部

〒582-0026 大阪府柏原市旭ヶ丘 3-11-1

TEL 0729-78-0088, FAX 0729-78-0377

E-mail takahasi@fuksi-kagk-u.ac.jp

1. はじめに

視覚障害を持つ人のコンピュータへのアクセシビリティを向上させるためには、入出力の多くを視覚にのみ訴えるユーザーインターフェイスから、聴覚によってもコンピュータの状態を知ることができるような音声出力機能を備えたユーザーインターフェイス、言ってみればユニバーサルインターフェイスに移行することが切望される。また、音声入力によって人が話した内容を読み取りのし易い漢字仮名交じり文に変換することができれば、聴覚障害を持つ人のコミュニケーションへのコンピュータの介助機能が大幅に増進する。音声入出力機能が付加されたユニバーサルインターフェイスは障害者のみならず、健常者にとっても多くの点でコンピュータへの入出力に伴う疲労を軽減することが期待される。このような観点から、障害者と健常者に共通するユニバーサルインターフェイスの開発が急務である。

ユニバーサルインターフェイスの立役者である音声合成装置や音声認識装置がうまく機能するという問題は、常に日本語をどう解析するのかと言う問題と深く関わっている。このような問題を解決するために、まず『現代表記の漢字仮名混じり文』に限って正確な読み上げを保証する日本語知覚関数の作成を目指した。

通常日本語文は、英語文などと異なり、単語ごとに区切られてはいない、従ってまずどの位置で単語に切断されるのかをコンピュータに知覚させなければ、単語すら確定せず、正確に読み上げることはできない。また、漢字を含む同一の単語がコンテキストによって読み方が異なることがある。従って、日本語文を正確に読み上げる人工知能には基本的に次のような大局的アルゴリズムを持たなければならない。

- (1) 基本的には、一般的単語について正確に切断でき、分かち書きができる。
- (2) 読みがコンテキストによる単語について、「その単語の前あるいは後を含めて読みが確定する慣用句」について優先的に切断できる。

このようなアルゴリズムを確立するには、単語や慣用句を、それらの文字数に関する階層構造を明確に捉え、かつデータ化する必要がある。データを階層的に構築し、階層構造をアルゴリズムに組み込んで活用できるプログラミング言語が囑望されるが、このような目的に経済的に対応できる言語としては M 言語がもっとも有望であろう。先の論文 [1] で我々は M 言語による日本語単語の文字数による階層構造がどのようなものであるか、そしてそれをどのようにしてアルゴリズムの道具として利用するのかを提唱した。この論文は先に提唱したアルゴリズムを完結させることを目的としている。

2. 日本語単語の文字数による階層構造化と大域変数

この節では、我々が提唱した日本語単語の文字数による階層構造がどのようなものであるか、そしてそれをどのようにしてアルゴリズムの道具として利用するのか、を明確にするため、先の論文 [1] の論旨に従って復習する。

宮沢賢治の「銀河鉄道の夜」の一文を例として用いたい。

「みなさんは夜にこのまん中に立ってこのレンズの中を見まわすとしてごらん下さい。」

この文の中で「見」という漢字は「み」と読むのが正しいが、同じ漢字は「けん」とも読むことができる。これの読みが決定するのは「見まわす」という合成語によってである。上の文を、単語(あるいは読みの単位)に分割していくとして、「見」という文字の直前までが完了している時に、「見」という文字に引き続く単語、つまり「す」の後で分割が可能であり「見まわす」という言葉が一つの単語であるということを判定する方法である。このようなアルゴリズムを考えると単語の辞書を用意しておき、「見」「見ま」「見まわ」「見まわす」「見まわすと」…といった言葉が辞書の中に登録されているかどうかを判定することを実行し、「見」「見まわす」が登録されていることを確認し、見つけられたものの内、最も長いものをとるという方式が考えやすい。現時点では、この文は「見」以後 13 文字しかないのでこの方式では 13 個の単語が登録されているかどうかを調べることになるが、このような試行回数は文の先頭から順に単語に分割していくに際して莫大な試行を要求する事になる。

M 言語によって、このような莫大な試行を避けるには「見まわす」という単語の辞書登録の仕方

WORD("見", "見ま", "見まわ", "見まわす")="ミマワス"

の様にすればよい。このような辞書は全ての単語が登録された後、図のような階層構造を持つことになる。

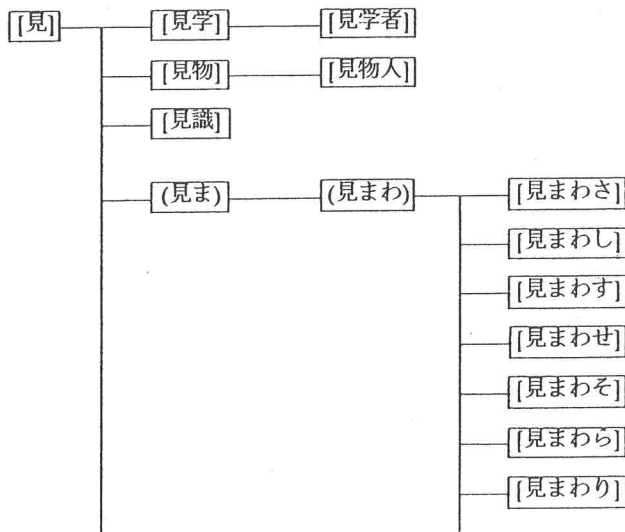


図: 見で始まる単語の階層構造

この図で、[]はそこにデータ(つまり読み)があることを示し、()はそこにデータが無いことを示している。M 言語には標準関数として、大域変数の階層構造を調べるための関数、\$DATA 関数が用意されているが、この関数が日本語切断のチェッカーとしての役割を果たす。大域変数 WORD が上のよ

うな形式で登録されていると、

\$DATA(WORD("見"))

\$DATA(WORD("見", "見ま"))

\$DATA(WORD("見", "見ま", "見まわ"))

\$DATA(WORD("見", "見ま", "見まわ", "見まわす"))

の値が順に 11, 10, 10, 1 となる。つまり \$DATA の値が 1 になった時点で下位のデータがないことが歴然であるから、4 回以上の試行は不要であることがわかる。また、上の 3 つの値は総て下位のデータがあることを物語っているから、「見まわす」を見落とすことはない。

3. 平仮名綴りの鎖状連結を回避する方法

第 2 節で述べたアルゴリズムを完結することをはばむ平仮名文特有の問題がある。この問題は一言で言うと平仮名文の多くが文の先頭から順に切断するという方法に馴染まないということである。一例を挙げてこの問題の本質を明らかにしたい。

「路はじめじめして」(風の又三郎)

という文に対しあらかじめ「はじめ」、「はじ」等の語が辞書に登録されているとすると、単純に前から決定していくアルゴリズムでは

「路 はじめ じめ して」

と切断する。従って「じめじめ」という副詞を認識するには「はじめ」は登録してはならない。「はじめ」を登録しないでおく

「路 はじめ じめ して」

と切断してしまうので、同様に「はじ」も登録してはならない。「はじめ」、「はじ」等の語が辞書から削除されてはじめて

「路 はじめじめして」

と、正しく切断される。このような問題の帰着するところは、「はじめじめ」という綴りの中に「はじめ」、「はじ」、「じめじめ」という言葉が一部重複して鎖状に連結しているためである。これを我々は「平仮名綴りの鎖状連結」の問題と呼ぶことにしたい。

鎖状連結のために辞書に登録してはならない一群の単語もしくはその変化形があることは上述の議論で明らかであるが、このような登録不可な語は、辞書から排除された結果として例えば

「彼ははじめその事を知らなかった。」

といった文の中で、

「彼ははじめその事を知らなかった。」

等として「ははじめ」のよう部分を残すことになる。したがって前から順に切断した後このような部分を見つけ出して

「ははじめ」 → 「はじめ」

のように再構成してやる必要がある。

4. 文法的知覚の方法

第3節で、平仮名文の鎖状連結を回避するために辞書登録不可な一群の平仮名单語が有ること、これらの単語に関連して過多に切断された平仮名文の再構成が必要であることを述べた。このような再構成においてもう一つの問題が浮上する。つまりスペースにより過多に切断された平仮名文のスペースを詰めるべきか詰めざるべきかを局所的に見ていたのでは決定不可な場合が存在する。例えば「では」も辞書登録不可な単語であるが、「では」のスペースを詰めるべきか詰めざるべきかはこれだけでは決定できない。接続詞「では」と、格助詞「で」＋副助詞「は」、の二様の場合が有るからである。しかしながらこのような問題に関して文法的知識がこれを判断する方法を与える。この例の場合、『接続詞「では」』は文頭に現れ、『格助詞「で」＋副助詞「は」』は文中に現れるという特性がある。

この問題はもはや日本語をどのように読むのかという問題ではなく日本語文中の品詞をどのように確定させるのかという問題である。このような品詞理解の主要な鍵を漢字かな混じり文の漢字を取り去った平仮名部分が握っているのは日本語のもっとも顕著な特徴であろう。

5. これからの課題

第3節、第4節で述べた問題は日本語を正しく読むために構文理解の問題が関与していることを示唆している。この事は裏返せば日本語切断の問題を解決することは日本語の構文解析の問題を既に射程に置いていることになる。我々の日本語を単語に切断する方法は構文解析の品詞認識の段階を既に捉えており、次の段階として存在する構文パターン種分けの問題が見え始めている。我々の次の課題は構文解析から和英翻訳の問題である。

引用文献

- [1] 高橋 亘, “視覚障害者のためのヒューマンインターフェイスにおけるユニバーサルデザインと人工知能”『関西福祉科学大学紀要』No. 1, 41-49 (1998).

診療データベースからのデータマイニング

津本周作¹⁾, 高林克日己²⁾, 山崎俊司^{*3)}

- 1) 島根医科大学医学部医学科医療情報学
〒 693-8501 出雲市塩冶町 89-1
TEL: 0853-20-2172 FAX: 0853-20-2170
E-mail: tsumoto@computer.org
- 2) 千葉大学医学部第二内科
〒 260-0856 千葉市中央区亥鼻 1-8-1
- 3) 琉球大学医学部附属病院医療情報部
〒 903-0215 中頭郡西原町字上原 207

1. はじめに

病院情報システムの発展とともに計測データの電子化、データベース化が著しく進み、癌プロトコルデータベースをはじめ、膨大なデータの集積が可能となってきた。このような電子化の進展によって集積された膨大なデータは人の処理能力をはるかに越え、計算機による有効な使用方法確立の必要性が指摘されている。さらに、電子カルテによる検査以外の診療情報がすべて電子保存されてくれば、データの蓄積量はますます増加すると思われる。

このような現在蓄積されつつある膨大なデータベースから癌や痴呆をはじめとする難病の予防に関する知識、あるいはより一般的に診療に役立つ知識を抽出する方法の開発は蓄積されたデータの再利用の観点から極めて重要であり、従来の統計学的手法のみならず、Data Mining 及び Knowledge Discovery in Databases (KDD) におけるルール生成法が期待されている [1]。

本論文では、大規模な病院で収集されたデータベースにルール生成法を適用した研究について報告する。さらに、現在、行っている千葉大学付属病院の病院情報システムから抽出したデータを用いた研究について概観、ルールの専門家による評価を通じた知識発見の可能性について論じる。

2. ルール生成法

2.1 解析・評価の基本的手法: 分割表

本論文で使用される手法は統一的に分割表の中で表される変量によって記述することが可能である。ただし、前向き、後ろ向き研究はこれらの変量に関する測定の手順が異なり、この順序は分割表上に明示的には記述できない。ここで用いる分割表は表 1 のような 2×2 の分割表である。

表 1: Contingency Table

		D(Target)		計
		+	-	
R(formula)	+	a	b	a+b
	-	c	d	c+d
計		a+c	b+d	T

2.2 Accuracy と Coverage

この表から、ルール生成で用いられる条件付き確率の代表である accuracy(confidential ratio) 及び coverage (support ration) は次のように定義できる

$$\alpha_R(D) = \frac{\text{card}([R] \cap D)}{\text{card}[R]} = \frac{a}{a+b} = P(d|R), \quad (1)$$

$$\kappa_R(D) = \frac{\text{card}([R] \cap D)}{\text{card} D} = \frac{a}{a+c} = P(R|d) \quad (2)$$

ここで、 $[R]$ はある属性=値の対からなる連言 R を満たす集合、 D は目標となるクラス d に属する集合を表現している。これら二つの指標はスクリーニングの理論では、それぞれ positive predictive value 及び sensitivity(=true positive rate) に相当する。

2.3 ルール生成

相関ルールを含め、ルール生成においては accuracy 及び coverage あるいはそれらの関数を指標として用いてルールを生成する。したがって、一般に、

$$R \stackrel{\alpha, \kappa}{\rightsquigarrow} d \text{ s.t. } f_1(\alpha_R(D), \kappa_R(D)) > \delta_{f_1}, \dots, f_n(\alpha_R(D), \kappa_R(D)) > \delta_{f_n}.$$

の形で表される。上式で、 R は属性=値の対の連言、 $\delta_{f_1}, \dots, \delta_{f_n}$ は二つの確率的測度のいき値を表し、問題領域に依存して設定されるパラメータである。

この中で、最も単純なものが、accuracy と coverage そのものを基準として使用したもので、

$$R \stackrel{\alpha, \kappa}{\rightsquigarrow} d \text{ s.t. } \alpha_R(D) > \delta_\alpha, \text{ and } \kappa_R(D) > \delta_\kappa.$$

の形で表され、この形式のルールは図 1 のようなアルゴリズムで生成することができる。Accuracy と Coverage が有用であるのは、以下の小節のように、その二つが trade-off の関係にあり、ルールの生成の制御が簡単だからである。

2.4 Trade-Off 関係

正確度 Accuracy と被覆度 Coverage の二つの指標の trade-off は Bayes の公式から容易に示すことができる。

$$\alpha_R(D)P(R) = P(D|R)P(R) = P(R, D) = P(D)P(R|D) = \kappa_R(D)P(D)$$

ここで、Accuracy $\alpha_R(D)$ を増加させるために、式 R の記述長を長くすれば、 $P(R), P(R, D)$ は減少、 $P(D)$ は一定であるから、 $\kappa_R(D)$ は減少する。Coverage が式 R の記述長によって単調に減少することは容易に示すことができるから、これは記述に対するコストであるとみなせば、Accuracy と Coverage の trade-off 関係は MDL 原理 [3] と同等であると考えることができる。

2.5 ルール生成と分割表

Accuracy と Coverage を使用したルール生成の方法の場合、分割表との対応を考えることができる。Accuracy と Coverage の定義から、ルールが R と D との intersection (表 1 では a) を R の立場からみるか、 D の立場からみるかを示している。したがって、ルール生成は各 R と D に対して、分割表を生成し、その中で accuracy と coverage を算出、閾値以上の条件をみたす R をルールの条件部として採用する。

ここで、 R に関して、ベイズ流の解析も含めて、統計的解析では各変量の独立性が問題となるが、ルール生成の場合、集合の intersection を算出する段階で、この独立性のチェックが組み込まれていると考えることができる。

procedure *Induction of Probabilistic Rules*;

var

i : integer; *M*, *L_i* : List;

begin

L₁ := *L_{er}*; /* *L_{er}*: List of Elementary Relations */

i := 1; *M* := {};

for *i* := 1 to *n* **do** /* *n*: Total number of attributes */

begin

while (*L_i* ≠ {}) **do**

begin

Calculate Accuracy and Coverage;

Sort *L_i* w.r.t. the value of coverage;

Select one pair $R = \wedge[a_i = v_j]$ from *L_i*, which have the largest value on coverage;

L_i := *L_i* - {*R*};

if ($\kappa_R(D) \geq \delta_\kappa$)

then do

if ($\alpha_R(D) \geq \delta_\alpha$) **then do** *S_{ir}* := *S_{ir}* + {*R*}; /* Include *R* as Classification Rule */

else *M* := *M* + {*R*};

end

L_{i+1} := (A list of the whole combination of the conjunction formulae in *M*);

end

end {*Induction of Probabilistic Rules*};

図 1: An Algorithm for Probabilistic Rules

3. 細菌検査データベース

使用したデータベースは某病院の細菌検査に関する医療データ (約 20000 件) であり、1994 年一年間に受け付けた細菌培養検査の結果、検体に関する情報及び関係する血液、尿検査データを約 60 の属性によって記述している。本データベースは [7] で用いた髄膜脳炎 (meningoencephalitis) の鑑別診断に関する医療データとは異なり、いくつかの決定属性を除き、欠損値を含んでおり、一部の属性は値が sparse に分布しており、KDD で指摘されてきたデータベースの条件を満たしている。

3.1 属性の説明

データの記述に使用されている属性は表 2 の通りである。本データベースの特徴の一つはリストとして記述された項目が存在することであり、通常、リストを表に埋めこむ際は記述の最大限を推定して、その上限値のコラムを用意しておくという方法が取られている。

3.2 データベースの特徴

データベースは実際に蓄積され、前処理を施していない生のデータで、その特徴は以下の通りである。

1. 属性の一部はリストとして保存されている。
2. 欠損値のある項目がほとんどである。
3. 検出菌、疾患名等の項目の値が極めて多く、領域知識による値の変換が必要である。
4. 同時に採取した材料に関するデータが含まれている。

表 2: 細菌検査データベースの属性 (診断)

項目	属性
個人情報	受付日, 性別, 生年月日
診療科情報	病棟, 採取月日時, 病名 (3)
オーダー	材料, 目的菌 (2)
症状等	発熱, カテーテル (5), 気切, 挿管, ドレナージ (5)
検査	白血球, 尿 WBC, 尿硝酸塩, 尿蛋白, 尿潜血, 尿定量
治療歴	投薬, ステロイド, 抗癌剤, 放射線, 消炎剤
培養結果	総菌数, 検出菌, biocode, VITEK カード, β -lactamase
抗生物質感受性	PCG, 合 PCs, Aug, PCs-緑, CEPs-1, CEPs-2, CEPs-3, CEPs-4, CEPs-緑,
抗生物質感受性	AGs, MLs, TCs, LCMs, Cs, CBPs, VCM, RFP/FOM

注. (数字) は項目数を示している。

5. 一部、時系列データも含まれている。

一番めの項目に関しては、現行の表形式データベースへの保存が前提になっているためである。この項目のためデータが sparse であるような場合もありうるが、本データベースはその効果を取り除いても、データが sparse になっている。四番め、五番めの項目に関しては、一部、同じ被験者から取った空間的ないし時間的に異なる検体も一括して登録されており、個々のデータの独立性は必ずしも保持されていない。

これらは KDD の研究で指摘されていたことであり、前処理、data cleaning がデータマイニングの非常に大きな部分を占める。

3.3 解析の Target

データベース中に含まれる検出菌、病名の項目数は極めて多く、このままでは汎化した知識は抽出しにくい。このため、これらの項目はグループ化して、よりルールを生成しやすい形へとデータベースを変換した。

以上の処理を踏まえて、以下の二つの Target に絞って、データを解析する。

1. 原因菌の欄は大きく分ければ、検出菌 (+) と (-) にわかれる。それぞれの場合、どの属性と関係があるか検出する。
2. 検出菌 (+) の場合、抗生物質の感受性テストにおいて、R(resistant: 抵抗性) という項目と検出菌の関係、あるいは R と S(sensitivity: 感受性あり) との場合で、どの属性と関係があるか検出する。

3.4 生成されたルール

Accuracy と Coverage の閾値をそれぞれ 0.5, 0.1 と設定して、ルールが 2195 個得られた。そのうちで、表 3 に示したような有用なルールが得られた。この表からもわかるごとく、検出菌の中で、いわゆるペニシリン耐性菌がほとんどであり、肺炎では検出率は悪い。また、MRSA で VCM 耐性が 10% ほど見られている点、神経内科病棟で圧倒的に MRSA 感染が蔓延している現状などがルールから推察できる。

4. 膠原病データベース

著者らは、千葉大学医学部附属病院の病院情報システムにたくわえられたデータベースから、膠原病に関するデータを抽出した。本データは以下の 3 つのデータ集合から成っている: データ 1: 膠原病外来で経過観察されている患者さんについての基本情報が含まれている (属性数:6)。データ 2: 血栓症に関わる特殊検査の情報が含まれている (属性数:14)。データ 3: 千葉大学附属病院の病院情報システムに蓄えられた時系列データ (1980 ~ 1999.3)(属性数:46)。以上のデータは、膠原病にかかわると思われる約 66 近い属性で記述されており、このデータを用いて、現在

表 3: 生成されたルール

β -lactamase(+) → 検出菌 (+)	(Accuracy: 0.667, Coverage: 0.541)
β -lactamase(3+) → 検出菌 (+)	(Accuracy: 0.702, Coverage: 0.553)
肺炎 → 検出菌 (-)	(Accuracy: 0.826, Coverage: 0.12)
発熱 (39) → 検出菌 (-)	(Accuracy: 0.790, Coverage: 0.11)
悪性腫瘍 → 検出菌 (-)	(Accuracy: 0.77, Coverage: 0.13)
<i>Fusobacterium</i> → <i>PCG(S)</i>	(Accuracy: 0.92, Coverage: 0.26)
<i>MRSA</i> → <i>VCMS(S)</i>	(Accuracy: 0.89, Coverage: 0.12)
扁桃炎 → <i>Aug(S)</i>	(Accuracy: 0.84, Coverage: 0.10)
神経内科病棟 → <i>MRSA</i>	(Accuracy: 0.6, Coverage: 0.4)

1. 血栓症 (Thrombosis) の診断に有効な属性・時間的パターンの解析
2. 各膠原病を正確に診断するためのパターンの解析
3. 各膠原病に特徴的な時間的パターンの解析

をターゲットとして、解析中である。

5. 考察: 仮説生成としてのルール生成

著者らは昨年度の人工知能学会全国大会でのパネルセッション [6], 第 42 回知識ベース研究会 [7], 本年度の人工知能学会全国大会での特別セッション [8] において、専門家の評価という観点から postprocessing に関する可能性に関する研究を提案した。これらの試みでは共通のデータを用いて、各パネリスト、参加者が自ら開発した知識発見の手法を比較検討、さらに専門家による生成された知識の解釈を加えて、計算機を用いた発見支援の可能性を検討した。

知識発見 [1] において、従来の機械学習における予測正答率の評価 [4] とは異なり、生成された知識を専門家の観点から解釈することが、発見のプロセスにおいて重要である。

例えば、相関ルールによる分析で有名なスーパーマーケットのデータベースの例を考えてみよう。この例では、ビールの売り上げと紙おむつの売り上げに相関があるというルールが生成されているが、これだけでは、このルールが単に overfitting のようなものか実際の事実を反映したものかは判別できなかったはずである。ルール生成アルゴリズムはパターンを生成するが、パターンの重要性を示してはくれない。この相関ルールの背景知識として専門家が提示した説明は「ビールを買いに来た男性が夫人に紙おむつの購入を依頼されたのであろう」という仮説である。専門家は生成されたパターンからこのような仮説を生成することができたわけである。実際に、この仮説がスーパーマーケットにおいて実証されたわけであるが、この例は知識発見を考える上で示唆的である。

ルール生成は、専門家の知識を抽出するという知識獲得の側面と専門家が予想しなかったパターンを抽出するという知識発見の側面とがある。著者はこの側面が実は独立して完全に分離することのできない、Fuzzy な分類を要求されていることを指摘したい。経験的に Support の高いルールは専門家にとって妥当であるが予想された知識を生成することがわかっているが [9]、実際に専門家の予想しない知識は Support がそれほど高くはないことが多い。つまり、発見に貢献しそうなルールは、いわゆる妥当なルールの「辺縁」にでも存在するかのようである。ここでは、第 42 回 KBS 研究会で発表された新美・田崎による C4.5、ADF-GP による解析が極めて良い例なので彼らの結果を以下に示す [7]。C4.5[2] によるルールは

```
(Cell_Mono <= 12) => BACTERIA
(Cell_Poly <= 140) && (Cell_Mono > 12) => VIRUS
(Cell_Poly > 140) => BACTERIA
```

で、これらは Certainty Factor(Accuracy) と Support が高いルールである。これらは専門家にとっては極めて妥当なルールであるが、発見としての意味は少ない。ただ、これらの的中率は 132/140 で、多くの例がカバーされている。次に、ADF-GP によるルール (一部を抜粋) は以下の通りである。

```

(EEG_FOCUS eq "-") && (CT_FIND eq "normal") && (SEX eq "M") && (RISK eq "n") => VIRUS
(EEG_FOCUS eq "-") && (CT_FIND eq "normal") && (SEX eq "F") && (RISK eq "n") => VIRUS
(EEG_FOCUS eq "-") && (CT_FIND eq "normal") && (SEX eq "F") && (RISK eq "p") => BACTERIA
(EEG_FOCUS eq "+") && (CT_FIND eq "abnormal") && (SEX eq "M") => BACTERIA

```

EEG_FOCUS 及び SEX=F or M がルール項目に含まれているのが専門家としては予想されない知識である。これらのルールの的中率は 112/140 で、必ずしもデータをカバーする率は高くない。上記の因子が重要なものとして選択されたことを専門家は妥当な説明あるいは検証をしようとする。このうち、SEX に関する評価は、すでに著者が報告した [9]。ここで指摘しておかなければならないのは、もし専門家がこのデータを解析する場合、年齢、性別は解析の対象から外されていたかもしれないということである [5]。

以上の例からわかるごとく、パターンとしては正解率やサポートが低くても、生成された知識が、専門家にとって仮説となる場合もありうる。ルール生成は仮説生成の方法として専門家の発見を支援する手法の一つになりうるものである。

6. おわりに

本論文では、診療データベースからの知識発見についての成果およびそれに関する専門家の評価方法についての考察を含め、報告した。今後、電子カルテの進展とともに、蓄積されたデータから有効な知識を抽出、診療を支援するシステムを構築、実装することは、電子カルテの構築を促進する動機づけになる筆者は考えている。また、データマイニングの研究そのものも発展途上であることを考えれば、これらの分野が結合することで両分野の研究がさらに進展することが期待される。

参考文献

- [1] Fayyad, U.M., et al.(eds.): *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 1996.
- [2] Quinlan, J.R. *C4.5 - Programs for Machine Learning*, Morgan Kaufmann, CA, 1993.
- [3] Rissanen J: *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore, 1989.
- [4] *Readings in Machine Learning*, (Shavlik, J. W. and Dietterich, T.G., eds.) Morgan Kaufmann, Palo Alto, 1990.
- [5] 津本周作, 遠間直人, 木内一郎, 渋谷正徳, 小松隆行, 北野邦孝: ウィルス性髄膜炎の予後に及ぼす因子の検討. 日救急医学会関東誌 12,710-711,1991.
- [6] 津本周作. 共通データベースによる知識発見手法の比較と評価. 1998 年度人工知能学会全国大会論文集, 1998.
- [7] 津本周作, 山口高平 (編) 第 42 回知識ベース研究会抄録, 人工知能学会,SIG-KBS-9802, 1999.
- [8] 津本周作. 共通データベースによる知識発見手法の比較と評価. 1999 年度人工知能学会全国大会論文集, 1999.
- [9] Tsumoto, S., Ziarko, W., Shan, N., Tanaka, H.: Knowledge Discovery in Clinical Databases based on Variable Precision Rough Set Model Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care, Journal of the AMIA 2, supplement, pp.270-274, 1995.

CORBAMED の活動と応用例

藤生尚光

(株) ニコン S I 開発部

140-8601 東京都品川区西大井 1-6-3

TEL : 03-3773-6313 FAX : 03-3773-1446

tfujiu@nikongw.nikon.co.jp

1. はじめに

現在、医療情報分野に於いてもオブジェクト技術を用いたインターフェースの標準化が行われようとしている。この活動の中心となっているのは米国に本部のある OMG (Object Management Group) であり、特に医療分野に関しては分科会 CORBAMED が担当分野での各サービスを定義し、これに対する標準インターフェースを作る作業を進めている。本報告では、CORBAMED の活動概況、Mテクノロジーへの対応、最新の応用について紹介する。

2. OMG の活動概況

OMG は CORBA (Common Object Request Broker Architecture) の推進と管理を行なう非営利民間国際標準化団体である。現在、800 以上の団体の加盟があり、2 ヶ月に一度程度の頻度で全分野を対象としたテクニカルミーティングが開催され、併せて各分科会のミーティングも行われる。

図 1 に OMG が提唱する OMA (Object Management Architecture) の概念図を示す。各オブジェクトは、必要なサービスを取得する際に単に ORB に対してのみ相互作用を行なうだけでなく、このことで OMG が目指す Interoperability を達成する。OMA は完成されたモデルではなく、各分野で行われているビジネスのモデル化とこれに伴うオブジェクト/サービスの定義と共に成長するものである。また、各アプリケーションに対しては標準化の活動は行なわず、アプリケーションの作成者は標準化されたインターフェースを介して他のオブジェクトが提供するサービスを利用する。

OMG の活動の中心は標準化のためのインターフェースである IDL (Interface Definition Language) を用いて仕様を決定することであり、基本的に以下のプロセスで行われる。

- ・ RFI (Request for Information) : OMG による情報提供の要求
- ・ RFI Responses : ベンダーからの回答
- ・ RFP (Request for Proposal) : OMG によるインターフェース仕様の提示要求
- ・ RFP Responses : ベンダーからの提案
- ・ 採択

活動はあくまでもインタフェース仕様の決定までであり実装はベンダーに任されているが、RFP Response を出し採択されたベンダーは一年以内に実装する責任を負うことになる。

標準化活動は、ORB とこれに付随する CORBA services の展開、管理、保守等を行なう Platform

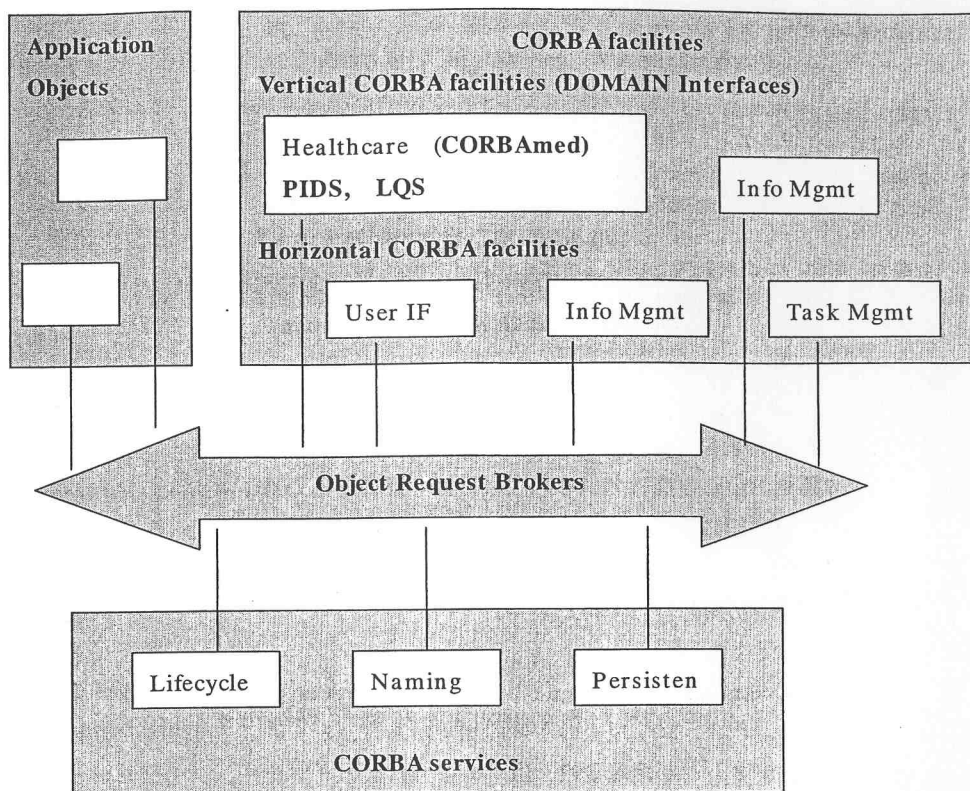


図 1. OMGの Object Management Architecture

Technology Committee と各ビジネス分野毎の IDL (Interface Definition Language) インターフェースを作成する Domain Technology Committee に二分される。DTC には、Electronic Commerce DTF (Domain Task Force), Transportation DTF, Telecom DTF 等があり、一部では既に標準化インターフェースが実用に供されている。CORBAmed も DTC に属し、Healthcare 分野での活動を行なっている。

3. CORBAmed の活動概況

CORBAmed は既に約 3 年の活動を行なっており、DTF への参加団体も当初の数社から約 100 社までに増加している。3 年の間に参加メンバー、団体の変遷も多く、チェア人も何回か代わり、現在はミシガン大学の Ms. Mary Kratz が務めている。

表 1 に現状での提案された各サービスの仕様に関する採択、提案状況を示す。初めの七つのサービスは、基本的に CORBAmed 発足当初に提案されたものであり、他はその後追加されたものである。当初提案されたもののうち、PIDS と LQS は既に OMG の正式なインターフェース仕様として採択されている。他の五つは、内容の変更、統合・分割を経て名前を変えたため、前記二つよりは遅れたが、いずれも RFP への回答提案があり、既に仕様の裁決プロセスに進んでいるものもある。最後の三つは、いずれもまだ RFI の段階に止まっている。

表 1. 採択されたインタフェース仕様と現在検討中の仕様

仕様	状況
Patient Identification Services (PIDS)	採択済み
Lexicon Query Services (LQS)	採択済み
Clinical Observation Access Services (COAS)	採択審議中 → 済
Healthcare Resource Access Control (HRAC)	採択審議中 → “
Healthcare Data Interpretation Facility (HDIF)	RFP 提案受理
Clinical Image Access Service (CIAS)	RFP 提案受理
Medical Transcription Management (MTM)	RFP 提案受理
Pharmacy Interaction Facility (PIF)	RFI 回答なし
Healthcare, Admini, Logi, Financial Management (HALFEN)	RFI 回答なし
CORBA/M Interoperability	RFI 回答受理

PIDS は、複数の異なった患者 ID 体系を持った医療機関に跨って患者を特定するためのサービスである。病院の統廃合が益々盛んな米国では極めて需要の多いサービスであり、一番始めに採択されることとなった。現在のところ最も多くのベンダーが RFP の回答提案に参加したサービスである。LQS は、医療記録を記述するための複数の辞書の相関を取り、記述内容が同じとなることを保証する。SNOMED、UMLS 等が対象の辞書。COAS は検査データから診断までの医療記録を取得するためのサービスである。CIAS はそのサブセットと見なせるものであり、DICOM 準拠画像から参照画像を取得するためのサービスである。

HRAC は CORBA の持つセキュリティにさらに医療独自のセキュリティを追加するものであり、HDIF は診断支援、MTM は COAS に基づくドキュメント管理を提案するものである。RFI の段階に止まる三者は、それぞれ、薬剤システムとのインタフェース、事務管理部門とのインタフェース、M とのインタフェースを提案するものである。

4. CORBA/M Interoperability

1998 年 2 月に、RFI が OMG から発行された。99 年 5 月の段階では、RFI への回答がベンダーから提出され OMG により受理されたものの、RFP はまだ出されていない。RFI の回答受理から RFP の発行までの時間に関する OMG の規則はないものの、他の例と比較するとやや遅い対応であると言える。以下に同 RFI から Information Sought の要点を列記する。

- Reference Information : M とのインテグレーションに関する、書籍、論文、URL の情報
- Business Opportunity : 技術的に統合できるか否かの検討では不十分であり、ビジネスとして成り立つか否かを見極めるための情報が必要。例えば以下の情報。
 - M のユーザとベンダーが CORBA の利用を望んでいるか
 - M テクノロジーの市場は拡大するか
 - CORBAmed の市場は拡大するか
- M における CORBA のニーズ : 例えば、以下の情報。
 - M システムの内部から外部のオブジェクトを使う必要があるか (逆も)
 - M global や M dbms を CORBAservice がアクセスする必要があるか

○PIDS、LQS 等を使う必要があるか

・ 予想されるソリューション

○OMG IDL と M 言語とのマッピング

○CORBA Service、CORBAmed Service のMでの実装

5. 応用例

Baptist Health Systems of South Florida はマイアミを中心とした総合病院グループであり、約 1400 人の医師を擁し、年間 26000 人の入院患者を受け入れている。独自の診断・治療プロトコルを作成し、医療行為の規格化を進めている。これを技術的に支えるのが CORBA をベースとした医療情報システムである。

現在全てのシステムが CORBA ベースではないものの、新規購入システムには必ず IDL からなるインタフェースを追加し、また既存システムへも一部で IDL インタフェースを作成して、実運用を行ないながら完全 CORBA 化への移行を進めている。但し、CORBAmed Service はまだ利用されていない。本施設の特徴は、ベンダーから新システムを購入する際に、OMG に準じた RFI、RFP プロセスを取っていることである。プラットフォーム、ハードウェアに関らず、ベンダーにはシステム納入の際に IDL インタフェースを提供させることとしている。

6. 参考情報

CORBA Fundamentals and Programming, Jon Siegel, John Wiley & Sons, Inc., 1996

www.omg.org/homepages/corbamed/

sumiACCEL/21(仮称)における電子カルテ機能について

○村上 英 木戸須美子 青木 浩二 上戸 隆 藤江 昭
住友電工システムズ(株)

〒542-0081 大阪市中央区南船場 4-11-28 (サン船場ビル)

TEL: 06-6258-5518 FAX: 06-6258-5520

murakami-ei@sesys.co.jp

1. はじめに

sumiACCEL/21(仮称)は、当社が 21 世紀の日本の医療情報における課題に対応すべく、鋭意開発中の次世代病院情報システムソリューションである。電子カルテの実現も、その対応すべき重要な課題の一つである。本稿では、sumiACCEL/21(仮称)の電子カルテ機能について技術的な観点から紹介する。

2. sumiACCEL/21(仮称)の概要

sumiACCEL/21(仮称)は、米国 Per-Se 社の医療情報システムパッケージ ULTICARE をベースに、当社の日本における病院情報システムに関するノウハウを組み込み、日本の医療情報システムのニーズに適應するよう改良したものである。sumiACCEL/21(仮称)は以下のような特徴をもつ。

- あらゆる患者に関するデータを、統一的な操作により、統合的に扱える患者中心の設計思想。
- 医療従事者のワークフローに沿った処理の流れを可能とし、診療のその場でのリアルタイム診療支援を指向。
- 入力されたデータの自動評価機能と、その評価に基づく表示や動作の切替え、適切なユーザへの警告通知などが可能。
- オブジェクト指向設計により、医療施設のビジネスモデルを内蔵し、各種設定テーブルの設定変更とコンパイルにより、プログラムレスで各ユーザに対応したシステムが構築可能。

なお、sumiACCEL/21(仮称)は、ULTI-MUMPS という独自拡張の MUMPS でプログラミングされているが、システムの安定性のため、ユーザにプログラミングを開放しない設計である。ただし、各種のツール群に MUMPS ライクなロジック記述手段が用意されており、データベースに格納されたデータを検索表示できるだけでなく、警告機能、ワークフローの制御など電子カルテのインテリジェントなふるまいを設定することができる。

3. 当社電子カルテ開発の取り組み

一方、当社は千葉大医学部付属病院殿との共同研究により、医師のための診療支援環境の提供を目的として、実用に耐え得る電子カルテの研究開発に取り組んできた。その特徴は以下の通りである。

- (1) プロBLEM(問題)志向型診療記録
患者の診療における情報の見落としを防ぎやすく、患者の問題解決の糸口を発見しやすいといわれるプロBLEM(問題)志向型診療録の考え方を基本とする。
- (2) データの標準化
データ形式を標準化することにより、投薬歴と検査結果の変化を同じ時間軸上にグラフ表示するなど、診療に役立つ種々の加工表示ができる環境を提供する。また、参照画像 DB などとの連係によりマルチメディアデータにも対応する。
- (3) テンプレート機能
標準化されたデータ項目のセットであるテンプレートを、個々の患者の病態に合わせて適用し、そのテンプレートに沿ってデータの記録および整理を行うことによって、記録の抜けを防ぐ。

4. 取り組みの継続性

sumiACCEL/21(仮称)における電子カルテ機能には、上記のこれまでの開発成果を継承するための機能が用意されている。

(1) プロブレム(問題)志向型診療記録

sumiACCEL/21(仮称)には、病名オーダと連係したプロブレム管理機能が備わっており、オーダ、実施入力、経過記録といった診療イベントをプロブレムに関連づける機能をもつとともに、プロブレム毎の治療計画機能をもっている。

(2) データの標準化

取り扱う全てのデータはデータエレメントと呼ばれる最小単位で管理される。データエレメントは異なる画面、機能で共有が可能でデータの統合化が実現される。各種データエレメントを統一的に扱うことができるため、投薬歴と検査結果の変化等のトレンド表示、グラフ表示が容易にできる。また、データエレメントにはマルチメディアタイプも用意され、参照画像等にも対応可能である。

(3) テンプレート機能

上記データエレメントを組み合わせることで入力画面を自由に構築可能であり、その入力画面を治療計画機能、クリティカルパス機能、各種カスタマイズ機能と組み合わせることで、個々の患者に応じたテンプレート機能を実現可能である。

5. 新しいアプローチと今後の発展性

sumiACCEL/21(仮称)では、上記のように、従来からの当社の電子カルテ開発のノウハウを継承できる機能を持っているが、それに加えて、より使いやすく、機能的な電子カルテを実現するための機能を持っている。

(a) リアルタイムの診療支援機能

最近 24 時間の患者の病態に関する情報を一覧表示する 24 時間サマリ機能、疾患毎のキーとなる診療データについて最近実施日時、データ、正常値、結果判定および一定期間未実施警告を一覧表示する診療プロトコル機能など多くのリアルタイム診療支援機能を持つ。これらは設定を調整することにより、容易に各施設の状況に合わせてカスタマイズ可能である。

(b) Java による柔軟な GUI

クライアントの GUI の構築に Java を採用することにより、柔軟な GUI とクライアント管理負担の軽減を両立する。

(c) HTML 出力機能

診療データベースに格納されたデータから、リアルタイムに HTML の出力を生成することが可能。生成された HTML の出力は電子カルテの画面にインラインで組み込まれる。

(d) ほぼ全てのオーダ種を統一的にサポート

オーダ種の区別を意識すること無く、セットオーダ、連動オーダなど多様なオーダを有機的にコントロール可能である。

6. まとめ

以上、述べたように sumiACCEL/21(仮称)は、電子カルテのプラットフォームとして従来からの当社の電子カルテに対する取り組みを継承しつつ、さらなる発展を可能とする機能が装備されている。今後各ユーザ様のご意見をもとに、より完成度の高い電子カルテシステムとしていく所存である。

標準化を目指す新患者検査データベースの設計と再構築

○姜 琳¹⁾ 本多 正幸²⁾ 鈴木 隆弘²⁾ 高林 克己²⁾
岩橋 俊樹¹⁾ 四方 昭弘¹⁾ 里村 洋一²⁾

住友電工システムズ株式会社¹⁾
千葉大学医学部附属病院医療情報部²⁾

〒260-8677 千葉市中央区亥鼻1-8-1
TEL: 043-226-2346 FAX: 043-226-2373
Kyorin@ho.chiba-u.ac.jp

1 はじめに

千葉大学病院には、情報システムを最初に導入して以来20年あまりにわたって、患者基本情報、検体検査情報、輸血情報、感染症情報、看護情報、処方オーダ情報などの各種患者情報を蓄積し、検索システム、研究支援システムを通してそれらの情報の有効活用を図っている。現時点において、患者検査DBは2000年問題、時系列表示上の問題、属性情報が不十分、データの交換などの問題点がこれまで以上に緊急で、かつ重要性を増してきた。この度、21世紀への医療構造改革を見据え、地域医療との連携を考慮したデータベース(DB)の設計を図り、将来的に各種の要求に耐え得るDBを目指した。具体的設計に当たり、HL7を意識し、医療施設間、医療情報システム間とのデータ交換が容易となることを考慮した。また、今後発生してくるであろう新しい患者情報(例えば、細菌検査、血中濃度検査、看護情報など)をこのDBに容易に取り込むことについても可能となるよう考慮した。

本報告では、M言語(MUMPS)を利用し、HL7を考慮した新DBの構造を述べ、蓄積されたDBの移行上の問題点とシステム間の連携などの問題点を検討する。

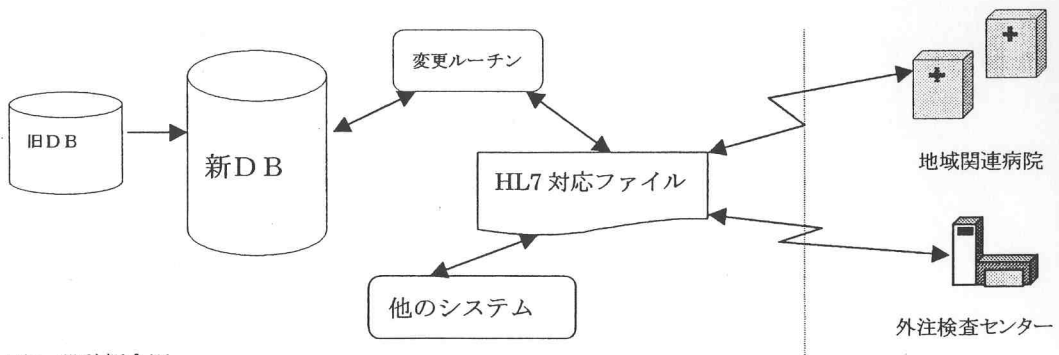


図1 設計概念図

2 設計方針

既存のデータベースの継続利用を考慮した新DB設計に当たり、以下の目的を掲げた。

- 正確かつ広く多様な通信環境で、迅速な診療情報の伝達。
- 情報の共通利用性を確保する(HL7準拠)。
- 分散している膨大な量のDBの整理、統合。
- 既存DBの問題点(検査DBの2000年問題、時系列表示上の問題等)の解消。

- 拡張性を持たせる。

以上の5つのポイントを中心に、新DBファイル仕様(表1)を設計し、段階的移行計画をたてた。第1段階として従来の検査DBから新検査DBへ移行すること、第2段階として、新検査DBから、変換プログラムを通じて、HL7対応ファイル(表2)に変換することとした。HL7対応ファイルは施設間或いはシステム間で転送に利用される。これにより、外注検査或いはサブシステムなどの情報交換も簡便に行えるようになる。

検査DBは80年代の初めの設計しただであったため、当時から物理的容量が制限され、データは圧縮の形で保存してきた。以下これまでに懸案となっていた具体的な問題点への対応を行う。

- A. 2000年問題。検査DB上での日付はYYMMDDの6桁形式で保存している。
- B. 時系列表示する詳細時刻の不足。同一日で同じ項目の複数検査結果は検査番号で区別してるが、時刻情報が保存していないこと。
- C. 項目の変更、追加の処理が複雑。検査項目は検査結果と一対一になっていないこと。
- D. 結果に関する付帯情報が乏しい。結果に対しての判定内容とコメントなどの情報を保存していないこと。
- E. 院内検査、外注検査、速報結果のファイルが分散しているため、検索および表示プログラムが複雑。

上記の問題に対応するため、日本臨床病理学会の臨床検査項目分類コード(表3)を採用した新検査結果DBを設計した。これにより、時系列対応、新項目の追加等の簡便化、分散しているDBの統合等の問題が解決されるであろう。新DBの設計は基本的に分析物の識別コードごとに結果と付帯情報を保存し、検体IDごとに報告時刻も保存するなどきめこまかい情報管理が可能となり、情報の増加にも柔軟に対応ができると考えている。

3 方法と問題点

まず第一ステップとして、日本臨床病理学会の臨床検査項目分類コード(表3)に対応した検査項目マスターを作成する。臨床検査項目分類コードが不十分のため(生理検査の肺機能部分など)、一時的に独自のコードを利用し、臨床検査項目分類コードが追加されてから、再度作成する。

次のステップとして、新ファイル仕様のDBのリハーサルを行い、従来検査DBから、新検査DBへの移行に必要な容量を計算する。一部の情報をういた試算では、約8.5倍になった。従ってディスク容量の確保を図らなければならない。

日本臨床病理学会コードでは、分析物、材料、検査方法などの複合コードであり、検査方法を変更すると、別項目として扱う仕様となっているが、システム利用する場合は一つの項目として認識し、変更に対する不変性を持たせる必要がある。項目名称マスターについては複数検査方法に対して同じ項目名で管理することが必要であり、標準値の変更には日付情報をつけて管理することで対応することとした。

以上のことを考慮し、リハーサルを行った後、従来検査DBから、新検査DBへの移行を行う。

最終的には、システムセキュリティの確保を前提条件に、施設間のデータ転送の実現を目指す。新DBのデータを変換プログラムを通じて、HL7対応ファイルに変換され、作成されたファイルが施設間(外注検査など)で転送できることを目指す。

また、HL7インターフェイスを採用しないサブシステムには、従来のインターフェイスを利用し、変換テーブル作成することで対応することとした。

4 考察

地域医療及び施設間のデータ交換を前提とするシステムの設計と構築は、まず、標準化が不可欠である。現状の技術で可能な範囲で具体的に提示したが、実際には、臨床検査項目分類コード(表3)を利用すると、コードが不十分であることが分かった。また、長期蓄積されているDBの再構築及びそのDBをプログラム利用している様々なプログ

ラムの改修には、大量な改修作業が必要である。したがって具体的作業を各段階に分けて、実現することになっている。今回のしんDBの作成により、分散されたDBを統合し、患者情報の検索の時間が短縮されることも期待している。

今回はHL7を念頭にシステム設計を行ったが、日本医療情報学会におけるMML/MERIT-9等の変換規約の整備を待って、これらを総合的に取り込んだシステムとしていきたい。

5 まとめ

今日、医療情報の標準化は、患者中心の医療、効率的な医療を進めるにあたって、重要な位置を占めるものである。また、医療は、チーム医療、中央検査室や外注検査センター、遠隔診断、地域医療、在宅医療へと変貌しつつある。これに対応して、医療情報システムの基盤のデータベースの標準化がますます重要になってきている。HL7を意識し、このような医療情報の標準化により、医療機関同士がネットワークを設けてお互いに患者の検査結果を参照することも可能になる。患者が複数の医療機関で受診しても同じ検査を繰り返す必要がなくなり、患者の負担も減るとみられる。実現すれば複数の医療機関がネットワークを通じて患者の検査記録などを共有し、患者の病状を迅速に把握できるようになり、患者疾患の治療にも役立つなど効果は計りしれない。

参考文献

- [1] 日本医療情報学会課題研究会の「MML/MERIT-9研究会」 <http://www.h.u-tokyo.ac.jp/mml/>
- [2] 日本臨床病理学会:臨床検査項目分類コード, 第10回改訂・第1版, 臨床病理出版, 1998
- [3] 保険医療福祉情報システム工業会 臨床検査システム委員会:医療情報交換仕様 JAHIS 臨床検査データ交換規約 Ver.1.0
- [4] Health Level Seven, Inc.: Health Level Seven Version 2.3 An application protocol for electronic data exchange in health environment
- [5] 木村 通男:医療情報交換規格運用指針:MERIT-9, 医療情報学, Vol.18 No.4 January 1999

表1 新検査結果ファイル仕様

臨床検査結果ファイル(仮名称 ^PMMXS)
^PMMXS(%HSP):
①Key: 病院コード ...%HSP X(3)
^PMMXS(%HSP,PID):患者基本情報
②Key:患者番号(外部)...PID F9(7)
^PMMXS(%HSP,PID,BCD):検査分類情報
③Key:検査分類コード...BCD F9(3)
^PMMXS(%HSP,PID,BCD,UYMD):患者付加情報
④Key:受付日付...UYMD F9(8) YYYYMMDD
^PMMXS(%HSP,PID,BCD,UYMD,KID):依頼と受付情報
⑤Key:検査ID(項目群)...KIDV F9(8)
^PMMXS(%HSP,PID,BCD,UYMD,KID,SKID):検査サブIDノード
⑥Key:検査サブID...SKID F9(4)
^PMMXS(%HSP,PID,BCD,UYMD,KID,SKID,LCD):項目情報
⑦Key:検査項目コード(運用コード)...LCD F9(6)
^PMMXS(%HSP,PID,BCD,UYMD,KID,SKID,LCD,HRC):結果情報
⑧Key:結果コード...HRC DF9(8)

表2 HL7対応ファイル

MSH ^~\& EKG CDB ORF^R04 X980701 P<CR>
MSA AA CDB2222 P<CR>
QRD 199807210943 R Q0012 10 RD 901118-8 RES<CR>
QRF EKG 199807300000<CR>
PID 1 901118-8 CHIBA^TAROU^ 19600101 M (043)222-7171 <CR>
OBR 1 9998^OE CM0062^LAB 199807211530 199807290800 9985^CHIBA^^^MD^ L SER 9997^INOH ANA^CHIROU^^^DR. (043)226-2346 199807221400 F<CR>
OBX 1 ST 3B035-0000-023-271-01^GOT 25 iu/l 7-40 N A F 19980721<CR>
OBX 2 ST 3B045-0000-023-271-01^GPT 17 iu/l 7-40 N N F 19980721<CR>
OBX 3 ST 3B050-0000-023-271-01^LDH 502 iu/l 216-450 H N F 19980721<CR>
OBX 4 ST 3B070-0000-023-271-01^ALP 396 iu/l 72-206 H N F 19980721<CR>

表3 日本臨床病理学会の臨床検査項目分類コード

新運用コード	15桁コード表示	分析物	識別	材料	測定法	分析物名
100010	1A0060000001920	1A006	0000	001	920	色調[尿]
100020	1A0070000001920	1A007	0000	001	920	混濁[尿]
100030	1A0100000001901	1A010	0000	001	901	蛋白定性[尿]
100040	1A0150000001271	1A015	0000	001	271	蛋白定量[尿]
100050	1A0150000004271	1A015	0000	004	271	蛋白定量[尿]
100060	1A0200000001901	1A020	0000	001	901	糖定性[尿]
100070	1A0250000001272	1A025	0000	001	272	糖定量[尿]
100080	1A0250000004272	1A025	0000	004	272	糖定量[尿]
100090	1A0300000001903	1A030	0000	001	903	比重[尿]
100100	1A0350000001901	1A035	0000	001	901	pH[尿]
100110	1A0400000001901	1A040	0000	001	901	ウロビリノーゲン定性[尿]
100120	1A0500000001901	1A050	0000	001	901	ウロビリリン定性[尿]
100130	1A0550000001901	1A055	0000	001	901	ビリルビン定性[尿]
100140	1A0600000001901	1A060	0000	001	901	ケトン体定性[尿]
100150	1A0750000001901	1A075	0000	001	901	白血球検査(試験紙)[尿]
100160	1A0850000001901	1A085	0000	001	901	食塩検査[尿]
100170	1A1000000001901	1A100	0000	001	901	潜血反応[尿]
100180	1A1050000001920	1A105	0000	001	920	沈渣[尿]
100190	1A1100000001920	1A110	0000	001	920	Bence Jones蛋白定性[尿]
100200	1A1150000001901	1A115	0000	001	901	フェニルケトン体[尿]
100210	1A1200000001271	1A120	0000	001	271	ヘモジデリン[尿]
100220	1A1250000001901	1A125	0000	001	901	インジカン反応[尿]
100230	1A1300000001901	1A130	0000	001	901	ジアゾ反応[尿]
100240	1B0100000015701	1B010	0000	015	701	虫卵(塗抹)[便]

電子カルテ EChart

○嶋 芳成 山本 樹

日本ダイナシステム株式会社

〒460-0007 愛知県名古屋市中区新栄二丁目 1-9 雲竜ビル東館 5F

Tel: 052-242-5441 Fax: 052-242-5984

E-mail: contact@jdynasys.co.jp

[1] はじめに

近年、診療録全体を電子化するシステム(以下、電子カルテ)に対する関心が高まっている。特に本年 4 月 22 日に厚生省より「診療録等の電子媒体による保存について」が通達されたことも、この傾向に拍車を掛けている。我々は、1994 年より、小笠診療所の小笠医師、株式会社リコーと協力して、電子カルテシステムを Echart を開発し、1995 年より小笠診療所の他3診療所で実稼働をしている。現在、そのシステムの改訂中であり、その内容について報告する。

[2] システム構成

従来のシステムは Windows 3.1 上で開発し、現在は Windows 95/98 においても稼働している。システム構成は次の通りである。

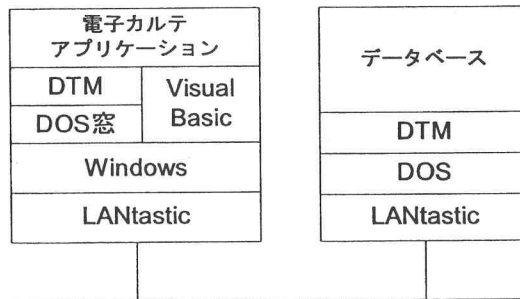
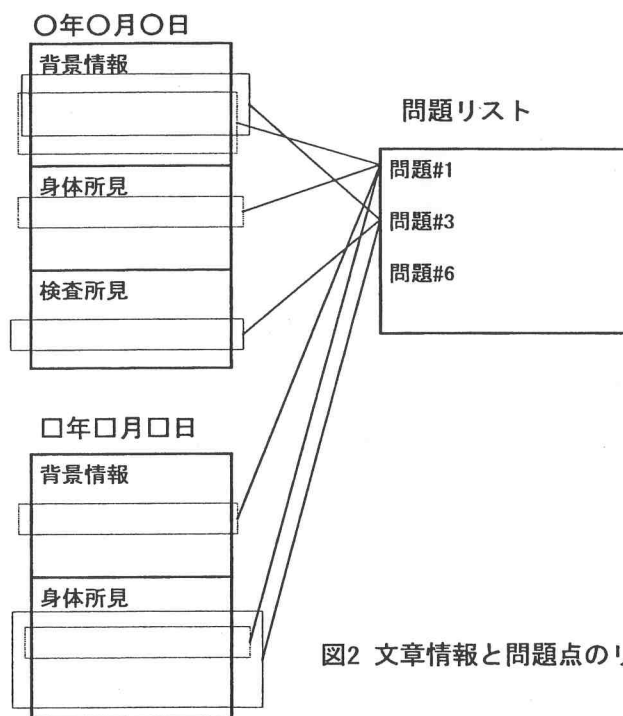


図1 従来のシステム構成

[3] 特徴

このシステムの特徴は次の通り。

- 1) 所見等の叙述的な情報(文章情報)を中心に扱う。
- 2) 入力支援のために、定型入力の機能を用意しているが、基本は文章情報である。
- 3) 文章情報を「背景情報」「身体所見」「検査所見」などの種類に分類することができる。(分類しなくてもよい。)
- 4) 文章情報を行単位で、問題(problem)とリンクすることができる。このとき同一行を複数の問題にリンクすることも可能。(図2参照)
- 5) 文章情報を表示する際、情報の種類、またはリンクされた問題で絞り込んで表示できる。
- 6) 問題の管理機能として、ある問題が別の問題に変化した場合に、その関係を保持できる。
- 7) 検査センターからのデータ取り込みと検査値の表参照機能を持つ。
- 8) 医事会計、指示入力機能はなく、他社の医事会計システムなどと合わせて運用する。



このシステムは、カルテの文章情報はどのように管理すべきか、という一点に焦点を当てて設計開発した。従って、医事会計、指示入力などの機能は他のシステムと組み合わせて利用することが前提であり、現在のところ画像を扱う機能もない。

一方、文章情報については、その種類と問題点のリンクという2つの軸で整理することができ、一種のハイパーテキストである。そして、この2つの軸をもとに、表示内容を絞り込むことができる。例えば高血圧症にリンクされた身体所見のみを表示して患者の変化を把握したり、投薬情報のみを表示して必要な投薬歴を探し出すのにも役立つ。

重要なことは、このハイパーテキスト機能は利用しなくてもよいという点である。利用しなければ、単に無構造な文章情報として格納される。日常診療の中で、特に管理の必要と思われる症例にのみ複雑なリンクを作るといった運用を想定している。

[4] 運用の結果

このシステムの運用の結果、2診療所においてはほぼ完全なペーパーレス化が実現できた。記録の真正性の保証や最終的なバックアップのため、情報は定期的に紙に印刷して保管している。しかし、これは日常診療には全く使用していない。また、画像データベースについては、他社のパッケージを組み合わせて利用しており、実用上特に問題にはなっていない。

カルテの記載内容を充実しようという動機が生まれ、より詳しいカルテができる傾向があり、また、特にハイパーテキスト機能を用いることで、患者の病像をダイナミックに捉えられるとの感想が寄せられている。

これらの運用結果については、医療情報学連合大会などで報告される予定である。

[5] 今後の見通し

従来のシステムは 16 ビットアプリケーションであるため、これを 32 ビットアプリケーションに移行する作業を現在すすめている。データベースエンジンは、DTM から Cache'(キャッシュ)バージョン 3.1 に変更し、それに伴って、ネットワークは Netbios ベースから TCP/IP ベースに変わる。この結果、処理速度や安定性が上がることが期待される。新システムの構成は次の通りです。

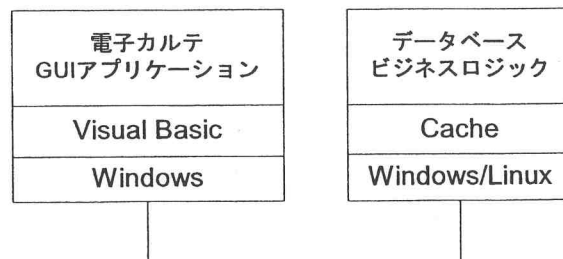


図3 新システムのシステム構成

この改訂に合わせ、ユーザーからの改良要望を検討中である。現在のところ、

- 1) 記録の改ざん防止を強化する。
- 2) 代行入力機能を追加する。
- 3) 情報の分類と問題指向システムの S、O、A、P の対応を拡張する。
- 4) 画像情報を直接扱えるようにする。
- 5) 操作方法を細かい点で改良する。

などの要望が上がっている。その他、外部との連携のために、MML とのインターフェースを装備することも検討している。

国際標準化の動向と電子カルテ

CPR system and International Standardization

里村 洋一

千葉大学医学部附属病院医療情報部

Yoichi SATOMURA MD, PH.D, Div. Medical Informatics, Chiba University Hospital

satomura@ho.chiba-u.ac.jp

1. ISO/TC215の成立背景

工業規格の標準化は元々アメリカの得意とするところであり、ヨーロッパはむしろこれに冷淡であった。ところが、EU統合を進める中で、あらゆる製造・販売業、サービス産業などにおける標準化が求められるようになり、急速に標準化の熱意が高まってきた。医療サービスもその例外ではない。CEN(Comite Europeen de Normalisation)はその標準化センターであり、TC251は医療情報の専門委員会として1990年に設立された。スタート直後の1992年にすでに、医療情報システムや電子カルテの基本構造、用語・分類、通信手順などをはじめとして64の作業目標が上げられていた。その後の6年間に具体的な成果が次々と報告されており、今やアメリカに対抗する有力なグローバルスタンダードの発信源である。

アメリカでは、民間や学会を中心に様々な分野でナショナルスタンダードを作ってきた歴史がある。前述のHL-7やDICOM、SNOMEDもその例である。一方、ゴア副大統領の指揮下でアメリカの国策として遂行されているNII(National Information Infrastructure)計画では、アメリカの新しい社会構造を作るための情報基盤を整えるとしている。医療はその骨格の一つであって、最も重点の置かれている対象のひとつである。このような背景で、アメリカの公式な標準化機構であるANSI(American National Standard Institute:日本のJISに相当)が医療情報の標準化活動を取りまとめることになった。ANSIにHIPPS(Health Information Standard Planning Panel)が設けられたのが1992年で、その後に改組されてHISB(Health Information Standard Board)となったが、民間で行われている標準化活動を支援し、相互の調整を図るための働きをしてきた。連邦政府はもう一方で、これらの標準化された医療情報を利用して難渋するアメリカの医療問題、特に医療費の増大に対応しようとしている。そのための法律が1996年に成立したHIPPA(Health Information Portability and Accountability Act)である。この法律では2000年をスタート時点と定め、医療機関や保険関係団体に、診療とその費用に関する情報を連邦政府に報告する事を義務づけたもので、報告は電子化され標準仕様に従ってなされなければならないとしている。罰金の条項までであるこの法律の施行は、否が応でも医療機関に情報の電子化とその標準化を迫るものである。

2. ISO/TC215の発足とその目標

1996年、ANSIはISO(International Standard Organization)の主要なメンバー国に呼びかけて、医療情報標準化への取り組みを提案した。ヨーロッパ諸国、カナダ、オーストラリア、日本などが応答し、1997年末までに4回の準備会議(その内1回は京都で開催)を重ねた後、1998年1月にISOのTMB(理事会)がTC215(医療情報)の設立を決定し、同年8月にフロリダ州オーランドで第1回ISO/TC215(Technical Committee for Medical Informatics)が開催されるに至った。

この会議には18カ国が参加したが、TC215は、19の“P”メンバー(決議権のあるメンバー)と10の“O”メンバー(オブザーバー)より構成される事になった。

議長国はオーストラリア、事務局はアメリカが務めることし、以下の四つのワーキンググループと二つのアドホックグループが作られた。

WG1: Health records and modeling coordination (医療情報のモデル)

WG2: Messaging and communication (メッセージ交換:通信)

WG3: Health concept representation (医学用語・分類)

WG4: Security (安全性)

WG2と4については、目標が明確で候補になる得る技術や仕様(たとえばHL7)が存在する。これらはかなり早くまとまりが付き、ISOの規格として登場する事が予想されている。WG1の目標はかなり抽象的であり、また各国の医療事情にも依存するので、医療情報の包括的なモデルに関する国際的な合意はなかなか困難と思われる。WG3は言語の障壁を越えて共通概念を完全に描ききれるか学術的にも興味ある課題に挑戦する。二つのAdhoc委員会は、それぞれ医療画像と健康カードについて、作業目標に取り入れるか否かを検討して、99年4月にベルリンで開催される予定の第2回会議で報告する事とした。

第2回の会議は99年4月12日から15日にかけて、ベルリンのDINにおいて開催された。この会議では、各WGのワークアイテム(作業目標)が討議された。

WG1は「すべての医療情報に横断的なモデルを作成する」という包括的な目標を掲げたものの、実際にこれから手がける標準化の作業は、救急医療データセットや、患者や医療職の識別情報の共通規格を検討するという現実的なステップを提案した。

WG2はHL7の様な実在のシステムを組上に乗せると考えられていたが、ヨーロッパのEDIFACTとの折り合いがついていないようで、現実的な目標を医療機器間通信の規格やXMLでの階層的記述の手法といった、小さな課題に絞ってスタートすることとなった。

WG3は、当初から遠大な計画を掲げていたが、具体的目標を定めた今回の会議でも絞り込みは困難であった。言語を論じる限りONTOLOGY(存在学)的検討が必要だとするドイツの代表者らの意見も無視できない状況にある。今回は、とりあえずメタボキャプラリー(言語システムを表現するための用語)の検討と用語集の備えるべき基本要件など

を当面の目標とする事になった。

WG4は、WG内の意見集約が遅れ、この会では公開鍵（暗号解読のための）の規格をワークアイテムとしたが、次回（東京）、までに他のセキュリティー関係の作業目標を整理することとした。

懸案だった医療画像の通信規格については、アドホックグループで討論した結果、WGを新たにつくることはしないと結論づけた。この背景には、事実上の国際規格となっているDICOMがISOへの取込みを好んでいないという事情がある。長年にわたって参加国の合意づくりをし、やっと結論にたどり着いた段階で、改めてISOの場で議論を繰り返すのは愚であると考えているようだ。これと同じような立場にあるHL7が積極的なISOへの協力姿勢をとっているのとは対照的である。もう一つのアドホックグループ「健康カード」は5番目のWGとして正式に設立されることとなった。とりあえず、カードのサイズに関する規格に取り組む予定で、カードのコンテンツや記憶方法などについては次のTCまでに検討してワークアイテムに取り上げるかどうかを決定する予定であるという。対象領域によって程度の差は有るものの、これからの医療情報の標準化はISOの動きを中心として展開されるものと思われる。国内では、厚生・産産両省の協力の下で、ISO/TC215国内対策委員会が組織され、ISOで協議される事項について国内の意見のとりまとめが行われている。(財)医療情報システム開発センターが事務局をつとめ、両省の他、関連学会、医師会、歯科医師会、企業団体などが参加している。

3. WG3における検討

WG3は第一回目の会合を98年11月AMIAのシンポジウムに合わせて、フロリダ州オーランドで開催した。この時は、メンバー国への通知から3ヶ月を経過していないことを理由に非公式とされた。アメリカ、イギリス、イタリア、日本の4カ国から9名が参加したが、コンベンナーのAnn. Harding氏が出席しなかったこともあり、99年4月のTC215へ向けての準備が中心となった。

1999年4月12日、ベルリンDINにおいてWG3の正式な会合が初めて行われ、13カ国（うち2カ国はオブザーバー）が参加した。前もって、作業目標（work item）の提案が、イタリア、イギリス、アメリカ、オランダの4カ国から出されており、これらの提案を対象として討論が行われた。提案された内容は以下のようなものである。

1) メタ・ボキャボラリーの開発 (Meta-vocabulary : イタリアの提案)

医療におけるコンセプトや用語、分類などを論じる際に使われる用語の整理と定義。

2) 用語集の基本要件 (Foundation of Terminology : イギリスの提案)

ISOが採用する標準の用語集が備えているべき要件を整理すること。

3) 電子カルテのための制限言語の整備 (controlled Vocabulary) (アメリカの提案)

すでにアメリカやEUで開発がなされている電子カルテのための用語集を国際的なものにする方法論

4) 用語の意味的関連と構造 (Semantic links and concepts : オランダの提案)

1) の **meta-vocabulary** については、かねてからイタリアの Rossi Mori 氏が主張しているところで、今日、用語集を意味する言葉の一つ取り上げても、**terminology, nomenclature, lexicon, thesaurus, dictionary, nosology** など類似語が数多くあり、それらの定義が定かでない状況であり、討論の混乱する原因となっていることから、先ず基礎を固めると同時に問題点をクリアーにしようとするものである。大きな異論なく、この作業目標については、Rossi Mori がリーダーとなり、日本では里村 (千葉大) が担当する事とした。

2) の **foundation** については、医療情報のモデルとも関連するので、WG1 との連携が重要であるとの意見が述べられ、イギリス (担当 Alan Recter 氏) が中心となって今後の検討を進めることとした。WG3 では、具体的な用語集の内容 (用語そのもの) に付いては作業を行わないこととしており、**foundation** についての具体的な作業は、用語集の構造、開発、機能、実装、使用、配布、評価及び維持 (編集環境、改訂、管理) などについての要件をまとめることである。日本では石川 (国立がんセンター) が担当することとした。

3) の **controlled vocabulary** は、既に ASTM で開発済みのものがあり、これをベースに検討してはどうかとの提案がアメリカからなされた。一方、NHS でも同様な作業がおこなわれており、6月に結果が公表されるとイギリスから紹介があった。

4) の **semantic link** は目標がやや曖昧であること、提案者のオランダが出席しなかったこと、また、2) の中でおおよそは検討される可能性があることなどから、具体的な作業目標としてはサスペンドとなった。(日本の担当 津本 (島根医大))

4. 電子カルテと標準化

ISOで進められている電子カルテのモデル化は、国際的に通用する医療情報のモデルを作り出そうとする壮大なもので、国によって医療制度、経済力、医療技術や民間の医療知識が様々に異なる状況で、簡単に結論を得ることは困難と思われる。また、各国が情報システムの普及に大わらわの状況で、無理矢理に国際標準に合わせた開発が適切であるかどうかとも疑わしい。むしろ、電子カルテは各国や地方の環境に合わせた形で開発を進め、国際的な電子カルテの通信が必要になってくれば、メッセージ交換の規格に従って、可能な限り自国のカルテから抽出したデータを提供するというのが現実的な方法であろう。

但し、この時点で、既に国際的な標準が **de-facto** であろうと存在する場合には、これのできる限り尊重し、取り入れた開発すべきものと考えられる。

この意味で、開発中の医療用語集も SNOMED や READ といった先行の用語集と整合性を求めて整備すべきものとする。

電子保存通達と電子カルテ

山本 和子

島根医科大学医学部医学科医療情報学

〒693-8501 島根県出雲市塩冶町 89-1

yam@shimane-med.ac.jp

1. はじめに

これまで紙媒体による診療録の保存が義務づけられていた。記載方法は手書きとされていた。病院にコンピュータが導入され、医事会計以外に診療現場で利用されるに従い、コンピュータへの入力と手書きとの二重記録の不便さが指摘される。やがて、1988年にワードプロセッサ等により印字された文書も認められる。その後、電子カルテが開発されると、紙カルテと電子カルテの二重構造の不便さが指摘されていた。今回、1999年4月22日に厚生省から診療録および診療に関する諸記録の電子保存に関する通達が出された。これはまさに画期的な出来事であり、多くの病院が一気に電子カルテに進むムードが醸し出され、熱気に包まれている。

2. 電子保存通達について

厚生省の電子保存通達の詳細については医療情報学誌¹⁾に掲載されている。その骨子は、真正性、見読性、保存性の確保であり、そのために運用管理規定を定め、保存されている情報の証拠能力・証明力をつけ、患者のプライバシーに留意することとある。但し、これらはすべて自己責任において行うこと、自己責任とは電子保存システムの説明責任、管理責任、結果責任を果たすことであるという。

3. 電子保存通達を遵守した電子カルテとは？

厚生省の電子保存通達はなかなか含蓄のある内容である。以下に電子保存通達の内容を紹介し、具体的にどのようにすれば法的に認められることになるかを検討し、疑問点等を記述し考察する。

A. 真正性の確保について

A-1. 作成の責任の所在の明確化

作成の責任の所在の明確化として、1) 作成責任者の識別及び認証、2) 確定操作、3) 識別情報の記録、4) 更新履歴の保存、が上げられている。

このうち、1), 3), 4) は現在の病院情報システムでも実施されている。ただ、3), 4) で記録されているものを事件が発生した時に、システム管理者が確認し、証明書と証拠書類を発行すれば良いのか、第三者が読める形でプリントする機能が付加する必要があるかどうかである。

残りの2) であるが、利用者の意識の問題もあり、「医療」が「記録」よりも優先されるという現実の中で、正確に実現させることは難しい。例えば、現在使用している「登録」キーの他に「確認」キーを追加することは、作業量が増加する点で現場では受け入れられないであろう。そ

れでは、作成責任者が入力した場合、「登録」＝「確定」とみなしていいのかどうか、現在の「登録」キーを「確定」キーに名称変更するのはどうであろうか。代行入力の場合は作成責任者による「確定」操作が必要であるように通達には書かれているが、作成責任者がはたして「確定」操作を実行するであろうか。そのようなことは現実的に不可能に近いように思える。作成責任者が代行入力者に全委任状を書いておけば、それは認められるのか。そうなると代行入力者は無責任に入力する可能性が増加するのではなからうか。そもそも作成責任者とは誰なのか。大学病院には指示医と入力医がある。しかし、出張中の医師から指示があつて別の医師が代行入力した場合、責任はどうなるのであろうか。指示医と入力医の記録は捨てるわけには行かない。また、医事課職員がレセプト作成時に代行入力した場合はどうであろうか。等々「確定」にはいくつかの問題点が存在している。

A-2. 虚偽入力、書き換え・消去及び混同の防止

これには、1) 過失によるもの、2) 使用する機器、ソフトウェアに起因するもの、3) 故意によるもの、があると通達に記入されている。これらを防止することは比較的实现可能で、現病院情報システムでも実施している。

B. 見読性の確保

見読性の確保として、1) 情報の所在管理、2) 見読化手段の管理、3) 情報区分管理、4) システム運用管理、5) 利用者管理、が上げられている。管理として記入されている内容については比較的实现可能と思われる。現病院情報システムでも同様な管理がされているので、どのように管理しているかを明記しておけばいいのではないかと解釈している。

ただ、「見読性」とは何かについて通達には明記されていない。即ち、情報の開示が求められた時に、紙に印刷しなければいけないのか、ディスプレイ画面に表示できればいいのか。画像などは紙に印刷しても意味をなさないから磁気媒体にコピーしなければならないのか。線画等の混在した文書を紙に印刷させるソフトウェアをわざわざ作成するのも高価な費用が必要である。ディスプレイ画面表示で可としたい。

なお、どのように記載されていれば診療記録と見なされるのか、この点も不明である。「紙カルテと全く同じ様式で表示されたもの」を電子カルテとみなす考え方もあるが、紙文化ではない情報文化としての電子カルテへと今後発展すると予想されるので、明記されないことを感謝しつつ、新しい電子カルテの創造へ挑戦したい。

とりあえずは、あまり常識を逸脱しない形でのディスプレイ画面への表示と簡単なプリント機能(画像はプリントしない)を追加することを考えている。

C. 保存性の確保

保存性の確保として、1) 媒体の劣化対策、2) ソフトウェア・機器・媒体の管理、3) 継続性の確保、4) 情報保護機能、があげられている。これらは費用さえかければ比較的实现可能である。病院情報システムで発生した医療情報は電子カルテデータベースへ転送し改竄不可とし、データベースを二重化し、且つバックアップできるシステム構成を考えればいい。

図1に電子保存通達に準拠した形での本学における医療情報ネットワークシステムの構成図を

示している。骨子は、改竄不可の電子カルテデータベースを構築したこと。電子カルテデータベースの二重化を図ったこと。災害時に備えて設置場所も別にしたこと。且つバックアップもとるようにしたこと。院内での利用とともに地域医療機関との情報の共有を考えていること。簡単な電子カルテのビューと印刷機能を持たせたこと。などである。なお、いずれ電子カルテサーバの故障時の対策用にサーバ機の増設を予定している。通達を満足させるためにはかなりの補強が必要である。

なお、改竄不可の電子カルテデータベースを構築する場合、処方、検査成績、経過記録、退院時要約などは実現可能であるが、患者の基本情報や病名など日々更新されている情報は改竄不可のデータベースへ転送することは難しい。「確定」とすれば改竄不可とできる情報と更新記録を残すことで良しとしなければならない情報とを区分し、明記しておく必要がある。本学で使用している情報を分類すると以下ようになる。

- ①はじめに速報的な情報があり最終的に「確定」されるもの
検査成績、検査報告書（病理診断報告書、画像診断報告書）
 - ②「確定」操作が明確であるが、「確定」後に修正が発生する可能性があるもの
退院時記録
 - ③日時で自動確定し、その後に修正が発生する可能性があるもの
経過記録
 - ④トランザクション情報であるが、ある時点が過ぎれば修正されることはないもの
オーダ情報、予約情報
 - ⑤日々変更されるもの
患者基本情報、病名、感染症・アレルギー
- 退院時記録と経過記録の確定後の修正は、「追加」として修正情報を記録することにしている。
- ④と⑤は更新記録は残しているが、電子カルテデータベースには転送していない。

D. 相互利用性

これに関しては医療情報の標準化が世界規模で進められている。

E. 運用管理規定とプライバシー保護

厚生省通達に運用管理規定が例示されている。これに準拠したものを作成し、プライバシー保護意識の徹底をはかるように努力すれば良いであろう。

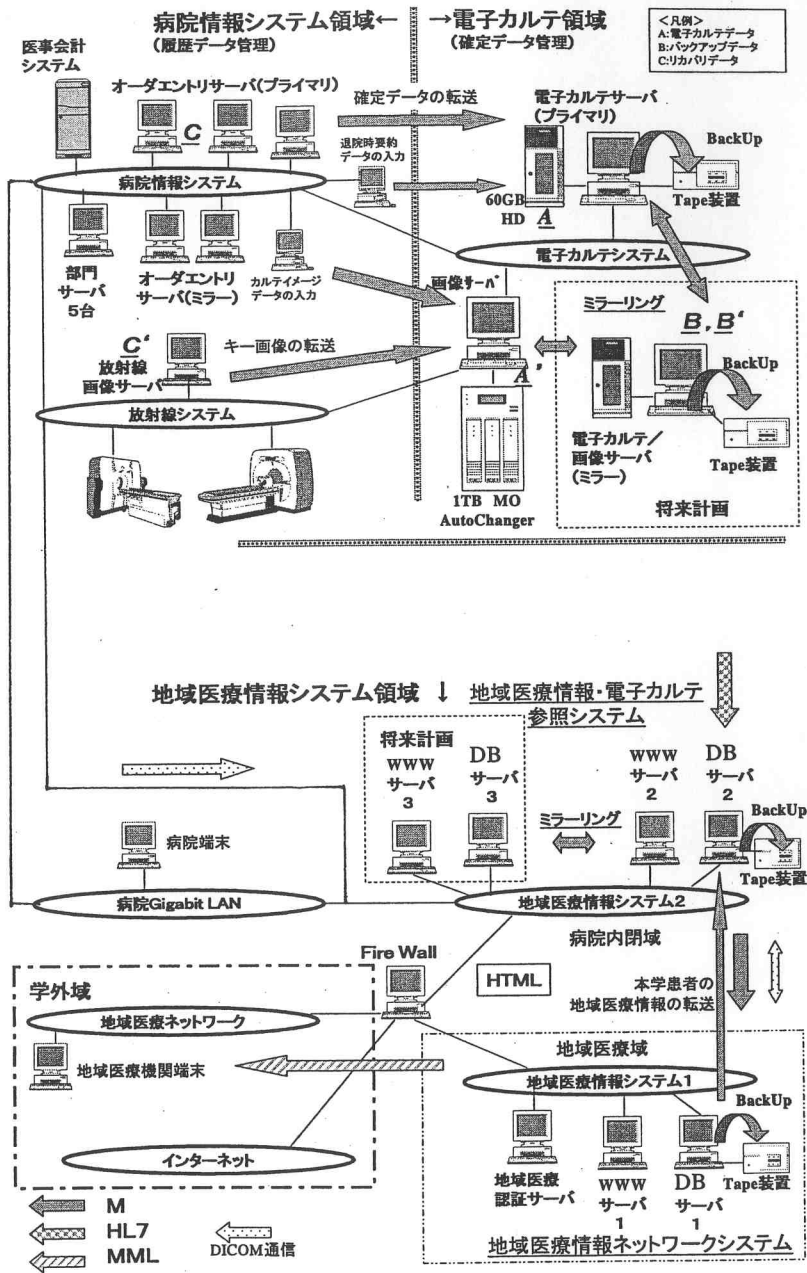
4. おわりに

厚生省の電子保存通達を解説し、電子カルテシステムをどのように構築すれば良いかを検討し問題点を提起した。これを契機に新しい電子カルテが創造されることを期待したい。

参考文献

- 1) 松本義幸：診療録および診療に関する諸記録の電子保存について，医療情報学，Vol.19
Suppl. May 1999

図1. 電子カルテのシステム構成



C 言語を用いた Cache' から DLL サブルーチンを呼び出す M 関数の作成

○岡田好一、大櫛陽一、大久保裕和*

東海大学医学部、*株式会社ケンウッド

〒259-1143 神奈川県伊勢原市下糟屋 143
TEL: 0463-93-1121 (2145) fax: 0463-93-5418
yoshikaz@is.icc.u-tokai.ac.jp

1. はじめに

M 言語から他言語のルーチンを呼び出す方法は、M の実装により異なる。InterSystems 社の Cache' では OS のプロセスを起動する方法と、Cache' 自身にリンクした \$ZF 関数を呼び出す方法とがある。

現在はパソコンを含めてマルチタスク OS が一般的であり、安全性を考慮すれば、M 以外の仕事に専念する別プロセスを生成し、名前付きパイプなどのプロセス間通信で Cache' と情報をやりとりするのが安全である。実際、Windows において Cache' がユーザーインターフェースに VisualBasic や Delphi を使う場合は OCX を用いて IP によるプロセス間通信が行われる。しかしながら、C 言語などを用いた一般のユーティリティに対して用意されているシステムルーチン(%CLI)ではプロセス間通信が自動生成されるわけではない。そのため、名前付きパイプは Open 命令で容易に生成できるが、接続のタイミング処理がやっかいとなる。

\$ZF 関数は Cache' の機械語プログラム自身に C 言語等で開発されたプログラムをリンクしてしまう古典的方法でルーチンを呼び出すしかけである。今回、暗号化復号化ルーチンが Windows のダイナミックリンクライブラリ(DLL)で提供されていたため、\$ZF を利用することとした。\$ZF 関数を利用すれば、開発が楽であり、また、高速動作が期待できる。反面、リンクしたルーチンが信頼に欠ける場合は、同じプロセス内の Cache' 本体に影響し、最悪の場合本体が破壊される可能性がある。本件の場合、DLL が信頼できること、C 言語で新たに開発した部分が非常に小さいことから \$ZF を採用した。

2. 方法

2.1. コンパイルとリンク

マニュアルの記述に従い、以下の作業を行う

(1) Microsoft C++ 6.0 等を導入する

その際、環境変数等、コマンドラインを受け付ける設定をしておく

(2) 開発は d:\¥Cachesys¥bin¥ で直接行うので、必要ならバックアップする

以下のファイルをコピーする

drans. c	C のプログラム本体
czf. c	オリジナルの czf. c を改造したもの
fdkcrypt. h	DRANS 開発キットに含まれるヘッダファイル
fdkctypt. lib	DRANS 開発キットに含まれるリンクライブラリ
clink. bat	リンク用のバッチファイルを改造したもの

(3) コンパイル(コマンドライン)

```
cl -c czf. c
```

-c はコンパイルのみのスイッチ。czf. obj ができる。(オリジナルの czf. obj は不要となる)

(4) リンク(コマンドライン)

```
clink( bat )
```

暗号対応の cache. exe ができる。

2.2. 暗号対応 cache. exe の使い方

(1) Cache. exe を d:\¥Cachesys¥bin¥ にコピーしてスタート

(2) Fdkcrypt. dll を d:\¥WINNT¥system32¥ にコピーする

(3) ネームスペース %SYS にルーチン %DRANS をリストアする

Cache のリストアツールで、Drans. rsa を読み込む

2.3. 関数の使い方

暗号化: s y=\$\$\$~%DRANS(x, 1)

復号化: s z=\$\$\$~%DRANS(y, 2)

3. 結果

メモリが十分にある PentiumII 300MHz のパソコンの場合、暗号化復号化ともに 1 レコード約 1 ミリ秒であった。単純なグローバルセットに比べ、暗号化を組み合わせるとセット時間が約 2 倍になる。

当初の目標を満たしているため、実用可能と判断した。

4. 考察

暗号化・復号化ルーチンが DLL で供給されており、信頼できること、リンクする C 言語のルーチンが短いことから \$ZF 関数を採用した。速度が速く、当初の目的は達成したと考える。

\$ZF の場合、改造した cache.exe を使うことになる。少量とはいえリンクしたルーチンが M 言語本体と同じプロセス内で動作することになり、別プロセスの場合に比べるとルーチン同士の干渉の可能性が高くなる。

プロセス間通信はネットワーク間でも動作するなど自由度が高い。そのぶん、時間的な不利があると考えられる。今回の場合、暗号化復号化の単位はレコードであり、数十バイトのケースが多いと考えられる。そのため、速度的な観点から \$ZF の使用が適当と考える。

最初、DLL をコンパイル時リンクではなく実行時リンクしようとしたがうまくゆかなかった。リンクのタイミングは問題にならないこともあるが、今回の場合は DLL の性格によると考えられた。

本ルーチンは暗号化されたデータベースを実現するために作成した。理想的には M 本体に暗号化の仕掛けが組み込まれていることが望ましい。現在、実現方法を検討中であり、M そのものへの暗号システムの組み込みを検討している。

暗号化されたデータベースは内部犯行やファイヤーウォールを突破したハッカーからデータを守ることができる。現時点では、速度の観点から暗号化データベースは実用されていない。M 言語と本稿で使用した暗号化復号化ルーチンの組み合わせで実用的な暗号化データベースが実現できると考える。

● ソースプログラム

```
[drans.c]
#include <windows.h>
#include <string.h>
#include "Fdkcrypt.h"

int drans(int j, ZARRAYP in, ZARRAYP out)
{
    unsigned long vv;
    switch (j) {
    case 1:
        (void)FssEncryptKey8Data("anykey01", in->len,
```



```

    in->data, out->data, &vv);
    break;
case 2:
    (void)FssDecryptKey8Data("anykey01", in->len,
        in->data, out->data, 1, vv);
    out->len = in->len;
    break;
}
return 0;
}

```

[czf.c]

```

#include    "cdzf.h"
#include    <stdio.h>
#include    <io.h>
#include    <string.h>

extern    int    test();
extern    int    catbytes();
extern    int    openfd();
extern    int    closefd();
extern    int    causeerr();
| extern    int    fssload();    }    }00
| extern    int    fssfree();
extern    int    zferro();

| #include    "drans.c"    }    }00

ZFBEGIN
ZFENTRY("TEST", "ipcfidIPCFD", test)
ZFENTRY("CATBYTES", "bB", catbytes)
ZFENTRY("OPENFD", "cP", openfd)
ZFENTRY("CLOSEFD", "i", closefd)
| ZFENTRY("DRANS", "ibB", drans)    ←    }00
ZFENTRY("ERROR", "ii", causeerr)

```

ZFEND

/* 以下、オリジナルと同じなので省略 */

```
[clink.bat]
echo off
if "%1"==" " goto notcallin
if "%2"==" " goto noname
cl %1 shdir.obj cache.obj czf.obj -o %2 advapi32.lib libc.lib
kernel32.lib user32.lib wsock32.lib winmm.lib lz32.lib
netapi32.lib winspool.lib fdkcrypt.lib
goto end
:noname
cl %1 shdir.obj cache.obj czf.obj -o ccallin advapi32.lib
libc.lib kernel32.lib user32.lib wsock32.lib winmm.lib
lz32.lib netapi32.lib winspool.lib fdkcrypt.lib
goto end
:notcallin
cl main.obj shdir.obj cache.obj czf.obj -o cache advapi32.lib
libc.lib kernel32.lib user32.lib wsock32.lib winmm.lib
lz32.lib netapi32.lib winspool.lib fdkcrypt.lib
:end
```

```
[drans.rsa]
Cache for Windows NT^INT^
%RO on 13 Oct 98 8:58 PM
%DRANS.MAC..57629,75264.
%DRANS(X,J) ;;981012;TRANSLATE STRING
; USAGE S Y=$$^%DRANS("STRING",J)
; J=1 ENCODE, J=2 DECODE
N (X,J)
START ;
I J=2 G DEC
I J=1 G ENC
Q X
DEC ;
S X=$ZF("DRANS",2,X,X)
```

```

S Z="",N=0,SI=$C(15),SO=$C(14)
S MOD=0 ; 0-255 MODE
LOOP1 S N=N+1,W1=$E(X,N)
I W1="" G EXIT1
I W1=SO S MOD=1 G LOOP1
I W1=SI S MOD=0 G LOOP1
I MOD=0 S Z=Z_W1 G LOOP1
I MOD=1 S Z=Z_$C($A(W1)*256+$A($E(X,N+1))),N=N+1 G LOOP1
EXIT1 QUIT Z
ENC ;
S Z="",N=0,SI=$C(15),SO=$C(14)
S MOD=0 ; 0-255 MODE
LOOP2 S N=N+1,W=$E(X,N) I W="" G EXIT2
S WA=$A(W) I WA<256 S:MOD=1 Z=Z_SI,MOD=0 S Z=Z_W G LOOP2
S WA1=WA¥256,WA2=WA#256
I MOD=0 S Z=Z_SO,MOD=1
S Z=Z_$C(WA1)_$C(WA2)
G LOOP2
EXIT2 I MOD=1 S Z=Z_SI
I $L(Z)<8 S Z=$E(Z_$C(15,15,15,15,15,15,15,15),1,8)
QUIT $ZF("DRANS",1,Z,Z)

```

開発支援ツール VBコード・ジェネレータとドキュメント・ジェネレータの紹介

広瀬清司

有限会社 エム ブイ ビイ

〒487-0033 愛知県春日井市岩成台8丁目3番地の6 504棟503号

TEL:0902-5794-286

YQT03304@nifty.ne.jp

1. はじめに

・システム開発における2つの工数（量的工数と質的工数）

現在のコンピュータシステムがカバーすべき業務範囲は拡大を続けており、システム開発にかかる量的工数は膨大なものとなっている。また、システム利用者からの要求レベルはますます高度なものとなり、システム開発にかかる質的工数も膨大なものとなっている。しかしながらシステム開発の予算工数は削減される方向である。つまり、以前にもまして短納期・高品質なシステム開発が求められている。

このような要求側（システム利用者）の予算工数と提供側（システム開発者）の必要工数のギャップを埋めていくためのキーワードとして下記の項目があげられる。

- ・信頼性の確保
- ・保守性の確保
- ・生産性の向上

これらのキーワードからコンピュータシステム、とりわけ業務処理プログラムに共通して求められる事は「いかにして既存の（動作が保証されている）プログラム・コードを利用していか」、「いかにして共通化されたプログラム・コード（ロジック）を利用していか」という事項である。

つまり、「理想とされる業務プログラム開発＝共通化された部品の利用＋動作が保証された部品の利用＋技術レベルの低い技術者でも容易に理解できるロジックの利用」という考え方が不可欠となっている。

また、システムの保守性向上を考えた場合、システム構築に関するドキュメントの作成（整理整頓を含めて）が重要な課題となる。

このような状況を踏まえて、システム開発に関わる問題解決を図る目的で2つのジェネレータ・システムの開発を行った。

(1) VisualBasic コード・ジェネレータ

（動作が保証されている VisualBasic プログラムコードを部品登録しておき、その部品の再利用を行うためのジェネレータ・システム）

(2) ドキュメント・ジェネレータ

（システム開発時のドキュメント作成工数を削減するためのジェネレータ・システム）

以下に今回開発したジェネレータ・システムの概要・構成、実際の使用感、および今後の課題について報告する。

2. ジェネレータ・システムの概要

今回開発した2つのジェネレータ・システムについて項を分けて報告する。

2.1 VisualBasic コード・ジェネレータ

VisualBasic コード・ジェネレータは、2つのサブシステムから構成されている。

(1) VisualBasic プログラム・コードをコード・ライブラリへの登録するサブシステム

・コード・ライブラリとコード・カテゴリ

今回開発した VisualBasic コード・ジェネレータの新しい概念として「コード・ライブラリ」がある。「コード・ライブラリ」とは、すでに動作が確認されているプログラムの中から、他での使用が見込めるプログラム・コードを登録管理しておくデータベースをしめす。

このコード・ライブラリの特徴は、システム開発担当者がプログラム・コードを自由にコード・ライブラリに登録することが可能なことである。自由にコード・ライブラリを更新できることにより、各担当者が必要と考えられるコードを登録してコード・ライブラリを成長させていくことができる。

この方式の問題点として考えられることは、各担当者が自由にコード・ライブラリにコードを登録できるために、コード・ライブラリが肥大化してしまうことが考えられる。このような問題点を回避する目的で「コード・カテゴリ」という概念を導入している。「コード・カテゴリ」とは、プログラムコードの機能的な集合体をしめす。例えば「データベース更新」というコード・カテゴリには「データ追加」「データ修正」「データ削除」というデータベースを更新する際に必要と考えられるプログラム・コードを1つの機能的集合体としてとらえることができる。

・キーワード変換

コード・ライブラリで管理されるプログラム・コードは、動作確認がされている VisualBasic プログラム・コードを VisualBasic 開発環境からコピー&ペーストして登録作業を行う。

VisualBasic プログラムでは、画面上のオブジェクト (TextBox, Listbox, etc) をプログラム内で指定する場合、「フォーム ID. オブジェクト ID」の形式で指定することが望ましい。このように記述されている VisualBasic プログラム・コードをそのままコード・ライブラリに登録して再利用すると、再利用する場合に指定すべきフォーム ID が正しく反映されないという問題点がある。そこで、コード・ライブラリにコード登録をする場合に、「@form@」というキーワードを利用することを可能とした。VisualBasic コード・ジェネレータがキーワード「@form@」を発見すると、コード・ライブラリの内容を反映させるフォーム ID に変換し、プログラム上の矛盾が発生しないように編集する。

同様に、オブジェクト ID を反映させるために「@object@」というキーワードを利用することを可能とした。

(2) VisualBasic コード生成サブシステム

コード・ライブラリに登録されたプログラム・コードを使用して VisualBasic プログラムを生成する場合には、「どの VisualBasic プログラムに対して生成処理を実施するか」「どのコード・カテ

ゴリを使用するか」を指定する。

プログラム生成を実施する対象としては、VisualBasic 開発環境で作成したファイル (*.frm) を指定する。また、使用するコード・カテゴリについては、画面上に表示されるコード・カテゴリの中から任意に選択することを可能とした。

以上の2点について指定した後、「処理開始」ボタンを押下してプログラム生成処理を実施する。

生成処理が完了したプログラム・ファイルを VisualBasic 開発環境の中で呼び出して使用することにより、生成されたプログラム・コードを利用することができる。生成後のプログラム・コードは自由に変更することが可能である。

VisualBasic コード生成処理は、新規にプログラム・コードを生成する方法と、既存のプログラム・ファイルに対してコード・ライブラリのコードを追加する方法を装備している。

2.2 ドキュメント・ジェネレータ

ドキュメント・ジェネレータは、管理用データベースにデータベース仕様情報を登録（更新）する機能と、登録されているデータベース仕様情報を MS-WORD 文書ファイルにフォーマットを整えて出力する機能を装備している。

(1) データベース仕様情報登録機能

データベース仕様情報登録機能では、データベース名称、グローバル変数名、キー（添え字）構成情報、データ構成情報（区切り記号の値と各値の内容）を管理用データベースに登録する機能を提供する。

管理用データベースは、指定されたデータベース名称をキーとして、各データベース仕様情報を管理する。データベース仕様情報を修正する場合には、データベース名称を指定して検索する必要があるが、この場合にデータベース名称を完全に指定するのではなく、データベース名称の一部を指定して検索することができる。

データベース仕様情報登録画面操作における生産性向上を目的として、データ属性値（文字列、数値、日付、等）に関しては画面上で指定された値を選択値用データベースに登録しておき、次の画面操作からは選択値用データベースの値を一覧表示し、その中から選択できる機能を提供している。選択値をリスト表示する場合に、使用回数の多い値がリストの上方に位置する仕組みとしている。

選択値の選択操作はマウスでの操作を基本としているが、キーボードのみの操作も可能としているので、大量のデータを登録する場合の生産性向上を図っている。

(2) データベース仕様情報出力機能

データベース仕様情報出力機能は、MS-WORD 文書ファイルを使用して実現している。

具体的には、専用の MS-WORD 文書ファイル (*.doc) の中に、マクロと VBA を用いて処理プログラムを作成し情報出力を実施する仕組みとしている。

マクロは、データベース仕様書のフォーマットを整える機能を分担している。

VBA は、管理用データベースからデータベース仕様情報を抽出する機能を分担している。具体

的な手法は、VisualBasic から Cache'データベースにアクセスする方法 (VISM.OCX を使用する
方法) と同様である。MS-WORD 文書ファイル内に VBA で画面を作成し、VISM.OCX (MVB.OCX)
を埋め込み Cache'データベースにアクセスする、という手法である。

データベース仕様書は、利用者が指定した分のデータベース仕様情報を一括して作成することが
できる (この作成処理単位で MS-WORD 文書ファイルが作成される)。

MS-WORD 文書ファイルに出力するデータベース仕様情報は、処理画面上でデータベース名称
を指定することにより実施する。この場合、例えば患者情報に関するデータを全て出力対象とする
場合には、データベース名称に「患者」と指定して検索ボタンを押下することにより、データベー
ス名称に「患者」が含まれるデータベース名称がすべて表示される。この中からさらに取捨選択し、
実際に出力するデータベース情報を確定することができる。

必要となるデータベース名称を指定してデータベース仕様書作成処理を実施した時に作成され
る MS-WORD 文書ファイルは、通常の MS-WORD 文書ファイルであるので、自由に編集して印刷
することが可能である。

3. ジェネレータ・システムのシステム構成

今回開発した2つのジェネレータ・システムは、クライアント/サーバー構成を前提として構築し
ている。ただし、各開発担当者が個々の開発環境に応じてジェネレータ・システムの活用を図ること
が想定されることからスタンドアロン構成の動作も確認している。

クライアント/サーバー構成とスタンドアロン構成の違いは、Cache'データベース・サーバー
(VisualBasic コード・ライブラリ、データベース仕様書情報を管理) がどこに存在するかの違いに
よるものである。Cache'データベース・サーバーはバージョン 2.3 での動作確認をしている。

OSは、サーバーOSとして WindowsNT4.0 SERVER, Windows95, Windows98 での動作を確認し
ている。クライアントOSとしては、Windows95, Windows98 での動作を確認している。

アプリケーション・ソフトについて、VisualBasic コード・ジェネレータは VisualBasic5.0J を使
用して作成している、ドキュメント・ジェネレータのデータベース仕様書登録機能は VisualBasic5.0J
を使用して作成している、そして、データベース仕様書作成機能は、MS-WORD97 のマクロ、VBA
と VISM.OCX (MVB.OCX) を組み合わせて構成している。(VisualBasic、MS-WORD について、
最新バージョンでの動作確認は実施していない。)

4. ジェネレータ・システムの利用状況 (使用感)

今回開発したジェネレータ・システムの利用状況や実際の使用感を以下に報告する。

(1) VisualBasic コード・ジェネレータ

VisualBasic コード・ジェネレータを使用することにより、通常の開発 (全てのプログラムコー
ドを作成する方法) に比べて、約50%程度の工数短縮が確認された。また、各担当者がそれぞれ
の経験に応じてコード・ライブラリをメンテナンスし使い勝手の向上を図れることから、プログラ
ム製造工数は、現在よりも短縮されることが予想される。

プログラムの質的な側面からは、コード・ライブラリに登録されているプログラム・コードが動

作確認済みのものであることから、プログラム単体テストでのテスト項目を新規作成分に集中することができる。

VisualBasic コード・ジェネレータの処理スピードに関しては、典型的なデータベースアクセス・プログラムの場合、約 3,000 行程度のプログラム作成を実施するが、この処理の所要時間は約 2 分程度と確認されている。(処理の所要時間は、クライアント/サーバー構成、スタンドアロン構成のどちらでもほとんど同じである。)

(2) ドキュメント・ジェネレータ

従来のシステム開発では、データベース仕様書の作成方法として直接 MS-WORD 文書ファイルにデータベース仕様情報を書き込みするという方法であった。データベース仕様情報を特定の MS-WORD 文書ファイルで管理する場合、多数の担当者が同時にデータベース仕様情報を登録・更新・参照することは、MS-WORD 文書ファイルの排他制御の関係から運用上の制限が発生する。データベース仕様情報を複数の MS-WORD 文書ファイルに分割して管理する方法も考えられるが、どの MS-WORD 文書ファイルにどのデータベース仕様情報が登録されているかを管理する必要があり、効率の良い管理方法と言えるものではなかった。

今回開発したドキュメント・ジェネレータを利用する場合、データベース仕様に関する情報は、管理用データベース (Cache) で管理されているので、複数の開発担当者がデータベース情報を同時に登録・更新・参照することが可能となった。また、データベース仕様情報を印刷する必要が発生した場合でも、各担当者が必要な分のデータベース仕様情報を必要な時に印刷することが可能となった。(あるいは、MS-WORD 文書ファイルとして作成することが可能となった。)

このように、データベース仕様情報を複数の担当者が同時に登録・更新・参照できるようになったことで、ドキュメント作成時の生産性向上を図ることが可能となった。ドキュメント・ジェネレータを使用した場合には、従来方式と比較して約 70% 程度の工数削減が確認された。

5. 考察

(1) VisualBasic コード・ジェネレータについて

システム開発を実施する場合、システム開発に参画するメンバー全員がシステム開発ツールやシステム開発基準に精通している場合は少ない。そのような場合に、開発するシステム (具体的にはプログラム) の品質の確保と工数の最適化をサポートするツールとして、今回開発した 2 つのジェネレータ・システムが有効であることが確認された。

・スケルトン (雛形) 方式との比較

プログラム開発時に実施される方法として「スケルトン (雛形) 方式」が考えられる。スケルトン方式では、作成対象となるプログラムの画面上の動作やデータベース更新ロジックが固定的なものとして設計できる場合には非常に有効な方式であると考えられる。しかしながら、スケルトンで提供されている画面上の動作やデータベース更新ロジックでサポートされていないプログラムを作成する場合は、スケルトン自体をメンテナンスする能力が担当者に必要とされる。したがって、担当者にスケルトンをメンテナンスする能力がない場合、スケルトンがプログラムの動作やロジック

クを制限する方向に作用してしまい、利用者の本質的な要求に応えることができなくなってしまう可能性がある。

今回開発した VisualBasic コード・ジェネレータは、スケルトンを提供するのではなく、プログラム内で必要と考えられる部品を提供するにとどまっている。それらの部品をどう組み合わせているかは担当者の考え方に依存することとなる。従って、利用者からの要求に対応する場合には、要求に対処できる最適なプログラムを作成して提供することができる。このことは、VisualBasic コード・ジェネレータを十分に活用するためには、VisualBasic のプログラミング知識が必須の要件となることを意味している。

スケルトン方式での開発を行うか、あるいは、VisualBasic コード・ジェネレータの活用を考えるかは実際に開発するターゲットシステムの内容に依存することとなる。VisualBasic コード・ジェネレータを活用する場合、スケルトン方式で提供するプログラムの全てをコード・ライブラリに登録してしまうという方法も考えられるので、VisualBasic コード・ジェネレータ方式の方が柔軟な開発支援ツールといえる。

(2) ドキュメント・ジェネレータについて

ドキュメント・ジェネレータの活用により、データベース仕様書の作成工数を大幅に短縮できたのでシステム開発担当者が、本来のシステム開発作業に専念できる環境となったことに大きな意義がある。実際のプログラム開発フェーズにおいては、データベース仕様情報が一括管理されており、また、その情報が最新であることから、プログラム作成時に随時データベース仕様情報を確認しながら開発を進めていくことが可能となった。

データベース仕様情報は、データベース名称の一部を指定することにより検索をすることが可能となっている。システム内で使用するデータベース種類が非常に多数となった場合にも、データベース名称からの検索が可能であるので、作業効率の向上が図られる。

6. 今後の課題

今回開発した VisualBasic コード・ジェネレータは、各担当者が必要性を感じたプログラム部分を「部品」としてコード・ライブラリに登録して管理する方式となっている。この方式では、実際には頻繁に使用されているプログラム・コードであってもコード・ライブラリに登録されていない可能性がある。そこで、実際に動作している VisualBasic プログラムを解析して部品情報を提供し、コード・ライブラリへの登録をサポートする仕組みが必要ではないかと考えられる。

また、現在はプログラム生成対象を VisualBasic に限定しているが、他の開発ツールへの対応も必要であると考えられる。

ドキュメント・ジェネレータについては、データベース仕様書の作成のみであるが、メッセージ仕様書、ファイル仕様書、プログラム仕様書の作成も行える仕組みが必要であると考えられる。

今回の2つのジェネレータ・システム開発の経験を踏まえて、さらに開発支援ツールの開発を行い、今後のシステム開発に活用してゆきたい。

M 言語用のコンパイラ・コンパイラの作成

○鈴木利明* 嶋 芳成*

*日本ダイナシシステム株式会社

〒460-0007 愛知県名古屋市中区新栄2丁目1-9 雲竜ビル東館5F

Tel: 052-242-5441 Fax: 052-242-5984

E-mail: contact@jdynasys.co.jp

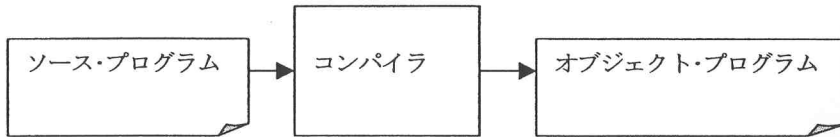
1. コンパイラ

コンパイラは高級言語を低級言語に変換するプログラムの総称である。一般的には FORTRAN, COBOL などの言語で記述されたプログラムをアセンブラや機械語などの等価プログラムに変換をするプログラムである。

入力は各言語で記述された応用プログラム(ソース・プログラム)であり、出力はその応用プログラムを実行するアセンブラであり、機械語(オブジェクト・プログラム)等である。広い意味ではアセンブラではなくても M 言語のソースを出力するものも、ここではコンパイラと呼ぶ。

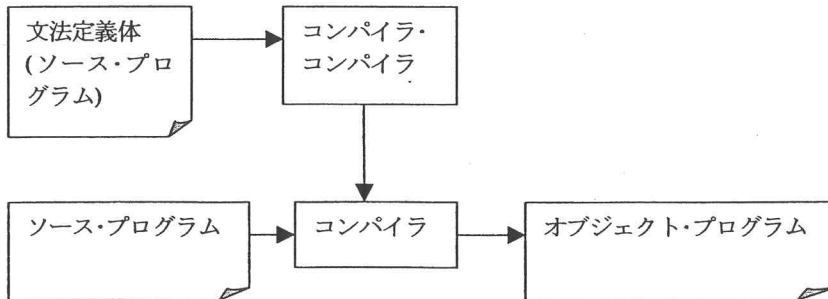
Caché では、SQL 言語を M 言語に変換してから実行している。

一般的には、コンパイラは動作速度、言語機能、最適化などの高度な振る舞いが要求されるため C 言語などで直接記述されることが多い。



2. コンパイラ・コンパイラ

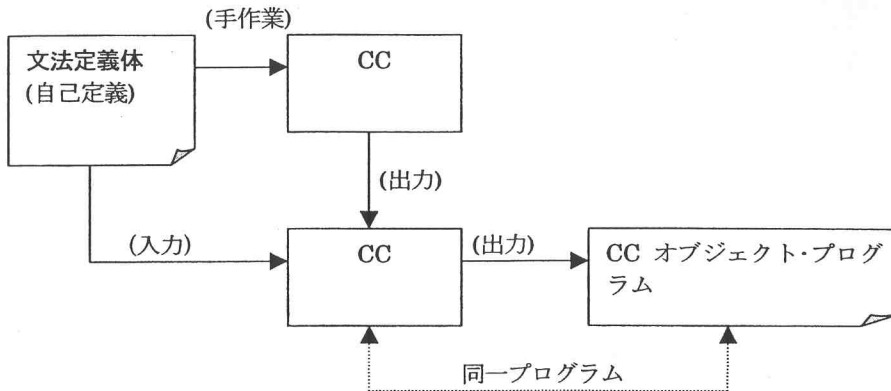
コンパイラ・コンパイラ (以降 CC) では、入力はコンパイラを記述したプログラム (文法定義体) であり、出力はそのコンパイラである。新しい言語のコンパイラを作成するとき、使われる手法である。参考文献 1 に CC の特集があり、今回はこれを元にした。



CDL → MAC → TUT → OBJ

3. プログラミング技法

CC は、再帰的処理、バックトラック処理、木構造などのプログラミング技法を使用する。再帰的処理は、DO、NEW 命令を使用する。スタックの使用は最小限とする。バックトラック処理は、C 言語では、longjmp と setjmp を組み合わせて実現している。M 言語では、二つの方法がある。ZETRAP などのエラー処理命令を使用する。または、すべての処理をユーザー定義関数で表わし、必ず真偽値を持って返る方法がある。木構造は、グローバル変数の添え字を使用することで実現することができる。これらの技法を使用することで、M 言語で CC を自己生成することができた。



4. 現在および今後

CC0 レベルは、上記方法で作成できた。現在 CC2 レベルを開発途中である。元にした CC は C 言語で記述されたシンプルなものである。ソースに対するオブジェクトの対応は、1対1のテンプレート変換が原則である。M 言語に書き換え拡張することにより、より高機能な CC 装備を作成する予定である。

5. CC 自己定義 一部

CC0 CC0

```
CC0 ::= "CC0" NAME BODY "END" : ";" /  
      ";" /  
      *1 /  
      *1 " ;Compiler-Compiler ver 1.0 ;" /  
      " q"  
      *2
```

;

```
BODY ::= STM BODY      : *1 *2  
       | STM           : *1
```

;

```
STM ::= NAME " ::= " EXP ";" : /  
      "% " *1 "(e)" #0  
      *2  
      " q"  
      / " ;"
```

;

```
EXP ::= RUL "|" EXP      : " d push^C0 zetrp " #1 /  
      *1 " d pop^C0 q"  
      / #2 " d err^C0"  
      *2  
      | RUL              : *1
```

;

6. CC の応用例

C 言語版の CC を適用して CC 技術が開発に十分な能力を備えていることを確認した。いずれは M 言語版の応用例となる予定である。

SGML および XML (Extensible Markup Language) には、DTD(Data Type Definition)と呼ばれる各 XML 文書の構造を決定する定義体が付随する。MML(Medical Markup Language)、病院経営分析情報交換データフォーマット(FAIR)や、POIX(Point Of Interest eXchange language)などの XML の応用毎にそれぞれの DTD あり、各 XML 文書のデータ構造を定義している。

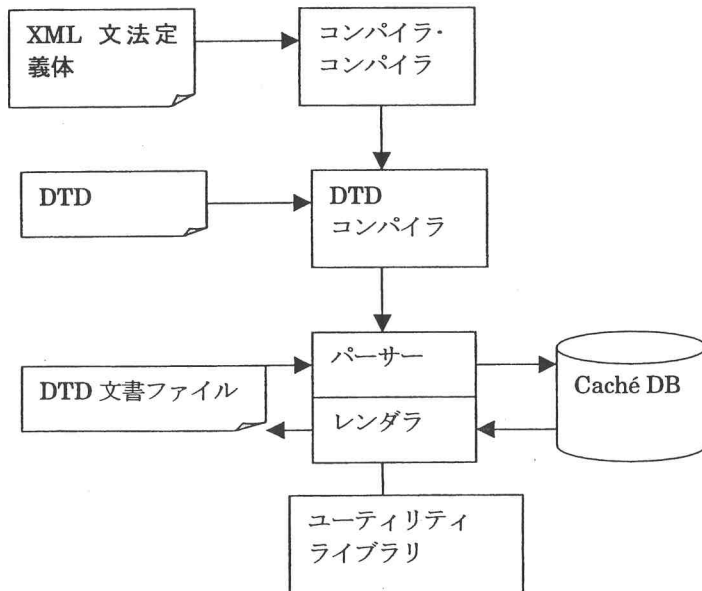
XML 文書の取り込みプログラムを一般的にパーサー、出力プログラムをレンダラと呼んでいる。

Caché 用の XML のパーサーレンダラを作るに際し CC 技術を提要した。理由は、

- 1) 正しいパーサー、レンダラを作成する必要があった。
- 2) DTD の変更に伴って、速やかにパーサー、レンダラも改変する必要があった。
- 3) 自動化への対応準備

ある DTD には 140 個の定義体があり、これを通常のプログラミング開発工程ではバグが入る可能性がたかかった。また、その DTD も改定される可能性があり、事実小さな改定も行われた。

DTD 全般に対応するためには、DTD からパーサー、レンダラを自動生成する必要があった。



参考文献

1. インタフェース 1995年12月号 特集コンパイラ・コンパイラの原理/制作/応用 村田 和信著
2. インタフェース 1997年2月号 特集 プログラマのためのANSI C 言語仕様の解剖学 p115-178 村田 和信著
3. インタフェース 1997年4月号 “ANSI C 言語 仕様の解剖学 の補足説明 p102-103 村田 和信著
4. ANSI/MDC X11.1-1995 Programming Languages - M
5. コンパイラの仕組み 朝倉書店 渡邊坦 著

JAVAによるM言語の装備

内田達弘

名城大学理工学部

〒468-8502 名古屋市天白区塩釜口1-501

TEL: 052-832-1151 FAX: 052-832-1169

uchi@meijo-u.ac.jp

1. はじめに

M言語の装備 [1] を1985年から行ってきたが、急激なオペレーティングシステムの変化に合わせるため開発に要する時間の大半をその対応に費やしてきた。近年、オペレーティングシステムやプラットフォームへの依存がきわめて少ないJAVA言語が登場しこの問題を解く鍵となると考え、JAVAによるM言語の装備を考えた。その結果、JAVA言語はマルチスレッド・ネットワーク・GUIを標準で備え、M言語の装備に必要な機能の大半を持ち、M言語はJAVAにない優れたデータベース機能とその処理機能をJAVAに追加することができるであると考え、装備を試みた。

本稿ではJAVAを使って装備したM言語の内部構造と動作の仕組みについて報告する。

なお、本稿ではこの装備をMJ1.0と呼ぶことにする。なお、1.0はバージョン番号である。

2. 装備の概要

MJ1.0の概要を表すため次の特徴を述べる。

2. 1 100%JAVA

MJ1.0は100%をJAVAで記述した。したがって原則としてJAVAが動くプラットフォームなら動作する。

2. 2 非標準M

M言語の標準仕様 [2] をみたくように装備を心がけたが、一部異なる部分があった。これはJAVAの機能をできるだけ多く活用するため標準M言語の仕様を完全に満たせなかったためである。特に入出力およびJOB関係の機能が仕様を満たしていないことを断っておく。

2. 3 中間言語

MJ1.0の処理方式はルーチンを直接解読して実行するインタープリタ形式ではなく、ルーチンを一旦、中間コードにコンパイルしてからそれを解読し実行する方式である。ルーチンおよびその中間コードはデータベースにグローバル変数と同じ要領で保存される。

2. 4 効率の良いデータベース構造

B+tree構造をもつ。キーの長さは可変長でありキー圧縮を行う。キーとデータ部は別ファイルに記憶される。

2. 5 スレッド

内部処理の中で独立して実行できるものはスレッド化されている。また、新しいJOBの起動に対して一つ以上のスレッドが生成され、そのJOBを処理する。

3. 内部構造

MJ1.0は、図1のようにKernel, Database system, Master manager, M application の4種類の基本モジュールからなり、それぞれがKernelを経由してデータの受け渡しと同期をとっている。

- Kernel はスレッドの生成、同期、エラー処理を行う。
- Database system はM言語のグローバル変数およびルーチンの管理を行う。アプリケーションから要求のあったグローバル変数に関する処理の多くはここで行われ、アクセスが効率よく安全に行われ

るようにキャッシュ機構やLOCK機構を備える。Kernelによりスレッドとして生成される。

- Master Manager はコンソール, Status viewer, Database manager, User manager, Routine editorの機能を持ち管理者とのインターフェースを行う。Kernelによりスレッドとして生成される。
- M application はジョブごとにKernelによりスレッドとして生成される。従って一つのジョブの処理は一つのM application スレッドが処理する。M applicationは中間コードをDatabase systemから読み出し、解読し実行する。GUIのようなユーザーインターフェースを持たず、TCP/IPでのインターフェースを提供する。GUIはJAVAやブラウザーを使って別途作る必要がある。

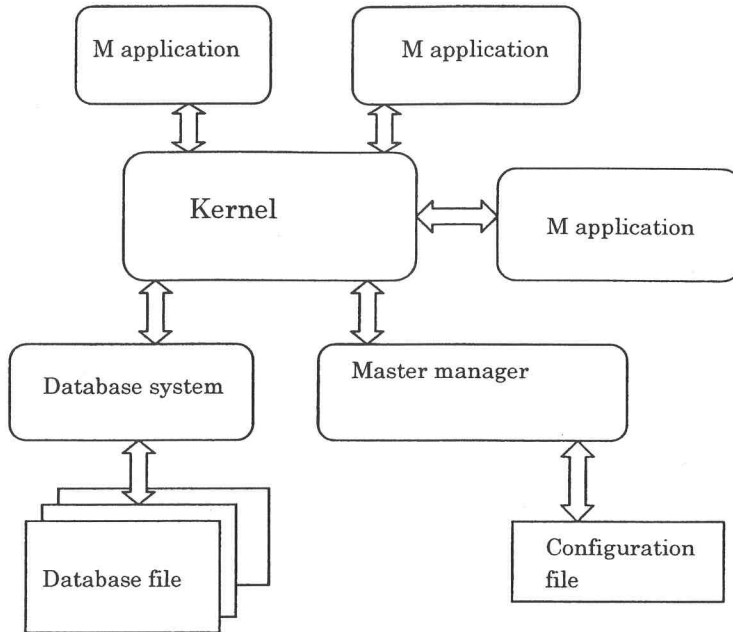


図1 MJ1.0の基本構成

3. 1 Kernel

MJ1.0の起動時に最初にJAVAによりロードされる。初期化時にMaster manager スレッドおよびDatabase systemスレッドの生成を行う。

機能

- ① スレッド間通信のため共有領域を提供し管理する。
- ② Master Managerから要求があればM applicationスレッドを生成したり、停止処理を行う。また、システムの停止も行う。
- ③ 通常のOSからの通信要求や割り込み、スレッドからのエラー通知があればMaster managerに通知する。
- ④ 回復不能のシステムエラーの通知をOSから受けた場合は管理下のすべてのスレッドに通知し停止処理を行う。
- ⑤ M applicationから要求されたグローバル変数名とデータベースファイル名および実グローバル変数名とのマッピングを行う。

3. 2 Database system

Kernelによりロードされスレッド化される。

機能

- ① Kernelから要求されたグローバル変数およびルーチンに関する操作を行う。変数アクセスに関する要求コマンドには表1のものがある。
- ② Database file にグローバル変数を保存する。
- ③ グローバル変数のアクセスを高速化するためデータのキャッシュを行う。
- ④ LOCK処理のためLOCKテーブルを持つ。
- ⑤ トランザクション処理のため、トランザクションファイルを生成する。(未実装)

表 1 グローバル変数に関するコマンドの一覧表

コマンド	処理内容
GET	指定したグローバル変数の値を返す
SET	指定したグローバル変数にデータをセットする
KILL	指定したグローバル変数を削除する
ORDER	\$ORDER関数処理
QUERY	\$QUERY関数処理
NEXT	\$NEXT関数処理
LOCK	LOCK処理
DATA	\$DATA関数処理
MERGE	未実装
TCOMMIT	未実装
TRESTORE	未実装
TROLLBACK	未実装
TSTART	未実装

3. 3 Master manager

Kernelによりロードされスレッド化される。

機能

- ① MJ1.0を制御するためのコンソール
- ② Status viewer システムの稼動状態をモニターする。
- ③ Database manager データベースはデータセットと呼ばれる単位で扱われるが、データセット名とグローバル変数名とのマッピングの設定、データセットの生成を行う。
- ④ User manager ユーザやグループの登録と削除を行う。
- ⑤ Routine editor ルーチンの作成と編集およびコンパイルを行う。

3. 4 M application

Kernelによりロードされスレッド化される。ユーザーーチンの実行は主としてこのスレッドで処理される。ただし、グローバル変数のアクセスとルーチンの呼び出しはKernelに要求する。

機能

- ① Xecute命令処理のための行コンパイラ
- ② 中間コードインタプリタ
- ③ TCP/IP通信機能
- ④ NAKEDリファレンス処理
- ⑤ ローカル変数処理

4. 実装の実際

MJ1.0を作成するために使用したシステムはWindowsNT4.0とJbuilder3である。J A V Aのバージョン

はJAVA 2である。C言語版のMを作成したことがり、それが役に立ったが、ポインターを明示できないJAVA言語にCソースを置き換えるのは非常に面倒な作業であった。

MJ1.0は30あまりのクラスで構成される。

基本設計を1999年2月から開始し、3月からコーディングを開始した。

すべて、MJで始まるクラスライブラリとして作成したため、任意のJAVAアプリケーションからインポートすれば再利用が可能である。

5. 終わりに

MJ1.0を装備することでM言語をJAVAで実装する際の利点と欠点が次のように分かった。

- ① オペレーティングシステムに依存しないコーディングができた。
- ② JAVAのオブジェクト指向の性質によりコーディング量が減った。
メモリ管理とエラー処理のコーディング量の減少は大きい。
- ③ ネットワークとスレッドが標準で用意されているためJAVAはMの装備を容易にする。
マルチジョブの実装はOSに深く関わるが、JAVAがそれを吸収する。
- ④ JAVAのVMの速度に実行速度が依存するため現状では遅い。
- ⑤ 漢字文字の扱いが楽である。

JAVAのChar型が2オクテット系の文字をサポートしているため日本語処理のために文字列処理に気を遣わなくて済むのは大きい。

M言語のインプリメンタが統合されCachêというM言語の機能を持ったシステムに集約されつつあることはインプリメンタの組織力の強化という面では歓迎できる面もある。しかしインプリメンタが少なくなってしまうことは残念である。このことは結果としてユーザがユーザの立場で使いやすいシステムを要求する場が減るということになってしまったような気がする。これからは自分たちが自由になるMを手に入れることが必要ではないかと思う。

このことがMJを作ろうとした動機の一つである。

まだ、完成度が低いが、本システムをオープンソース化して公開する予定である。

参考文献

- [1] 内田達弘, [1985], MUMPS の Prolog シンタクスの包含装備. Proceedings of MUG Japan Meeting, Vol.12, pp.19-32
- [2] 日本工業標準調査会: プログラム言語 MUMPS JIS X3011-1995, 日本規格協会, 東京, 1995

Java の最新動向

今泉幸雄(Yukio Imaizumi)

ファイザー製薬株式会社 IT 統括部

〒163-0461 東京都新宿区西新宿2-1-1 新宿三井ビル内 私書箱226号

TEL: 03-3344-7404、7538

1 はじめに

米 Sun 社のホーム・ページ(<http://www.sun.com>)からジャバ言語の開発環境の JDK (Java Development Kit)をダウンロードすれば、Java プログラムは書ける。最新の JDK は 1998 年 12 月に JDK1.2(通称 Java2)が提供された。ここでは、Java プログラムの動作形態と処理スピードの高速化の HotSpot、Web 環境におけるアプリケーションサーバの流れについて説明する。

2 Java プログラム動作形態

Java のプログラムには3つの形態がある。一つは Java アプリケーション (Application)、通常のクライアント・サーバ(C/S)やスタンドアロンで動く、一般的なアプリケーションである。2つめはサーバで動作するプログラムつまりサーブレット(Servlet)。最後にブラウザで動く Java アプレット(Applet)である。特に Java 言語の特長をすべて受け継いで、インターネット上のコンポーネント技術のプログラムが JavaBeans である。Java アプリケーションの基本は JDK であり、サーブレットは JSDK(JavaServletDevelopmentKit),JavaBeans は BDK(BeansDevelopmentKit)として提供される (Table 1: Java Servlet/Applet and JavaApplication/JavaBeans)。

業務アプリケーションが C/S からイントラネット Web 移行から、今アプリケーション・サーバに移ろうとしている時、サーブレットに注目が集まる。JavaBeans を基にした Java 版ビジネス・オブジェクトの EnterpriseJavaBeans (EJB)もある。

インフラにおいては Java の特長の分散オブジェクトとどの OS 間でも通信可能の実現に、オブジェクト指向技術の標準化の非営利団体の米 OMG (Object Management Group)の標準仕様としての CORBA Common object Request Broker Architecture)準拠の Java IDL (Interface Definition Language)、Java アプレット同士の通信インターフェース JavaRMI (Remote Method Invocation)が利用されている。

3 処理スピードの高速化のHotSpot

JDK1.2からHotSpotが提供されて高速化が期待されている。従来のJIT(JustInTime)コンパイラにおける問題を解決するためのエンジンである。バイトコードをJavaVM(JavaVirtualMachine)で実行しながら、遅い部分や頻繁に利用する部分をマシンコード(NativeCode)に変換するコンパイラである(Graph2 : HotSpot Java VM)。さらに使用済みになったオブジェクトに対するメモリのクリーンアップの効率化であるガーベッジ・コレクション(GarbageCollection)の導入と複数のI/Oデータストリームを同時に実行するマルチスレッディング(Multithreading)が導入された。

4 アプリケーションサーバとJava

アプリケーション・サーバを運用するコンピュータ環境は、クライアントPC、アプリケーション・サーバとDBサーバの3つからなる。クライアントPCは、HTML型のWWWブラウザ、JavaやActiveXControl(米マイクロソフト社)のアプレットを用いるアプレット型、従来の3階層型C/Sタイプに分けられる。アプリケーション・サーバと呼ばれる製品は、TP処理、オブジェクト開発ツール、WWWとDBMSの組み合わせツール、3階層型C/SとDBMSの組み合わせツールなど出発点はことなるが、クライアントPCのTHIN(簡易性の薄さ)向けの方向性は同じである。拡張性と信頼性を向上させて業務システムへの適応性とJava等の新技術への適応性が期待される。米Sun社からJavaWebServerが標準で提供されている(Graph3 : Java Web Server)。

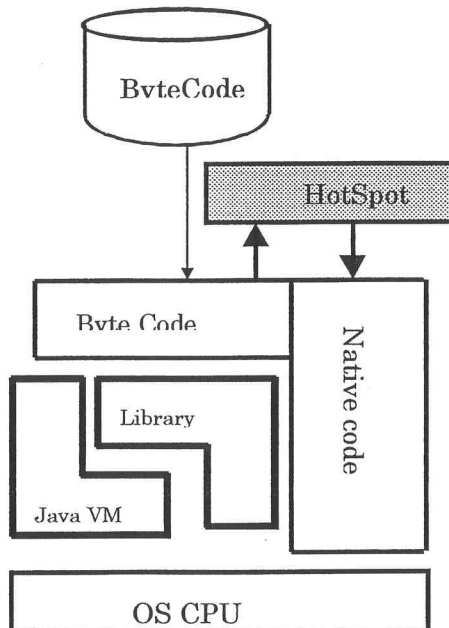
業務ロジックが集中するアプリケーション・サーバでは、クライアントPCが増加した場合、サーバで動くアプリケーション・プログラムが多くなった場合、WWWサーバとアプリケーション・プログラムが動くサーバを異なるマシンにする場合やアプリケーション・サーバを複数台にする場合にはシステムの耐障害を高めるなどの信頼性を確保する必要がある。

参考文献 :

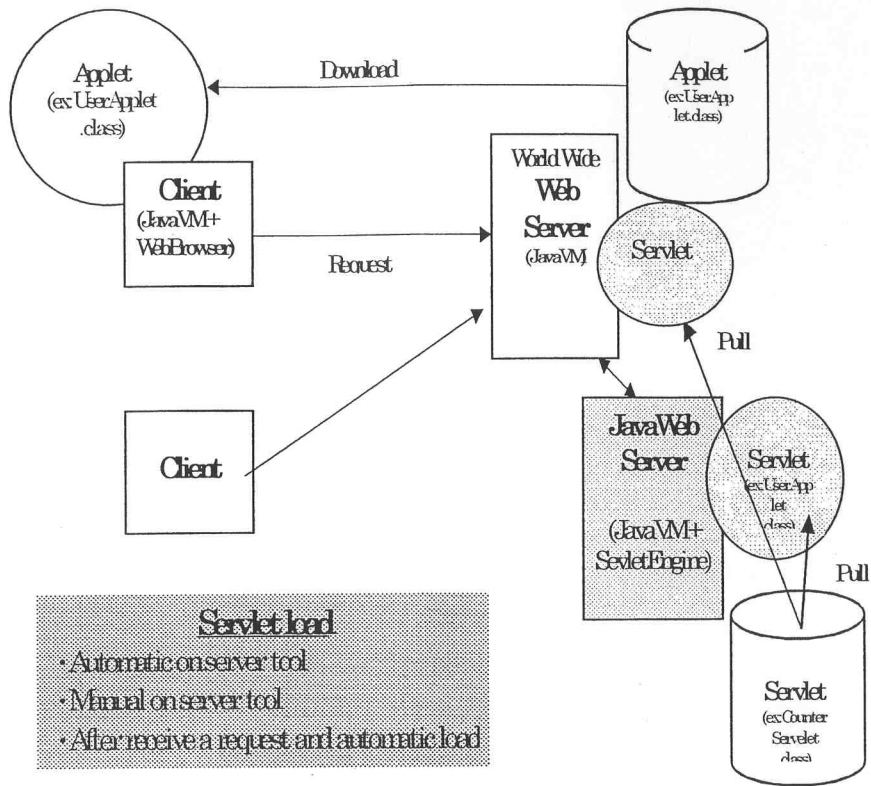
- 1 SAMS Teach Yourself Java1.2 in 24 hours
- 2 SAMS Java1.2 Class Libraries Unleashed
- 3 <http://java.sun.com/products/hotspot>
- 4 CHRIS TAYLOR, TIM KIMMET "CORE Java Web Server"

Table 1: Java Servlet/Applet and JavaApplication/JavaBeans

	Java C/S		Java Application		
	Java Servlet	Java Applet	Application	Develop JavaBeans	Java Beans
Server	○	×	○	-	○
Client	×	○	○	-	○
Development Kit	JDK, JSDK	JDK	JDK	Jar(Archive) /JDK BeanInfo-class	BDK
Sub class (extends.)	HTTPServlet	Java.applet Applet	[java.lang. object]		
others	resident style package(Javax. servlet.Javax. servlet.http)	demand-load style In HTML /client		Jar file	



Graph2 : HotSpot Java VM



Graph3 : Java Web Server

Andres Diekow

M 日本

Joe Gallant

Object.

Stavros Maniatis

www.

Phil Pylfus

Terry Ragon

Stephen Ragon.

Maccowski

第27回日本エム・テクノロジー学会大会のお知らせ

第27回日本エム・テクノロジー学会大会

大会長 沢田 潔

名古屋第二赤十字病院 医療情報部

記

第27回日本エム・テクノロジー学会大会(MTA2000)を名古屋第二赤十字病院にて開催いたします。

会期(予定) 2000年8月下旬頃

会場(予定) 名古屋市昭和区妙見町2-9
名古屋第二赤十字病院 1病棟10階 カンファレンスホール(大会会場)

アナウンス用 URL(設定中)

<http://www.nagoya2.jrc.or.jp/dmi/mta2000/index.html>

MTA2000 専用の公開メーリングリスト(設定中)

mta2000-ml@nagoya2.jrc.or.jp

大会事務局： 〒466-8650

名古屋市昭和区妙見町2-9

名古屋第二赤十字病院 医療情報部

FAX：052-832-1130

TEL：052-832-1121 内線：3066

担当：川田 新一、沢田 潔

E-mail：mta@nagoya2.jrc.or.jp