

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

Proceedings '97 M Technology Association of Japan

Proceedings '97

M Technology Association of Japan

The 24th Conference

Tokyo Fashion Town, Ariake. July 11-12, 1997.

Organizing Committee of the 24th Conference



Proceedings '97
M Technology Association of Japan

第24回日本Mテクノロジー学会大会

平成9年7月11日(金)12日(土)
東京ファッションタウン, 有明

第24回日本Mテクノロジー学会大会 組織委員会

Computing Paradigm の変革における M ～ 成長する実運用システム と Mソリューション ～

廣瀬 康行

第24回日本Mテクノロジー学会大会 大会長

ここ十年ほどの間に Computing 環境は激変しました。最近では Internet や Intranet というはやりの言葉とともに、OpenDoc/OLE, Active-X, CORBA, Java/Castanet などの急速な普及が如実に示しているように、充実した・そして最新の network ならびに database 機能が不可欠となっています。

このような流れは、今やハードウェアとソフトウェアの両面において、システム構築の考え方自体にも甚大な影響を及ぼしています。この意味において、Computing Paradigm の変革が求められている時代のように思える昨今です。

Mはこれまで、処理速度やコストパフォーマンスの良さ、開発効率や移植性などにおいて優位性を誇ってきました。しかし今後は、そのような価値観のみでは concurrency を維持していけないのかも知れません。換言すれば、時代の流れはMからの新しい対応を求めているようです。

エンドユーザは、何を見て・何を求めていくのでしょうか。そしてベンダーやVARは、何を提供することができるのでしょうか。時代の流れから、何を学び取って・何を捨て・何に素早く対応するとともに、その成果を効果的に周知できるか否かは、今後ますます重要となってくるでしょう。

好むと好まざるとに関わらず、おそらく相当の長期間にわたって、潮流は network computing と object handling というキーワードを掲げ続けていくものと思われまます。

第24回日本Mテクノロジー学会大会ではこのような観点も踏まえつつ、種々の成果と多様な問題について活発に議論して戴きたいと存じます。また、大会参加者各人の豊富な経験とアイデアを互いに吸収しあいながら、明日のMのありかたや応用可能性を考えていきたいと願っております。

本大会がMの発展と普及の一助となることができれば、大会長として誠に幸甚に存じます。

目次

■ ■ 第24回日本Mテクノロジー学会大会	
■ 大会長	
廣瀬 康行	(東京医科歯科大学歯学部 歯科麻酔学講座) (同歯学部附属病院 情報処理委員会企画調整専門部会長)
■ 組織委員会	
岡田 好一	(東海大学医学部 医療情報学講座)
笹川 紀夫	(島根医科大学 医療情報学講座)
田久 浩志	(東邦大学医学部 病院管理学教室)
山崎 俊司	(千葉大学医学部附属病院 医療情報部)
山下 芳範	(福井医科大学 医学情報センター)
小倉 常睦	(住友電気システムズ株式会社 応用システム事業部営業部)
松本 重雄	(日本MSM株式会社)
■ ■ 第3回Mテクノロジーフェア	
■ 実行委員長	
小倉 常睦	(住友電気システムズ株式会社 応用システム事業部営業部)
■ 実行委員会	
松本 重雄	(日本MSM株式会社)
嶋 芳成	(日本ダイナシステム株式会社)
前海 好昭	(日本デジタルイクイップメント株式会社 西日本第一統合システム部)

● 巻頭言	1
● 開催組織	2
● 目次	3
● 概要	4
● ビジネスデイ プログラム	5
● アカデミックデイ プログラム	6
● 論文	9
1) Mによる病院システムでのMML実装試験とサーバクライアントの利用	10
2) M言語を用いたMML対応の画像情報システムの設計	14
3) 市町村における予防接種業務のシステム化	16
4) 光カードを利用した診療支援システムの構築	20
5) M/WebLinkを用いたマルチメディア電子教科書	22
6) MとWWWの連携～M言語を用いたWWW文書作成について～	26
7) MとWWWの連携による看護過程支援システム	30
8) Visual Mとビジュアルコンポーネントとの結合	34
9) Visual BasicとOpenMによる保健指導システムの開発	38
10) Mにおけるリモートプロシージャコールの利用	42
11) Mストアードプロシージャを用いた自動再来受付機の開発	46
12) MグローバルをアクセスするJavaライブラリの試作	50
13) OpenM/SQL+ODBC+PowerBuilderによる販売管理システム開発	54
L1) はやわかりデータベースシステム動向97	62
S1) InterSystems ワールドワイド開発者向け戦略会議報告	78
S2) 北米M Technology Association 参加報告	82
14) M言語による構文解析における英文のスケルトン構造	86
15) 決断過程の構造化分析	90
● 広告	95
● 協賛者一覧	103
● 日本Mテクノロジー学会	104
● 編集後記	105

概要

■ 会期と会場

会期： 1997年 7月11日(金) 12日(土)
会場： 東京ファッションタウン (TFT) 東京都江東区有明 3-1

7/11 ビジネスデイ & チュートリアル

・チュートリアル	TFT 東館9階	9-A, 9-D
・プロダクトレビュー	TFT 東館9階	9-D
・理事評議員会	住友電工(株)	会議室
・懇親会	TFT 東館9階	9-A

7/12 アカデミックデイ

・口演ならびに講演	TFT 東館9階	904
・シンポジウム	TFT 東館9階	904
・学会総会	TFT 東館9階	904
・大会本部	TFT 東館9階	902

案内： 7/11 は9-Aを、7/12 は902を 総合案内 といたします。
TFT 9階へは 緑色A のエレベータを御利用下さい。

■ 交通機関

いづれの交通機関を用いるにせよ 東京駅から会場まで45分程度 はかかります。

新交通ゆりかもめ： 新橋駅から 国際展示場正門駅まで 約20分

臨海副都心線： 新木場駅から 国際展示場駅まで 約5分

水上バス： 日の出桟橋から 有明南客船ターミナルまで 約25分

■ 御食事

昼食は、東京ファッションタウンの2階にある フードパティオ を利用されるのが便利です。

ファーストフードから懐石まで、各種の店舗がそろっております。

TFTガイド： <http://www.JOLF.co.jp/odaiba/wanza.html>

またナイトライフを楽しむには、お台場駅周辺の各種施設や新橋界隈が、近場でのお勧めです。

お台場ガイド： <http://www.JOLF.co.jp/odaiba/index3.html>

■ 参加費

参加者は受付にて登録してください。参加証などをお渡しいたします。

学会大会 (学会員)	¥ 10,000	(抄録集込み)
学会大会 (非学会員)	¥ 12,000	(抄録集込み)
学会大会 (学生)	¥ 6,000	(抄録集込み)
懇親会	¥ 6,000	
初級チュートリアル	free	
中上級チュートリアル	¥ 5,000	(資料等込み)

■ Mテクノロジーフェア

DataWareHouse EXPO '97：東京国際展示場 (東京ビックサイト) にて 7/09 から 7/11 まで。

ビジネスデイ プログラム

■ TFT 9-D

09:30 -	受付開始		
09:50 -	大会開催 挨拶		
10:00 - 12:00	初級チュートリアル		(東海大学：岡田)
13:00 - 14:30	プロダクトレビュー	[MSM]	(司会：SSY 上戸, 北尾)
14:30 - 16:00	プロダクトレビュー	[U.S. InterSystems]	(司会：SSY 上戸, 北尾)
16:00 - 17:50	ユーザ交流会	[事例紹介：(株)新和]	(司会：JDS 嶋)

■ TFT 9-A

09:30 -	受付開始		
10:00 - 12:00	中上級チュートリアル	[DB設計とSQL]	(JDS：広瀬)
13:00 - 15:30	中上級チュートリアル	[Visual-MとWeb技術]	(DEC：平野, 西田)
15:30 - 17:15	中上級チュートリアル	[MSM-GUI]	(MSM-J：松澤)

■ 理事評議員会

13:00 - 住友電工システムズ株式会社 会議室にて

■ 懇親会 (TFT 9-A)

18:00 - 夏の夕暮れどき、宴もたけなわとなるころには、このレセプション会場からは美しいイルミネーションを楽しむことができます。
これを肴としながら、クライアントとベンダーの隔たりを越えて交流と親睦を深めたいと思っております。

発表要項

■ 発表者の方へ

発表登録：発表予定時刻の30分前までに登録を済ませて下さい (9:00より登録開始)。
メディア：OHP または 35mmスライド (スクリーンは1台のみ)
口演時間：発表10分+討論5分 (合計15分)

■ 座長の方へ

座長登録：セッション開始予定時刻の15分前までに登録を済ませて下さい。
進行：一任しますので活発な議論をお願いします。ただし終了時刻は厳守願います。

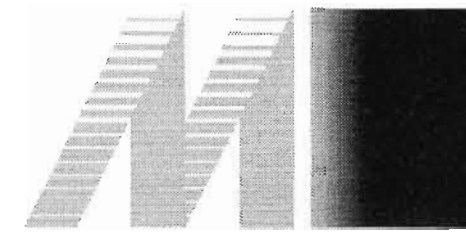
アカデミックデイ プログラム

■ TFT 904

- 09:00 - 受付開始
- 09:30 - 10:00 座長：山崎 俊司（千葉大学医学部）
- 1) Mによる病院システムでのMML実装試験とサーバクライアントの利用
（MML：Medical Markup Language ～ DTD/SGML による診療情報交換規約）
山下 芳範
福井医科大学医学情報センター
 - 2) M言語を用いたMML対応の画像情報システムの設計
山本 和子¹，笹川 紀夫¹，柳楽 真佐美¹，
栗原 潤一²，村上 英²，八川 剛志²
島根医科大学医学部医療情報学講座¹，住友電気システムズ（株）²
- 10:00 - 10:30 座長：笹川 紀夫（島根医科大学）
- 3) 市町村における予防接種業務のシステム化
大櫛 陽一¹，奥山 純二²，飯野 咲子²，小林 佐枝子²，小川 一博²
東海大学医学部医療情報学¹，山形県西川町保健センター²
 - 4) 光カードを利用した診療支援システムの構築
大門 宏行
診療システム研究フォーラム
- 10:30 - 11:15 座長：山下 芳範（福井医科大学）
- 5) M/WebLink を用いたマルチメディア電子教科書
岡田 好一，春木 康男，大櫛 陽一
東海大学医学部医用情報工学系
 - 6) MとWWWの連携 ～ M言語を用いたWWW文書作成について ～
八川 剛志，岡田 康
住友電気工業（株）MEシステム部
 - 7) MとWWWの連携による看護過程支援システム
山下 芳範
福井医科大学医学情報センター
- 11:15 - 11:45 座長：岡田 好一（東海大学医学部）
- 8) Visual M とビジュアルコンポーネントとの結合
佐藤 比呂志
日本デジタル・イクイップメント（株）西日本第一統合システム部
 - 9) VBとOpen M による保健指導システムの開発
大櫛 陽一¹，嶋 芳成²，西 満里子³
東海大学医学部医療情報学¹，日本ダイナシステム（株）²
岐阜県国府町保健福祉会館³
- 12:30 - 13:00 日本Mテクノロジー学会 総会

- 13:00 - 13:30 座長：土屋 喬義（土屋小児病院）
- 10) MにおけるRPCの利用
杉山 閑照¹，城崎 礼子¹，山下 芳範²
日本デジタル・イクイップメント（株）西日本第一統合システム部¹
福井医科大学²
 - 11) Mストアードプロシージャを用いた自動再来受付機の開発
鈴木 隆弘¹，山崎 俊司¹，高林 克日己²，中野 広樹¹，劉 亜斌¹，
本多 正幸¹，里村 洋一¹
千葉大学医学部附属病院医療情報部¹，同医学部第二内科²
- 13:30 - 14:00 座長：田久 浩志（東邦大学医学部）
- 12) MグローバルをアクセスするJavaライブラリの試作
堀田 稔
日本デジタル・イクイップメント（株）西日本第一統合システム部
 - 13) OpenM/SQL + ODBC + PowerBuilder による販売管理システム開発
広瀬 清司
日本ダイナシステム（株）
- 14:00 - 15:00 特別講演
- 座長：今井 敏雄（日本デジタルイクイップメント）
- L-1) はやわかりデータベースシステム動向97
今泉幸雄
ノバルティス ファーマ（株）
- 15:00 - 15:40 シンポジウム：明日のMテクノロジーを考える
- 座長：河村 徹郎（鈴鹿医療科学技術大学），嶋 芳成（日本ダイナシステム）
- S-1) InterSystems ワールドワイド開発者向け戦略会議報告
今井 敏雄
日本デジタル・イクイップメント（株）西日本第一統合システム部
- S-2) 北米 M Technology Association 参加報告
上戸 隆¹，北尾 真一²
住友電気工業（株）¹，住友電気システムズ（株）²
- S-3) 学会長コメント：日本のMの現在と将来
河村 徹郎（日本Mテクノロジー学会長）
- 15:40 - 16:15 Mテクノロジーの新しい応用へ向けて
- 座長：藤江 昭（住友電気システムズ）
- 14) M言語による構文解析における英文のスケルトン構造
～ 構文解析におけるMソリューション ～
高橋 亘
関西福祉科学大学
 - 15) 決断過程の構造化分析
～ マネージメント支援システムの新しいありかたへ向けての考察 ～
廣瀬 康行
東京医科歯科大学歯科麻酔学講座，同歯病情報処理委員会企画調整部会長
- 16:20 - 16:30 大会閉会 挨拶

論文



Mによる病院システムでのMML実装試験と サーバクライアントの利用

A trial of MML on M based HIS

山下 芳範

Yoshinori Yamashita

福井医科大学 医学情報センター

Center of Medical Informatics, Fukui Medical School

Abstract: In the Fukui medical school, the hospital information system was completely changed to the new decentralized system. For new application of medical records, the information processing was moved to the client side.

The electronic medical record has examined and the problem concerning the system by this system and making the prototype.

We changed the patient database and data communication for EMR, try a handling MML interface.

Keyword: HIS, Electronic Medical Record, MML

はじめに

本学では、M言語によって構築された病院情報システムの中で電子カルテを実現できるように、システムの更新後から従来の機能を見直している。

電子カルテについては、試作として実験を行っているが、病歴データベースについては電子カルテに対応できるように、変更を加えてきた。また、この病歴データベースを利用して、MMLへの対応についても実装試験を行っている。現時点では、電子カルテそのものとしては稼動していないが、基本的な構成としては電子カルテに対応できるような方式へと転換を行っている。現在は、MMLへの対応も含めて全体の改良を行っており、これらのシステムにおける、電子カルテ対応やMMLの組み込みのなど周辺要素の試行をすすめている。

概要

現在のシステムは、クライアント側にwindows NTを採用し、同時に機能の分散を図っている。更新当初は、全端末にMを載せてGUIとしてMWAPIを用いて処理を行っていたが、本年からは、Visual BasicによるGUI開発を取り入れ、Visual Mによるシステムへと転換を行った。

本来、電子化カルテの実現にはオブジェクト型のデータベースを用いて作成することが適しているが、既存システムとの統合や他システムとの結合や共有などの点から、従来と同様なサブシステムやデータベースを相互交換する仮想的な管理による方法でアプローチしている。このため、以下の方針で段階的に改定を行っている。

1. 独立可能なシステムについてはHL 7等の標準プロトコルを利用すべく、電子化カルテの対応側で考慮する。

2. 会計系統・部門系統および部門系統と共有する部分は、従来の方法との互換を保証して独立させる。

3. 病歴系データベースは、必要に応じて内容を見直しする。

このような点に従って、現状のシステムをもとに実装する作業を行っている。

これらのことを実現するために、異なるシステムやデータベースを統合することや、GUIでの病歴表現を容易にするため、関連付けをMMLでの表現をもちいることとした。

方法

電子化カルテおよびMML実装にあたって、データの整合性やトランザクションに関しての検討を行った。既存アプリケーションについては、順次対応を考慮し、システム間通信を基本とするように独立させた。これは、本学のシステムが、会計系及びオーダー系のシステムが病歴データベースを共有するようになっているためである。

MML実装及び電子化カルテの対応するために、システム内部向けの標準インターフェース関数を準備し、これを経由することで、既存及び新規のアプリケーションに対応させた。また、システム間独立は現実的には困難であるため、この関数での管理下でコントロールできるようにした。

このインターフェース関数の下で、それぞれの要素を管理するためのデータベースを定義し、このデータベースがそれぞれのオブジェクトの実体、状態、履歴(バージョン)を管理することとした。このデータベースは、新しく作成した電子化カルテに向けたGUIベースの表示のインデックスとしても利用することになった。このインターフェース関数は、システム間での取り扱いの違いや目的に応じたデータのハンドリングを隠蔽化することが目的であるが、HL 7の標準手順採用時にも機能拡張をここで可能とすることにも配慮した。

ここでのMML対応におけるインターフェース関数の対応機能は以下の通りである。

1. MMLのタグに対応した、データベース内のオブジェクトとの対応
2. そのオブジェクトと関連した履歴の保持と管理
3. 外部からの参照や変更にとまなう実体と関連するオブジェクトとの同期処理
4. 実(既存)運用システムとの仲介

これらの機能をインターフェース関数で実現することで、実際の運用を行っているシステム内に組み込むとともに、新規運用アプリケーションはこれを経由する方法を組み込み実運用の中で利用している。しかし、実運用のシステム自身では、MMLを直接利用していない。

まとめ

MMLの実装を通して、現状の病歴データベースや既存システムに関するシステム間通信やトランザクションを明確にすることができ、電子化カルテに向けてのシステムの変更や見直しをすることに役立った。

特に、オーダーシステムから発生する、変更更新イベントによる日付等に関する部分や先日付や予約分に関する情報更新イベントについて、システム間での取り扱いの差異を明確にし、調停などの整合方法をまとめることができた。

今回の実装試験は、MMLの対応だけでなく、電子化カルテの実現に向けたテストであったが、実

装のために開発したシステム内部用の関数については、実用的に利用可能であるため、既に実際のシステムに組み込んで運用を開始している。

今回の実装試験を通して、MML対応によるデータの取り扱いは、現時点のシステム上で移行を進めているGUI対応のシステムでの利用でも大きな効果があったといえる。

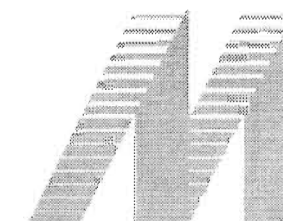
また、Mの文字処理機能やデータベースの柔軟性は、MMLへの対応を容易なものとし、HL7などのメッセージについても容易に取り扱えた。

特に、表現が階層となるような構造は、Mのデータベースでの取り扱いでは簡単に実現できる。

今後の方向としては、MMLのタグに対応した、管理定義を作成する部分を半自動化するために、タグとの対応の書式化とこれに伴うデータやシステムのコーディングの自動化を行い、電子化カルテへの対応を行う予定である。

このような部分でも、文字処理や階層データを扱うので、Mでの利用は容易なものであると考えられる。

また、MWAPIからVisualMによるサーバークライアントへの移行に関しては、変更の複雑化やネットワーク負荷も予想よりも低く、スムーズな移行が行えた。GUIに関しては、豊富なVBの資産が利用できるため、従来実現できなかった表現なども取り入れることができた。



M言語を用いたMML対応の画像情報システムの設計

山本和子¹⁾、笹川紀夫¹⁾、柳楽真佐実¹⁾、栗原潤一²⁾、村上英²⁾、八川剛志²⁾

1) 島根医科大学医学部医療情報学講座

2) 住友電工システムズ(株)

〒693 島根県出雲市塩冶町 89-1

TEL : 0853-23-2111 内線 2941 FAX : 0853-25-2764

yam@shimane-med.ac.jp

1. はじめに

画像情報の蓄積・検索システムは、いくつかのキーワードを登録しておいてそれで検索する独立型ファイル管理システムがこれまで開発されている。しかしながら、高速LANが普及し、パソコンの性能が急速に向上し、画像情報をパソコン画面に表示可能になるに従い、これまでのスタンドアロン型のファイル管理システムでは対応しきれなくなっている。そこで本学では、画像データベースの索引部分をM言語のデータベースで構築し、ネットワークによるMML対応の情報参照系システムのサブシステムとして設計したので、今回はその概要について報告する。

2. 情報参照系システムの概要

情報参照系システムは図1のような構成をしている。病歴情報データベースサーバは、機種はSUN-7/300 M170(磁気ディスク10.5GB, 100GBの光ディスク付)である。使用言語はM言語で、情報検索用に索引部(IndexDatabase)を持っている。医療画像データベースサーバは、機種は同様にSUN-7/300 M170(磁気ディスク21GB, 424GBの光ディスクと600GBのCD-ROMのオートチェンジャー付)である。使用言語はM言語で、画像検索用に索引部(IndexDatabase)を持っている。

データベースは、将来電子カルテとしてMMLに自動変換可能なように設計している。また、インターネットのWWWサーバに掲載できるように、ページ単位の形式としている。

3. 情報の収集・蓄積

病院情報システムの医療情報(オーダ情報、検査成績、退院時要約等)が病歴情報データベースサーバに毎日転送され蓄積される。同時に情報検索用の逆ファイルが作成され、索引部(IndexDatabase)に蓄積される。

放射線部で作成される検査報告書のデータは、報告書作成システムからNTサーバを通じて転送され、データベースと索引部に必要な情報が蓄積される。

放射線部で作成されるDICOMフォーマットの医療画像は、テープ渡しのオフラインで医療画像データベースに転送され、参照画像はgif/jpeg形式に変換され蓄積される。同時に画像Indexが作成され、このIndex情報は病歴情報データベースの索引部にも転送される。

4. 情報の参照

病歴情報データベースの索引部には、病歴情報と医療画像の情報が集められているので、ここを検索すれば必要な情報を取り出すことができる。病院情報システムの端末の

GUIはVisualBasicで作られているので、医療画像は病歴データベースの索引部からの指示により直接、端末に表示される。また、情報参照系は病院内の医療情報WWWサーバを通じて、Netscapeのブラウザで情報を表示することも可能である。この時はMMLのタグに対応した形式でデータを取り出すこともできるように設計している。

5. 今後の課題

当初は、放射線部で作成された画像を医療画像データベースサーバにオンラインで転送することを考えていたが、容量が多く、負荷の関係で中止し、毎日、バックアップ用に作成されたテープを借りてきて読み込ませることにした。また、参照系システムとして参照画像のデータベースを構築したいところであるが、現状は参照画像の選別ができないので、全ての画像を蓄積している。いずれ放射線部からの画像転送は参照画像のみとし、オンラインで収集できるようにしたいと

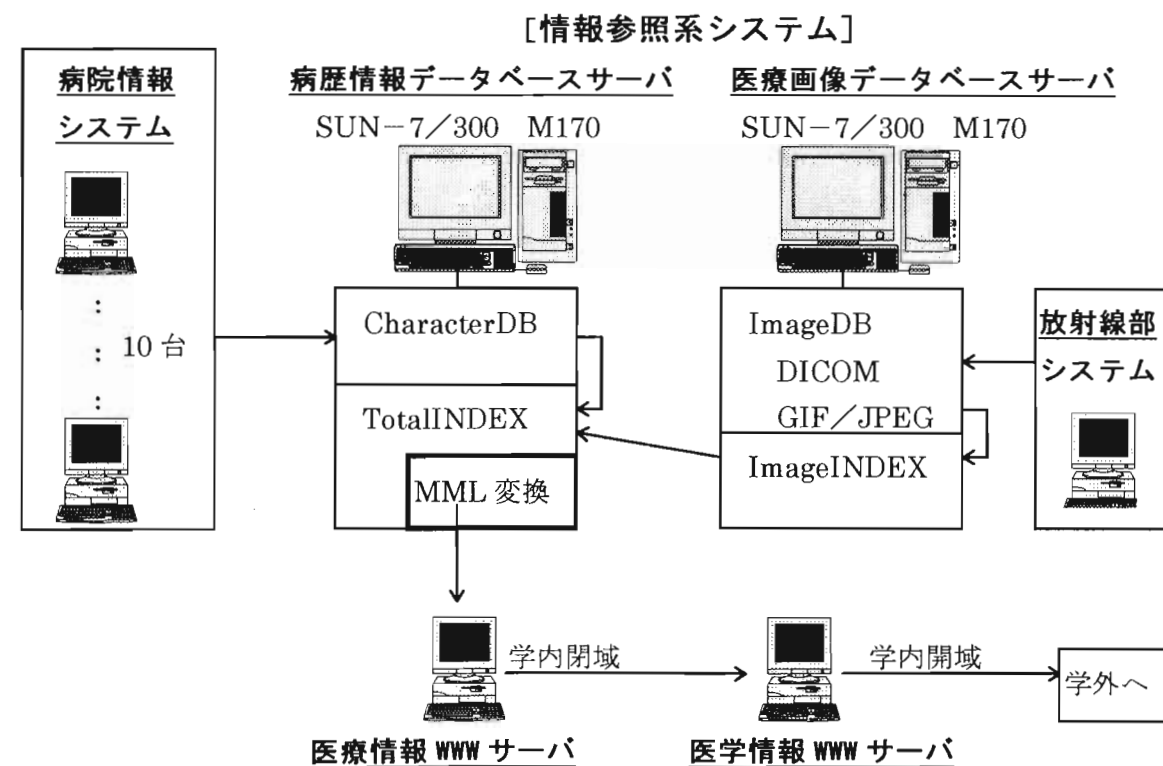
考えている。

今後の病院情報システムは、WWWサーバのイントラネット機能を用いたシステムが発展すると考えられる。本学においても、本情報参照系システムを利用してNetscapeのブラウザで参照可能な基盤を構築したので、今後は機能の充実を図りたいと考えている。

6. おわりに

病院情報システムで発生する各種病歴情報はM言語の分散型データベースで蓄積されている。医療画像データベースの索引部をM言語のデータベースにすることにより、病歴情報と画像情報とを統合化した分散型データベースを構築することができ、必要な時に必要な情報を必要な場所で、画像を含む医療情報を統合化して参照・検索が可能となった。また、MMLのタグに対応した形式でデータを抽出できるように設計し、将来の電子カルテ時代に対応できるようにした。

図1. 情報参照系システムの概要



市町村における予防接種業務のシステム化

大橋陽一¹⁾、奥山純二²⁾、飯野咲子²⁾、小林佐枝子²⁾、小川一博²⁾

1) 東海大学医学部医学情報学：〒259-11 神奈川県伊勢原市望星台

Tel:0463-93-1121 Fax:0463-96-4301 youichi@keyaki.cc.u-tokai.ac.jp

2) 山形県西川町保健センター：〒990-07 山形県西村山郡西川町海味543-8

Tel:0237-74-2111 Fax:0237-74-4811

Abstract: We have developed a information system for vaccination in a city. This system contains the following functions: listing of subjects, resistration of reservations, listing of approvals, booking control of medical faculties, resistration of results of vaccination and listing of personal vaccination histry.

Key words: Vaccination, Reservation, Approval, Booking Control, Personal vaccination histry, MUMPS, M technology, MWAPI, MSM

1. はじめに

市町村は、予防接種法、結核予防法の基ついで、疾病の発生及びまん延を予防することが義務づけられている。予防接種法の平成6年の改正により、予防接種の健康被害回避と救済にさらなる努力が要求されるようになった。また、従来の集団接種から個人接種へと流れが変わり、これに伴い市町村の事務量が大きくなった。

2. システムの機能

開発したシステムでは、以下の予防接種を対象としている。ポリオ初回、ポリオ追免、ツ反(乳児)、ツ反判定(乳児)、BCG接種(乳児)、再ツ反(乳児)、再ツ反判定(乳児)、再ツ反BCG接種(乳児)、ツ反(小1)、ツ反判定(小1)、BCG接種(小1)、ツ反(小2)、ツ反判定(小2)、BCG接種(小2)、ツ反(中1)、ツ反判

定(中1)、BCG接種(中1)、ツ反(中2)、ツ反判定(中2)、BCG接種(中2)、三混Ⅰ期1回目、三混Ⅰ期2回目、三混Ⅰ期3回目、三混Ⅰ期追加、三混Ⅱ期、二混Ⅰ期1回目、二混Ⅰ期2回目、二混Ⅰ期追加、二混Ⅱ期、麻疹、風疹、日本脳炎Ⅰ期1回目、日本脳炎Ⅰ期2回目、日本脳炎Ⅱ期、日本脳炎Ⅲ期。これらの予防接種に関して以下の機能を持っている。対象者の条件指定抽出指示(図1)、対象者名簿印字(図2)、希望入力(図3)、接種券発行(図4)、接種結果登録(図5)、予防接種個人台帳(図6)。

3. システム構成

OSはWindows95、開発言語及びデータベースはMSM、GUIはMWAPIを使い、M言語のみで開発を行った。プラットフォームは、携帯型パソコンであるFMV-VIBLO(ペンティアム70MB)と、プリンターはCANONのレーザープリンターを使用している。

4. システムの特徴

- ①対象者の抽出に当たっては、接種種類、生年月日範囲、性別、地区範囲より、細かく対象者の把握ができるようにした。
- ②対象者名簿では、前回までに対象疾患の罹患以外の理由で非接種となっている人を対象者リストにあげるようにした。
- ③希望入力では、医療機関ごとの予約管理を行い、予約日時のチェックと希望者の調整及び医療機関への予約状況の提供ができるようにした。GUIによる操作の簡便化、年度ごとの交付番号の管理、接種券の発行などにより、事務処理の省力化を図った。
- ④接種券の発行などの帳票出力では、ページプリンター用のM言語ドライバを開発し、Windowsを経由しないで直接プリンターに印字することにより、印字スピードアップと、ダイレクト印字位置やフォント指定及びフォームオーバーレイ機能などによる開発しやすく見やすい帳票とした。
- ⑤接種結果登録では、接種状況やツ反結果などの登録により、個人予防接種履歴管理や個人台帳印字を可能とした。また、接種者の確実な把握を行えるようにした。また、学校などでの一斉接種のために、集中入力画面も用意した。
- ⑥予防接種個人台帳では、個人の生涯の予防接種についての履歴が時系列で印字されるようにした。

5. 結果

- ①従来、人手では困難であった予防接種もれ者の確実な把握が可能となった。
- ②システム化により、個人接種という細やかさと事務作業の軽減を両立することが可能となった。
- ③接種希望登録と医療機関の予約管理を一体化することにより、希望日の調整という住民

へのサービスの充実と、スケジュール管理という医療機関へのサービス充実を同時に達成できた。

④予防接種対象者名簿、接種券、医療機関ごとの接種希望者リスト及びスケジュール管理表、予防接種個人台帳などの多くの帳票により、進捗状況や結果がわかりやすくなった。また、これらにより、より安全かつ確実な予防接種事業が可能となった。

6. おわりに

一市町村内での予防接種システムとしては、ほぼ完全なシステムが完成したものと考えている。今後は、他市町村との移動の際の情報の引継や広域化が達成されることが望ましい。

技術的にも、M言語のみで短期間の開発が可能であったし、操作性が良く、スピーディできれいな帳票出力というレベルの高いシステムで満足度の高い結果が得られた。今後は、蓄積されたデータベースの分析により予防接種事業のさらなる安全性や効率性の向上へのヒントが得られるものと期待される。

予防接種対象者名簿(条件指定) 1997.05.07

接種種類等 ツ反(乳児) Quit

生年月日 H07.04.01 ~ H09.03.31

性別 両方

地区指定 1) 熊野 ~ 55) コーポ陸合

印刷中: 129682

一覧印刷

図1 予防接種対象者名簿(条件指定)画面

予防接種対象者名簿 発行日: 2009.08.07 Page: 3

接種種類: ツ反(乳児) 性別: 両方 対象人数: 男: 06 女: 05 計: 104

対象児生年月日: H07.04.01 ~ H09.03.31 地区: 熊野 ~ コーポ陸合

生年月日	対象者氏名(番号)/住所	保護者氏名	町内会	電話番号
H04.02.18	森田 隆雄(森田町5番地2)		吉川第一	74-8888
H04.02.20	森田 隆雄(森田町5番地2)		上間次	74-8888
H04.02.23	森田 隆雄(森田町5番地2)		宝沢	74-8888
H05.03.14	森田 隆雄(森田町5番地2)		徳沢第二	74-8888
H08.03.23	森田 隆雄(森田町5番地2)		沼山第二	74-8888
H08.04.22	森田 隆雄(森田町5番地2)		吉川第一	74-8888
H08.04.27	森田 隆雄(森田町5番地2)		清和第三	74-8888
H08.04.28	森田 隆雄(森田町5番地2)		上間次	74-8888
H09.04.23	森田 隆雄(森田町5番地2)		入間第三	74-8888
04.24	森田 隆雄(森田町5番地2)			74-8888

図2 予防接種対象者名簿

希望入力・接種券発行指示 1997.06.07

姓名: 西川 太郎 男 S15.06.26 56才

希望接種名 三混I期1回目 登録済 Quit

希望医療機関名 西川町立病院

希望接種日 H09.05.07 接種時間帯 PM

保護者氏名 西川 春男 電話番号 9-1234

交付番号 3 本年度発行済: 3

計画参照 登録 接種券発行 削除

図3 希望入力・接種券発行指示画面

予防接種券 三混I期1回目

医療機関名 西川町立病院

平成 9 年度三混I期1回目予防接種

接種券

交付番号	0000000000000000	接種券番号	0000000000000000
接種種類	三混I期1回目	接種場所	西川町立病院
接種日	H09.05.07	医師名	●●●●●●
接種時間	PM	接種状況	接種済
接種場所	西川町立病院	ツ反結果	ツ反判定
接種券種類	三混I期1回目	ツ反備考	硬結
接種券発行日	H09.05.07	ツ反備考	二重発赤
接種券発行時間	PM	ツ反備考	水泡

上記の欄に於いて三混I期1回目接種券を発行して下さる。

平成9年5月7日

西川町立病院 印

図4 接種券

接種結果登録 1997.05.07

姓名: 西川 太郎 男 S15.06.26 56才

接種年月日 H09.05.07 PUSH 保護者氏名 西川 春男 Quit

接種種類等 三混I期1回目 接種場所 西川町立病院

医療機関名 西川町立病院 医師名 ●●●●●●

接種状況 接種済み ワクチンLot-No 123-456

り患 小児マヒにり患
結核にり患
百日せきにり患

ツ反結果 ツ反備考 硬結
二重発赤
水泡

ツ反判定

特記事項 軽度の発熱があった。

登録 個人台帳 削除

図5 接種結果登録画面

予防接種個人台帳

対象児氏名(苗子): ニシカワ タロウ 生年月日: S15.06.26

対象児氏名: 西川 太郎 性別: M

町内会: 清和第三
住所: 清和5-4-3番地8
電話番号: 0-1234

保護者氏名: 西川 春男

接種年月日	接種種類等 接種場所	医療機関	医師名	接種状況	Lot-No
H09.04.08	ツ反判定(中1) 熊野病院	熊野病院	森田 隆雄	接種	242-1 陰性(-)
H09.04.08	B.C.G接種(中1) 熊野病院	熊野病院	森田 隆雄	接種	243
H09.05.01	ツ反(中2) 熊野病院	熊野病院	森田 隆雄	接種	261
H09.05.02	ツ反判定(中2) 熊野病院	熊野病院	森田 隆雄	接種	262-2 弱陽性(+)
H09.05.08	ツ反判定(中2) 熊野病院	熊野病院	森田 隆雄	接種	262-3 中程度陽性(++)
H09.05.08	ツ反判定(中2) 熊野病院	熊野病院	森田 隆雄	接種	262-4 強陽性(+++)
H09.05.08	ツ反判定(中2) 熊野病院	熊野病院	森田 隆雄	接種	

図6 予防接種個人台帳

光カードを利用した診療支援システムの構築

大門宏行⁽¹⁾, 大櫛陽一⁽²⁾, 白男川史郎⁽³⁾, 平井浩⁽¹⁾

診療システム研究フォーラム⁽¹⁾, 東海大学医学部⁽²⁾, 熊本県医師会⁽³⁾
診療システム研究フォーラム 〒102 東京都千代田区麹町3-2-4
Tel (03)-3265-2266 Fax (03)-3265-0055

1. はじめに

医療・福祉分野の情報化は、大学病院、大病院を中心に進められ患者サービス向上、医療スタッフの事務軽減、新技術の情報の交流に大きく貢献している。しかし我々がよく利用する町の診療所では、システムの導入コストや運用スタッフの問題により情報化が浸透していないのが現状である。

我々は、病-診の連携、保健福祉との情報の連携には、大規模なオンラインシステムで構築された情報ネットワークよりも、患者本人が持参することで情報連携が可能な「光カード」を利用したオフラインシステムが有効であると考えた。そこで昨年6月より六つの県の医師と研究会を発足させて、共同で「光カードを利用した診療支援システム」の開発を進めているので、これを報告する。

2. システムの概要

このシステムは「診療支援システム」と「医事処理システム」とを院内LANで結んで、診察室と事務室とを双方向で情報交換を行うシステムである。またエンボスカードや一般的な診察券に代えて診療データを継続的に記憶している「光カード」を利用することによって、かかりつけ医や診-診、病-診連携に役立てることが可能である。

3. システムの構成

機器構成および開発環境は次の通りである。

プラットフォーム: Compaq DESKPRO 4000 (Pentium 166Mhz 32MB)
Compaq DESKPRO 2000 (Pentium 120Mhz 32MB)
NEC PC9821 V13(Pentium 166MHz 32MB)
OS: Windows 95
周辺機器: 光カードリーダーライター、プリンタ、モデム
開発言語: VisualBasic 4.0, OpenM 7.03 for Windows 95 and NT

基本構成では3台のPCより構成され、1台のPCは医事処理システムであり、患者登録、会計処理、請求処理を行う従来のレセプトコンピュータの機能に加え以下の機能を強化している。

- (1) 「光カード」へのデータ入出力機能
- (2) 診療情報から会計情報への変換機能
- (3) 管理帳票機能の強化

2台目のPCは診療支援システムであり、今回新たに開発されたシステムである。現場にたずさわる医師の意見を基に構築し、以下の特徴を持っている。

- (4) コンピュータに不慣れな医師でも使える段階的なメニュー設計およびタッチパネル式モニタに対応した画面設計
- (5) 医事処理システムとの情報連携機能
- (6) 診療支援・患者指導機能の充実
 - ・紙のカルテと同様な表示機能
 - ・光カードによる他医療機関との連携
 - ・投薬履歴の一覧表示
 - ・検査結果の時系列グラフ
 - ・画像登録機能（正常画像、医療画像）
- (7) コード体系、光カード記憶形式の標準化技術の採用

最後のPCはサーバ専用機であり、操作の簡便化とデータ保護の強化のために使われる。

4. おわりに

現在このシステムは、協力医療機関にモニタシステムを設置し機能面や操作性の評価を行っている。今後、評価結果もとにシステムを更に充実させていく予定である。

Mを利用したシステム開発は今回が初めての経験であり、他のPCベースのデータベースシステムに比べ、非常に高速でしかもコンパクトなシステムであるというのが第一印象である。またVisualBasicとの相性についても、お互いに文字処理系を得意とする言語なので可変長データを操作できるので非常に都合がよい。

今回の開発では、VisualM 経由でDo コマンドを発行し、結果をテキストファイル経由で受け渡す方法をとった。開発過程では GUI 部分とデータ抽出部分を独立して開発する事ができ、しかもこの方法がもっとも早いレスポンスでデータの受渡ができた。

ネットワークへの対応も、Mのマルチユーザの実現という基本設計により容易に対応でき評価できるものであった。

なお開発比率はVisualBasic が約7割、Mが3割程度であり、GUI 部分の比重が非常に高い結果となった。

M/WebLinkを用いた、 マルチメディア電子教科書

岡田好一、春木康男、大櫛陽一
東海大学医学部、医用工学情報系

東海大学医学部： 〒259-11 神奈川県伊勢原市望星台
tel: 0463-93-1121, fax: 0463-96-4301
E-Mail: PFB00710@niftyserve.or.jp

Abstract:

A multimedia textbook of urology has been made and evaluated using M/WebLink - middleware between OpenM (M language) on Windows NT and Web browsers.

Because the textbook contains many links (about 85,000 links in 20,000 lines), the ordinaly hypertext can not represent whole of them. We use a search system on Web pages using M/WebLink to show simple and useful multimedia textbook for medical students and medical practitioners.

Keywords: web browser, electronic textbook, multimedia, satellitic line

1. はじめに

M/WebLink は、M 言語で HTML 文書を構成し、Web ブラウザに送り、また、HTML のフォームのデータを M 言語に送る、いわゆるミドルウェアに相当するソフトウェアである。

機能的には CGI(Common Gateway Interface) と同等であるが、C 言語や Perl 言語を使わず、直接 M 言語で Web ブラウザの機能が利用できる。

構成的には、Microsoft Windows NT と同社の IIS(Internet Information Service)、Intersystems 社の OpenM(M 言語)を必要とする。

2. システムデザイン

用いた泌尿器科教科書は、B5 版、639 頁のもので、約 1.4MB の日本語の文書と、662 枚の図版からなる。

図版は表や線画のイラストと、X 線写真などのグレースケールの必要なものに分類できる。前者はビットを落とした上、GIF 形式で保存し、後者は JPEG 圧縮を行った。

約 4,000 の日本語の医学用語と、6,000 の英語のキーワードは、本文に対し、約 85,000 のリンクを構成するので、通常の HTML のリンクでは表現できない。そこで、M 言語の検索能力を利用した。

本システムは、NTT と東海大学の共同研究である、静止衛星利用のマルチメディア共同実験の一環としてデザインしている。この実験では、Web ブラウザが良好に稼働しているため、Web ブラウザが利用できる M/WebLink を利用した。衛星回線は、DB からの転送速度が大きい(30Mbps)

イントラネットとして利用できる。

3. システムの現状

プログラムは、若干の固定の HTML 文書を除き、すべて M 言語で記述されている。

M 言語が Web ブラウザに送る HTML 文書には JavaScript などの他の言語を用いていない。

電子教科書としての効果を見るため、学生アルバイトを対象としたフィールド実験を行っている。

4. 技術的な特徴と問題点

Web ブラウザを端末画面として利用する場合の、ビジュアルな開発用ユーティリティがなく、複雑な画面を短時間で構成するのは、現状では無理がある。

Web ブラウザの制限を受けるので、たとえば、ファンクションキーが利用できない、タブ順が指定できないなどの問題点がある。

Web ブラウザにはキャッシュ機能があるので、前の画面にさかのぼって操作した場合の配慮が必要となる。

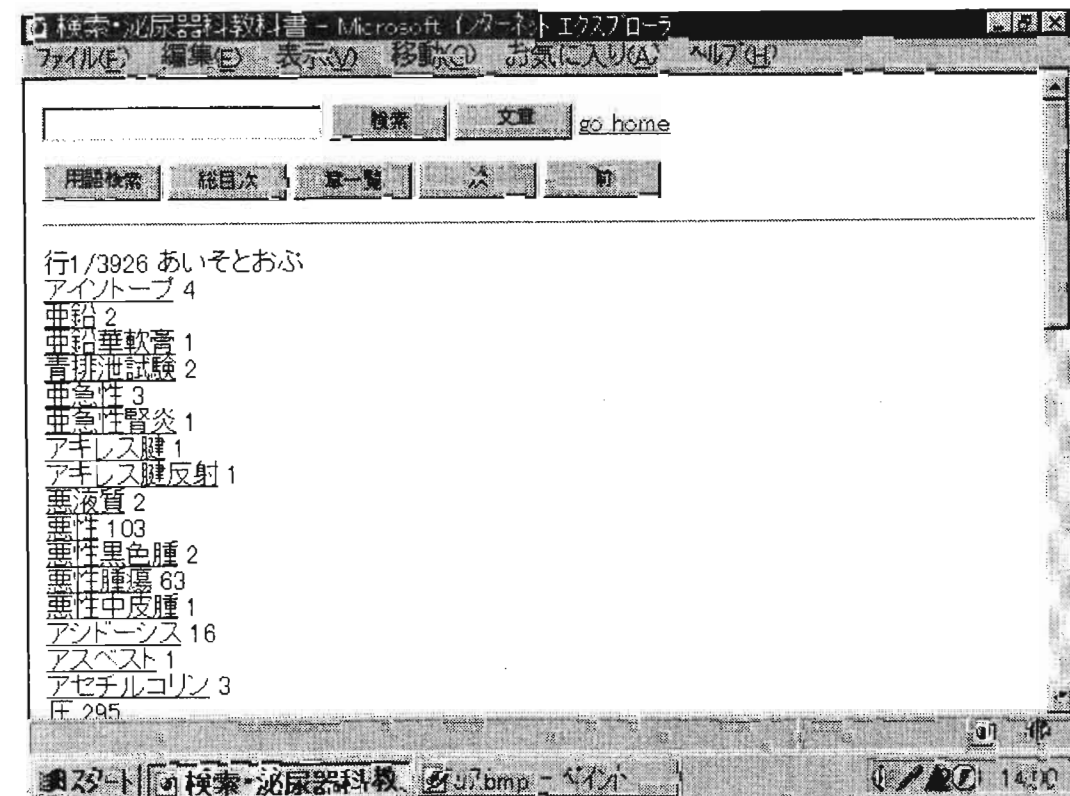
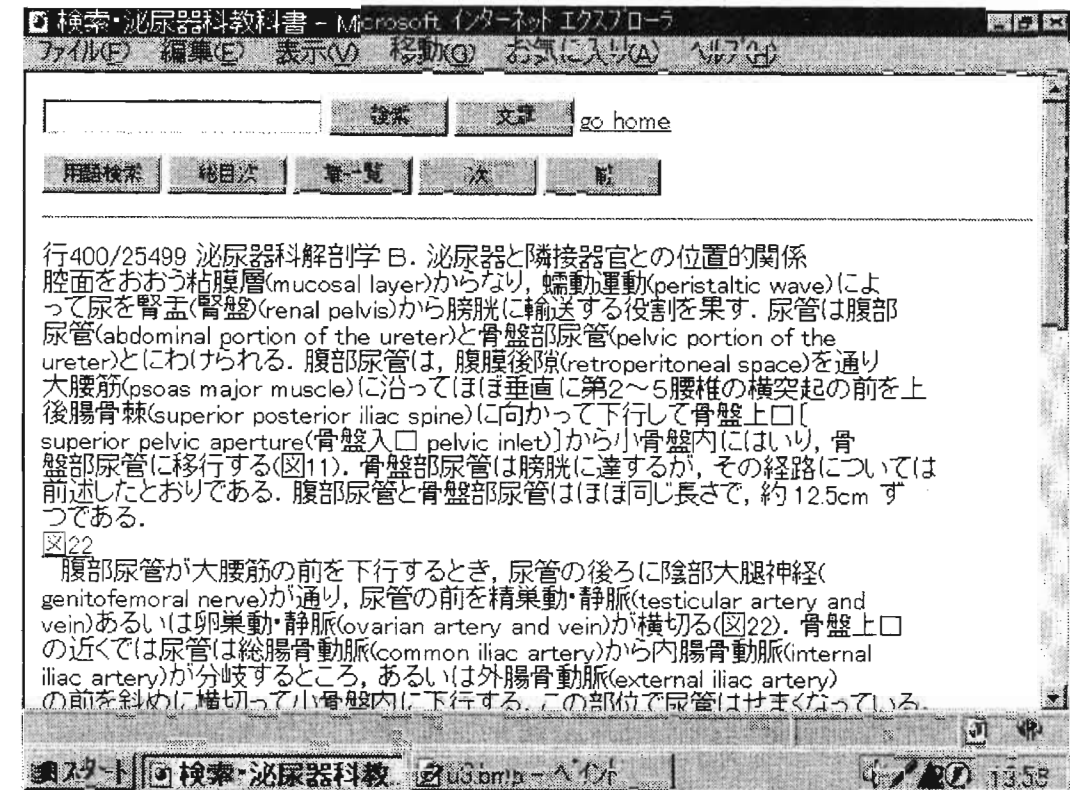
ホームページの見慣れた画面に類似しているので、利用者に親しみやすい印象を与える。

ウィンドウシステムの教育効果の高さはそのまま維持される。

OS や M 言語以外の開発系からの影響が、比較的小さいと思われる。

5. 今後の課題

Web の機能を生かすことのできる、開発の指針の確立と、ユーティリティの充実が望まれる。



検索・泌尿器科教科書 - Microsoft インターネット エクスプローラ

ファイル(F) 編集(E) 表示(V) 移動(Q) お気に入り(A) ヘルプ(H)

検索 文章 go home

用語検索 総目次 章一覧 次 前

#	件数	検索文字列
#1	63	悪性腫瘍
#2	27	腎臓
#3	51	腎動脈
#4	4	#1 & #3

一つ削除

スタート 検索・泌尿器科教科書 u&bmp - ペイント 14:01

検索・泌尿器科教科書 - Microsoft インターネット エクスプローラ

ファイル(F) 編集(E) 表示(V) 移動(Q) お気に入り(A) ヘルプ(H)

検索 文章 go home

用語検索 総目次 章一覧 次 前

z1122.gif

大動脈裂孔 腹大動脈 食道および食道裂孔 後枝(内側枝) 下横隔動脈 前枝(外側枝) 上腹動脈 上下横隔静脈 中腹動脈 胸腹静脈 上腹動脈 腎動脈 腎静脈 腎静脈の反折部 (尿管静脈へ)

右腎動脈 右腎 右尿管 右尿管 右尿管

大静脈系と下大静脈 右下横隔静脈

静脈(あるものは単に静脈と尿管枝(尿管静脈へ))

スタート 検索・泌尿器科教科書 u4bmp - ペイント 13:58

検索・泌尿器科教科書 - Microsoft インターネット エクスプローラ

ファイル(F) 編集(E) 表示(V) 移動(Q) お気に入り(A) ヘルプ(H)

検索 文章 go home

用語検索 総目次 章一覧 次 前

行1/4

- §1.C.C. 泌尿器の画像解析解剖学的所見
- §2.B.1.d. 後腎(腎臓)
- §3.1.E.E. 腎クリアランス
- §8.1.A.e.e. 予後

次

スタート 検索・泌尿器科教科書 u9bmp - ペイント 14:03

検索・泌尿器科教科書 - Microsoft インターネット エクスプローラ

ファイル(F) 編集(E) 表示(V) 移動(Q) お気に入り(A) ヘルプ(H)

検索 文章 go home

用語検索 総目次 章一覧 次 前

z1123.jpg

腎臓 腎臓 腎臓

図23 後腹膜気体注入法(PIR)による腎臓のX線像

スタート 検索・泌尿器科教科書 u1.bmp - ペイント 14:13

MとWWWの連携

= M言語を用いたWWW文書作成について =

八川 剛志, 岡田 康 (住友電気工業株式会社)

〒542 大阪府大阪市中央区南船場 4-11-28(サン船場ビル)

Tel: 06-258-5519 Fax: 06-258-5520

hachi@apsys.ssy.sei.co.jp

yokada@apsys.ssy.sei.co.jp

Abstract: Nowadays, WWW server provides dynamic WWW pages using a database. On the other hand, Mumps has a characteristic of fast search, but U-MUMPS doesn't have the so-called "Internet/Intranet function". So, we have developed Mumps-CGI(M-CGI), which provides CGI (Common Gateway Interface) between WWW server and U-MUMPS. As a result, we can refer U-MUMPS data by using WWW browser.

Keywords: WWW server, CGI, U-MUMPS, M-CGI

1. はじめに

インターネット/イントラネット技術が急速に浸透し、WWWサーバがデータベースと連携しながら、動的にWWWページを構成する事が必要となっている。データベースシステムMumpsは、高速に検索が行えるという特徴を持つため、インターネット上の検索エンジンとして適しているにも関わらず、U-MUMPSはいわゆる"インターネット連携機能"を持っていなかった。そこでU-MUMPSに、WWWサーバと外部プログラムのインターフェース(CGI)の機能を付加するモジュール(M-CGI)の開発を行ったので、報告する。

M-CGIとは、WWWページ内容をMルーチン内にプログラムする事により、Mデータ又はその加工結果を含んだページをブラウザ上に表示させる事を可能とするモジュールで、以下の2部分より構成される。

①UNIX上で動作し、WWWサーバより呼び出される部分(以下、M-CGIコマンド)

②WWWページ内容をプログラム化したMルーチン(以下M-CGIルーチン)

なお、M-CGIはU-MUMPS Ver2.3.0以上でサポートされているストアプロシジャ(以下MSP)機能を用いている。

2. MSP(Mumps Stored Procedure)とは

ストアプロシジャとは、リレーショナルデータベース(RDB)の分野で用いられる用語で、データ操作を行う一連のコマンド群をデータベースシステム上で定義し、クライアントから呼び出して結果を得る手続きの事である。そのストアプロシジャを、U-MUMPS上で実現する手法をMストアプロシジャ(MSP)と呼ぶ。

本手法を用いる為には、U-MUMPS上でMSPサーバジョブが動作している事、及びこれにア

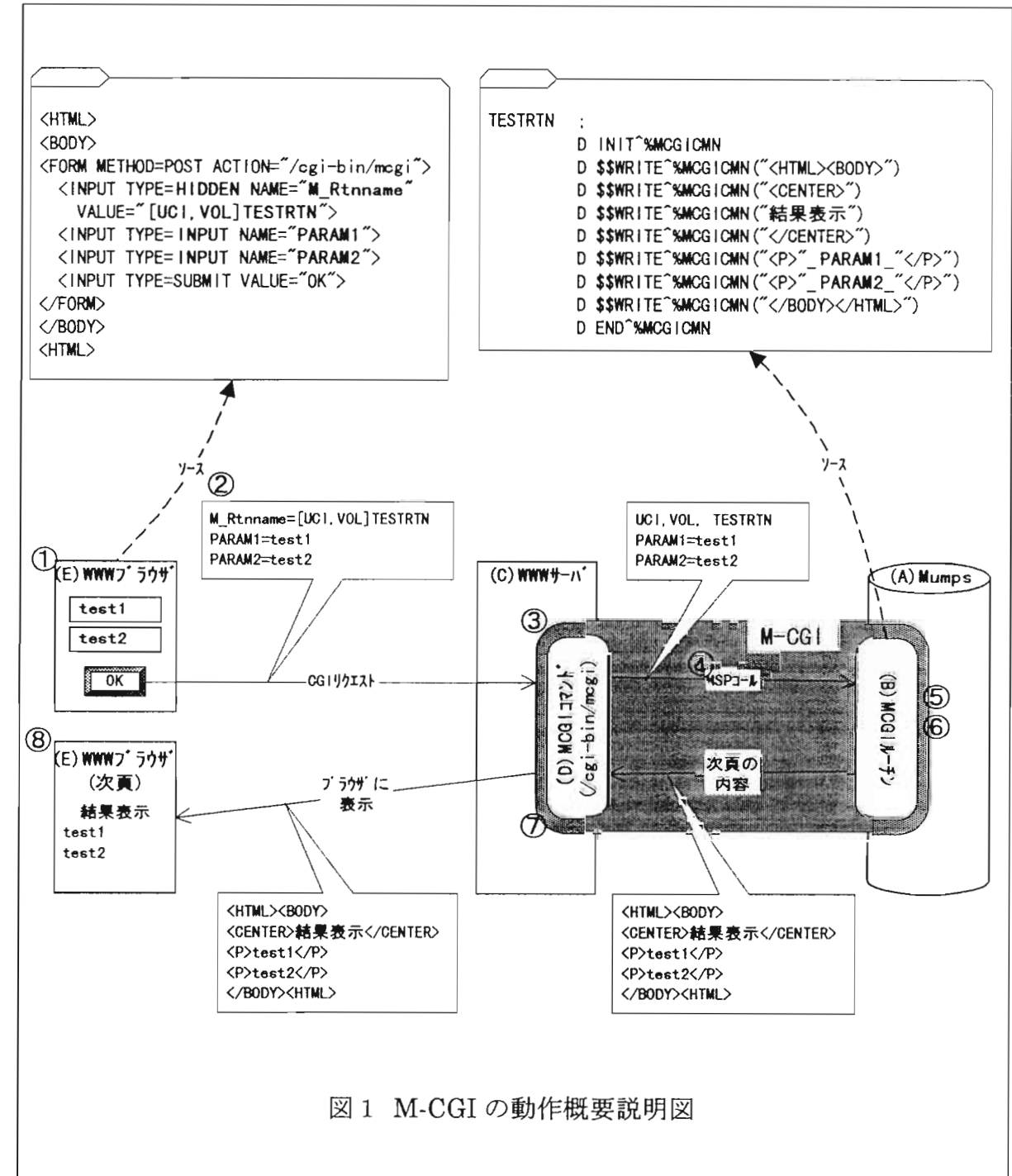


図1 M-CGIの動作概要説明図

クセスするクライアントモジュールが必要である。MSPによるルーチン呼出しは、以下の3手順からなる。

①Mumpsに対して「起動ルーチン名」「起動引数」を与える。

②Mumps上で指定のルーチンが起動され、一連のデータベース処理を行う。

③実行結果が得られる。

MSPは、従来の1ノード毎のデータベースアクセス(DDP)に比べ、

①データ検索処理を高速に行える

②ネットワーク負荷が少ない

という特徴を持つ。

3. M-CGI の構成、動作概要

M-CGI を動作させるのに必要な構成は、以下の通りである。(図 1 参照)

・データベース

A) U-MUMPS(Ver2.3.0 以上、MSP サーバジョブが動作している事)

B) M-CGI ルーチン

・ WWW サーバ (UNIX)

C) WWW サーバソフト

D) M-CGI コマンド

・ 端末

E) WWW ブラウザ

以上の構成のうち、WWW ページ作成者が用意する物は(B)M-CGI ルーチンのみであり、(A)(C)(D)(E)はシステム管理者により用意される。

ブラウザが M-CGI を用いてページを表示する際の動作を以下に説明する。(図 1 参照)

①WWW ブラウザに<FORM>タグを用いて CGI を呼び出すページが表示されている。

↓「OK」ボタンを押す

②WWW サーバに、CGI リクエストが送信される。パラメータは次の通り。

| M_Rtnname=[UCI,VOL]TESTRTN

| PARAM1=test1

↓PARAM2=test2

③WWW サーバが M-CGI コマンド(実体は/cgi-bin/mcgi)を起動する。この時、CGI の標準仕様(POST)に従い、送信されてきたパラメータは標準入力より受け渡される。

↓

④M-CGI コマンドは、MSP のプロトコルに従って M-CGI ルーチンを起動する。起動するルーチン名は、受信パラメータの中で "M_Rtnname=" の値として記述されている。(本例の場合は UCI, VOL, TESTRTN。)

| UCI, VOL, TESTRTN

| PARAM1=test1

↓PARAM2=test2

⑤起動された M-CGI ルーチンは、受信したパラメータを、変数として利用できるように処理を行う(INIT^%MCGICMN 内で行なう)。本例の場合、PARAM1, PARAM2 という変数が利用可能となる。

↓

⑥M-CGI ルーチンで、表示する HTML 文を生成する。その HTML 文を WWW サーバ上の M-CGI コマンドに対して送信する時には、\$\$WRITE ^%MCGICMN 関数を呼び出す。(引数に HTML 文を与える)

| \$\$WRITE^%MCGICMN("<HTML>")

| ...省略...

↓\$\$WRITE^%MCGICMN("</HTML>")

⑦HTML 文を受け取った M-CGI コマンドは、標準出力にその内容をそのまま出力。CGI の標準仕様に従って、標準出力された内容が WWW サーバによって WWW ブラウザに送信される。

↓

⑧WWW ブラウザに表示される。

4. M-CGI ルーチンの記述方法

図 1 の右上にある TESTRTN のサンプルプログラムに示すように、M-CGI ルーチンの記述ルールとして以下の 3 つを定めている。

表 1 従来の方法との比較

	M-CGI	ファイル渡し
HTTP 同時セッション数	MSP サーバジョブ数 (数十本)	4 (HFS の数による制限)
システムの構成	Mumps サーバと WWW サーバの分離が可能	Mumps サーバと WWW サーバの分離が不可能

・ ルーチンの最初に INIT^%MCGICMN 関数を呼び出す

・ HTML 文を出力する時には \$\$WRITE ^%MCGICMN 関数を用いる

・ ルーチンの最後に END^%MCGICMN 関数を呼び出す

なお、サンプルプログラムの 7~8 行目で用いられている PARAM1, PARAM2 は、WWW ブラウザより送信されたパラメータの変数名が、そのまま Mumps の変数として利用可能である事を意味している。

5. M-CGI の特徴

U-MUMPS を CGI として用いる為に、ファイル渡しによりパラメータや検索結果のやり取りを行う方法も考えられる。表 1 に、ファイル渡しによる方法と本手法との特徴の比較を示す。ファイル渡しの方法では、U-MUMPS の HFS (Host File Server)の制限により HTTP 同時セッション数が 4 しか取れない問題があった。本手法ではその制限は無くなり、サーバジョブの数を増やす事で対応する事が出来る。また負荷分散の為に、Mumps サーバと WWW サーバを

物理的に別のマシンに分けるシステム構成も採用できる。

6. おわりに

今回は、M-CGI を実現する為に MSP クライアントの UNIX 版作成という手法を用いた。CGI 技術は、動的な WWW ページ作成の為の手法の一つであり、将来的には Java/ActiveX 技術を用いたものの比率が増すであろう。今後は、MSP クライアントの Java 版を作成し、Java を用いて Mumps にアクセスできる機構についても検討を行いたいと思っている。

参考文献：

1. ローラ・リメイ(1995), 続・HTML 入門:プレジデントホール出版

MとWWWの連携による看護過程支援システム

山下 芳範
福井医科大学

Abstract: The new hospital information system has been operating since 1996. This system changed by way of the network all terminals and came to be able to use the internet. The environment and software which uses the internet are largely changed. The environment and software are straightened so that the hospital information system may use these network environments. We make a WWW interface on M database language and use for a application on hospital information system.

Keyword: HIS, Intranet

はじめに

本学では、96年より第3期の新しいシステムが稼動している。このシステムでは、全端末がネットワーク経由に変わり、インターネットも利用できるようになった。また、最近では、インターネットを利用する環境やソフトウェアが大きく変化し、WWWによる情報検索も急速に変化している。特に、これらの技術を企業内システムにおいても取り入れる方向に動いている。

このような状況であるので、本学の病院情報システムでも、これらのネットワーク環境を最大限に利用できる環境を整えてきた。特に、新システムになって、Mの利用も新たな要素も加えて改良を進めている。

この中でも、インターネットで活用されている技術を利用するイントラネットを、Mとの連携を行い、病院情報システムの中で試験的に適用を始めた。病院情報のデータベースとも関連のある看護情報システムへの試験を開始した。

概要

今回は、Mのイントラネットへの応用としてMとWWWとの連携を試みた。WWWは、文字だけでなく画像を含む各種情報が取り扱えるため、今後のデータ利用を考えた場合、適応できる範囲が大きい。病院情報システムでも、現在のデータベースだけでなく、各種の情報をネットワークを通して共有することになる。このような目的で作成されているWWWをMと融合することで、利用範囲も広げることができる。

今回の開発は、WWWを通してユーザーインターフェースを統合することも目的としており、院内で利用できるWWWによる情報利用とともに基幹業務も同様に扱うことの実現である。

Mは文字処理が得意であり、WWWの作成書式であるHTMLのハンドリングも容易行えらるとともに、DB上でも保管可能である。

方法

WWWは、通常のハイパーテキスト(HTML)を表示するだけでなく、外部のプログラムを利用することが可能である。今回は比較的实现が容易であるこの仕組みを利用して、病院情報システムの

データベースとのインターフェースを行っている。

しかし、この接続方法は、WWW上での入力操作からのデータに関してWWWからの特有の情報の解読が必要であるので、これらを容易に取り扱うため、病院情報システムで利用しているM側で解読して、処理が簡略できるように考慮している。また、単なるWWWのインターフェースだけでは、今後の開発の効率化にはつながらないので、Mのプログラムツールとして、FORMやデータベース参照が容易に設計でき、既存データベースの検索なども簡単な定義から自動作成ができるような仕組み(WEBMAN)も利用している。

M側から考えると、WWWが従来のスクリーンの出力先となるので、HTMLに対応したハンドリングを行うとともに、WWWのFORMからの情報は、ボタンなどのイベントによって特定のプログラムが呼び出されるイベント対応するとともに、データが適切な構造体として引き渡される。

看護支援システムでの利用

今回は幾つかのシステムを、試験的に開発を行っている。

WWWの利用として効果的に開発できるものは、患者検索などのような検索を主体とするものである。これらは、WWWインターフェースの仕組みの中では、データベース参照の定義を行うだけで、検索キーや結果の表示まで、かなりの部分を自動生成することができるため、HTMLを意識することなく作成が可能である。

しかし、より現実の業務に適応するため、看護情報システムへの適応を試みた。この応用では、単に情報を検索し表示するだけでなく、必要に応じて患者の情報入力を行う情報収集も含んでいる。

今回のシステムでは、分散している情報を用途に合わせて整理する過程をコンピュータで行い、整理された情報を参照し、分析、統合、判断を加えるという看護診断とその評価の過程を支援するものである。このシステムの主な目的は以下の通りである。

1. 看護婦の経験や能力による差を解消し、最低限必要な患者基本情報をもれなく収集する。
2. 分散している膨大な量の情報から、問題となる情報の整理と一覧を可能にし、看護婦の分析、統合、判断を支援する。
3. 看護計画のベッドサイド活用と入力による記録時間の短縮と情報のフィードバック。

また、このシステムでの機能は以下の通りである。

(1) 患者基本情報の入力

成年期を基本型として、年齢により青年期、老年期、小児期の3期、疾患、症状により呼吸、循環などの9つの系統毎に、各特徴を含んだ情報を追加でき、個別性に応じて必要な項目が網羅され、かつ重複しないように考慮した。初回問診の面接時には年齢別に加えて、この9つの系統別の患者基本情報を選択し、聞きながら入力する。全ての情報は、ゴードンの11の機能別健康パターンに分類し、選択入力できるようにリストアップした。

個別性、特殊性に応じてテキスト入力も可能にし、入力した情報は正常、異常の判別ができるようにした。又テキスト入力できない場合はその分スペースを空けて紙の出力をした後で記入できるように考慮した。さらに、緊急入院等の場合は、必要最低限の情報の入力登録後日追加入力を可能にした。基本的に、入力は順番どおりに行うが、途中で、他の項目(ゴードンの11の機能別健康パター

ンの他の項目)に移る事も可能とした。又、褥瘡のブレードスケールや、意識レベル判定基準、痴呆判定の長谷川式スケール等は、その内容をウインドウ表示し、必要に応じてその場で参照しながら入力できるように考慮した。

(2) 異常な情報の整理と看護診断支援

患者基本情報として入力し異常と判別された情報を、ゴードンの11の機能別健康パターン毎に、「問題情報リスト」として明示する。正常と判別された情報や、テキスト入力したもの、ワープロ入力した情報も看護婦の判断により、看護診断する上で問題となるものは、参照情報として整理して「問題情報リスト」に明示できるようにした。同時に患者基本情報とゴードンの一つ一つの診断ラベルの定義上の特徴を結び付ける事で、診断ラベルが導き出せるようにした。この「問題情報リスト」と、導き出された診断ラベルのリストを参照し、個々の看護婦が、分析、統合し、判断を加え診断をする。

(3) 標準看護計画の選択

診断ラベル毎に、当院で作成した標準看護計画が表示され、必要な計画を選択できるようにした。ここには原因又は関連因子、解決目標、具体的計画が列挙して表示されており、これらは個別性に合わせて選択又はコメント入力し追加修正する。計画を選んだ理由も「問題情報リスト」に明示した情報より選択できるようにした。又計画は、より具体的内容とするために看護手順等を参照できるように考えている。さらに観察内容についてはフローシートに評価基準を決めて表示し、ベッドサイドで入力し評価し易くした。

(4) 情報のフィードバック

看護計画はベッドサイドでも参照できるように、病院情報の端末だけでなく、携帯型の端末でも利用できる。観察内容はフローシートで随時入力でき、項目毎に歴として参照し、計画の評価に活用する。又看護計画の評価は、その計画を導きだした情報がどう変化したかによって評価し、変化した情報は、患者基本情報にフィードバックさせ、計画の終了や新たな問題提起により改めて看護診断を行い看護展開する。患者情報から心理的・精神的な診断ラベルを導き出す事は不十分なので、心理的な事柄や一時的な問題等、その時点で計画に含まれていない新たな問題は、プログレスノートに記載し蓄積する。

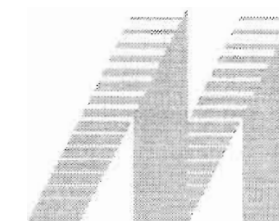
まとめ

今回は試験的に実際の病院情報でのアプリケーションの作成を行っているが、WWW自身がページ単位で動作するため、入力項目ごとのアクションなどについては、クライアント側での特別な処理を利用しない限りは、プログラムの工夫やコーディングも増加するので、単純な作成を行うという観点では不向きと考えられる。

また、最新の機能(ActiveXやJava等)を利用するためには、スクリプト等や実行モジュールを埋め込むことが必要となる。

しかし、検索を伴う画像表示などは、特別なことを行わなくとも可能であるので、比較的入力が単純な部分への応用は容易に行える。

また、M言語は、HTMLのような文字列処理を要求されるものに対しては、非常に効果的に利用できるとともに、データベースでも取り扱えるので、今回の試験でも容易に実現が可能であった。



Visual M とビジュアルコンポーネントとの結合

Visual M and Visual Component Combination

佐藤比呂志

Hiroshi Sato

日本デジタルイクイップメント(株) 西日本第一統合システム部

Digital Equipment Corporation Japan Network and Systems Integration Services

〒530 大阪市北区中之島2丁目2番2号ニチメンビル TEL (06) 222-9211 FAX (06) 222-9408

This is a trial implementation to extend the existing visual components of Delphi to have access to M database. By using Visual M, We can build a GUI based application to access M Database with a GUI tool such as Visual Basic and Delphi. But, Visual M supplies only a basic interface to access M resources (M Database and M routines and M functions etc.) from a GUI tool. So usually we have to write a glue logic to connect Visual M and a GUI tool on both side of GUI tool and M server. This introduces some kind of programming difficulty (how to separate a Server logic and a Client logic, debugging difficulty etc.) which we have not seen in CHUI based M applications. I think there are two approaches to reduce these difficulties. The one is to develop automated code generator tool such as Visual Architect that is developed by Atlas Development Corporation. The other is to add an encapsulated access logic to M database into commonly used Visual Component such as ListBox, ComboBox and Label controls. This time I have tried the latter one.

キーワード: Visual M, M, MUMPS, Delphi, DASL, VCL, Visual Component, Visual Basic, Visual Architect

1 はじめに

Visual Mによって、M言語による GUI ベースのクライアント・サーバー・アプリケーションの開発が可能になった。しかしながら、Visual Mは、M言語と GUI ツール(ビジュアルベーシック等)をネットワーク経由で繋げるミドルウェアにすぎない。従って、統合化された純粋なM言語開発環境(キャラクター・ユーザ・インタフェース)と比較すると、開発ノウハウの不足、処理の分離(M側または GUI ツール側)によるオーバーヘッド増加、デバッグの難しさ等の要因によるシステム開発工数の増加が散見される。今回、Visual M アプリケーション開発工数削減の方策として、GUI 部品(テキスト、リストボックス等)を拡張し、Mグローバルを直接それらの GUI 部品に結合する方法を試作したので、ここで紹介してみたい。

2 Visual M による開発の問題点

Visual M による開発の効率を妨げる要因として以下のものが考えられる。

2.1 開発ノウハウの不足

ビジュアルベーシック等の GUI ツールは、非常に柔軟性のある高機能ツールである。しかし、このことは、逆にいうと、自由度が大きすぎることを意味する。つまり、同じ処理をさせるのに開発者によって、様々な方法、記述が可能である。システム開発が大規模になり、多数のプログラマが関わるにつれ、これらの統一性のなさが、様々な不整合の要因、将来の保守工数の増大の要因ともなりうる。

2.2 処理の分離の問題

Visual Mのアプリケーション構造は、クライアント側、サーバー側双方に異なる言語処理系が存在する構造である。(M言語と Basic 言語など) 従って、同じ処理をするにも、クライアント・ヘビー(ファット・クライアント)、クライアント・ライト(シン・クライアント)の間で様々な形態を取りうる。

また、2つの言語間の違いにより、データ交換に伴うオーバーヘッド処理が必要になる。(型変換、配列の違い、構造体データの引き渡し等)

2.3 デバッグの難しさ

アプリケーションの構造上、M側のロジックをインタラクティブにデバッグするのは非常に困難である。

3 Visual M 開発効率向上の方法

一般的に以下の様な方法が考えられる。

3.1 開発ノウハウの蓄積

よく使用するロジックは、共通ルーチンまたは、クラス化して共有できる様に公開する。

プログラミング・ガイドラインを作成し、開発者に周知徹底し、開発者間の実装の違いを最少化する。

OCX(ActiveX)等のコンポーネントを作成し、よく使用するロジックを隠蔽し、開発者に公開する。

3.2 処理の分離

Visual M アプリケーションを生成するジェネレーターを作成し、サーバー・クライアント処理の分離を自動的に行う。(Atlas 社の Visual Architect, Freedom 等)

4 Delphi VCL を使用した Visual M 部品の作成

Visual M アプリケーションの個々の要素を分析してみると、各 GUI 部品にMの要素を結合する際に、ある一定パターンで処理が行われていることがわかる。

そこで、これら頻繁に行われると思われる処理をコンポーネントとしてカプセル化し、利用時にプロパティの設定、およびメソッドの実行による処理の簡略化ができる様、実装を試作した。

今回の実装のツールとして Delphi を使用した。Delphi を選択した理由は、以下の通りである。

●オブジェクト指向言語として非常に良く実装されており、既存の GUI 部品を簡単に拡張できる。

●Globalware 社が Delphi VCL(Visual Component Library)による Visual M コンポーネントの実装サンプルを提供している。

5 試作コンポーネントの構成

5.1 開発環境

サーバー

Open M NextGen for WindowsNT/95 V2.0 Beta2

クライアント

Delphi V2.0 + Visualm.dll (Visualm.ocx)

5.2 構成図

図1を参照

6 拡張 M VCL の仕様

6.1 メタデータ定義

Mグローバルと各 VCL を結合する際に、メタデータ定義があると、プログラミング時の設定項目を削減することができる。

今回、弊社の DSM 4GL 製品である DASL(DSM Application Software Library)のデータ定義を参考にし、以下の様なメタデータ定義を規定した。なお、以下の定義は、実際の DASL のデータ定義とは構造が異なることをお断りしておく。

```
^DDN("Dataname", DNAME)      DNAME = データ名
                                ,"Global Reference")
                                ,"Validation")
                                ,"Piece")
                                ,"Extract")
                                ,"Input Transform")
                                ,"Output Transform")
                                ,"Cross Reference")
                                ,"Primary Key")
                                ,"Description")
                                ,"Data Type")
                                ,"Data Length")
                                ,"Attribute")
```

6.2 コントロール仕様

1) 拡張 M Edit Control

追加プロパティ

DNAME	データネーム
Namespace	ネームスペース
Error	Visualm.Error
ErrorMessage	Visualm.ErrorMessage

追加メソッド

Fetch DNAME の内容を Text に持ってくる

Update Text の内容を DNAME に設定する
(この際にバリデーションをチェック)

追加イベント

KeyPress Return key が押された時には、
バリデーションのチェックを行う。

2) 拡張 M ListBox Control

追加プロパティ

DNAME	データネーム
PKEYNAME	プライマリ・キー・ネーム
Namespace	ネームスペース
Error	Visualm.Error
ErrorMessage	Visualm.ErrorMessage

Selectmask 選択マスク
(A*, ABC*, *a*, *S など)

追加メソッド

lookup DNAME の内容をリストボックスに持ってくる

追加イベント

Click 選択された行に対応する DNAME およびその PKEYNAME に値を設定 (ローカル変数)

3) 拡張 M Label Control

追加プロパティ

DNAME データネーム
Namespace ネームスペース
Error Visualm.Error
ErrorName Visualm.ErrorName

追加メソッド

Fetch DNAME の内容を Caption に持ってくる

4) 拡張 M CheckBox Control

追加プロパティ

DNAME データネーム
Namespace ネームスペース
Error Visualm.Error
ErrorName Visualm.ErrorName

追加メソッド

Fetch DNAME の内容を CheckBox に持ってくる。(TRUE or FALSE)

Update CheckBox の現在の状態で DNAME の内容を更新

5) 拡張 M ComboBox Control

追加プロパティ

DNAME データネーム
PKeyName プライマリ・キー・ネーム
Namespace ネームスペース
Error Visualm.Error
ErrorName Visualm.ErrorName

追加メソッド

lookup DNAME の内容を Text に持ってくる

追加イベント

KeyPress Return の時、Text の値を含むデータのみを選択する

Click 選択された DNAME およびその PKEYNAME に値を設定 (ローカル変数)

6) 拡張 Memo Control

追加プロパティ

DNAME データネーム
Namespace ネームスペース
Error Visualm.Error
ErrorName Visualm.ErrorName

追加メソッド

Fetch DNAME の内容を Text に持ってくる

7 おわりに

今回の試作は、実際の開発現場で使用するには、機能、パフォーマンス・チューニング両面でまだ検討の余地がある。また、今回の部品は、現時点では、Delphi 環境でのみしか利用できない。しかし、同様のことは、より汎用性のある ActiveX Control (OLE Custom Control) または、Java Beans として実装できるはずである。Visual Basic バージョン 5 より、VB 自身で ActiveX Control を開発できる様になり、従来に比べて ActiveX Control 開発の敷居が低くなった。(従来は、C++ で開発する必要があった。) また、Delphi バージョン 3 から VCL を Active X Control に変換する機能が追加される様である。コンピューティング・パラダイムがネットワーク・セントリックに移行するなか、コンポーネント (部品) 化、オブジェクト化は、必然の流れであり、M もこの流れから外れることはできないだろう。今後も引き続きこのテーマで模索を続けたいと考えている。

参考

- [1] Visual Architect: Atlas Development Corporation
- [2] Visual M Delphi VCL: Globalware Corporation

図 1

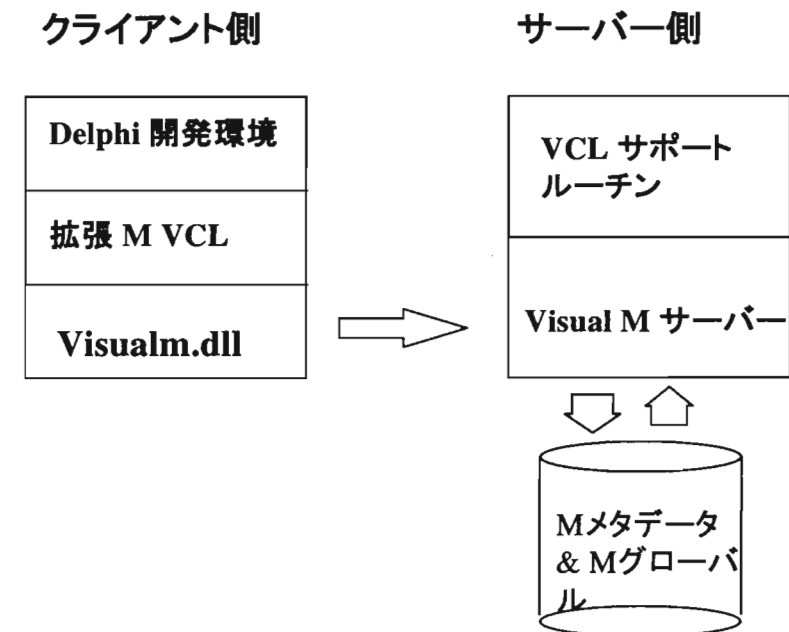


図 2 拡張 M Listbox の例(開発画面)



V B と O p e n M による保健指導システムの開発

大櫛陽一¹⁾、嶋芳成²⁾、西満里子³⁾

1) 東海大学医学部医学情報学：〒259-11 神奈川県伊勢原市望星台

Tel: 0463-93-1121 Fax: 0463-96-4301 youichi@keyaki.cc.u-tokai.ac.jp

2) 日本ダイナシステム株式会社：〒460 名古屋市中区新栄2-1-9 雲竜ビル東館

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

3) 岐阜県国府町保健福祉会館：〒509-41 岐阜県吉城郡国府町木曾垣内650

Tel: 0577-72-3853 Fax: 0577-72-3671

Abstract: We have developed a health consultation system for public health nurses. Visual BASIC is used for Graphic user interface and Open-M is used for database system and main programming language. Data and parameters are input with a screen made with Visual BASIC. These data are set as a text file. An Open-M routine reads the file and accesses the database and the results are set another text file. Another Visual BASIC program reads the file and display the results. The combination of Visual BASIC and Open-M has some advantages.

Key words: Health consultation, Public health nurse, Graphic user interface, MUMPS, M technology, Open-M, Visual BASIC

1. はじめに

日本ダイナシステムでは、市町村の保健福祉システムとして、DTMで開発したHⁱⁿHabitと名付けられたパッケージソフトを持っており、飛騨高山地区を中心に多くのユーザがいる。このシステムは、健診の自動データ収集や職種間のデータコミュニケーションなどのすぐれた機能と性能を有している。しかし、過去のハードウェア資産の継承を強く意識してきたために、マンマシンのGUI化や住民に見せて指導する画面などの開発が遅れ気味であった。

今回、蓄積されたデータの活用として、グラフィカルな画面を多用した携帯型保健指導

システムを開発したので報告する。

2. システムの機能

開発したシステムでは、以下の機能を持っている。

①ユーザごとのセキュリティ管理

②氏名からの住民検索

③家族からの個人検索

④個人の健康履歴の表示

⑤帳票形式でのデータの表示

問診、健診結果、精検結果

⑥表形式でのデータ表示

血液検査結果、尿検査結果など

⑦保健活動の登録と表示

健康相談、健康指導など

⑧グラフ形式でのデータ表示

折れ線グラフ、散布図、レーダチャート、度数分布図

3. システム構成

このシステムはWindows95の上で、Visual BASIC (V4) と Open-M (V7.0.3) により開発され、IBM Think-Padなどで使用される。Visual BASIC と Open-M とは、OCXにより接続されている。しかし、データのやりとりは開発のしやすさ、保守のしやすさ、大量受け渡しデータへの対応から、テキスト・ファイル渡しとした。

4. システムの特徴

①全面的にGUIを採用した。

②家族関係からも個人検索を可能とした。

③文字数字ベースのデータ表示を、一つのタブシートに統合した。

④定量的変化を、表形式で表示出来るようにした。

⑤随時、グラフ表示出来るようにした。グラフは、すべてリサイズ可能とした。また、タブシート及び表形式と同時表示することも可能とした。

⑥グラフ表示ではセット項目を用意した。また、指定された集団全体と個人のデータを同時表示できるようにした。

5. 結果

①GUI化により、操作の統一性と簡便性、保健指導の判り易さが向上した。

②家族関係から検索できるため、家族全体としての保健指導がやりやすくなった。

③文字数字データが一つのタブシートとなったため、表示切り替えが簡単となり、見やすくなった。

④表形式でのデータ表示により、通常の医療カルテで行われている検査結果票の罫張りのようにデータが表示されるため、変化と全体像が見やすくなった。

⑤グラフの表示、移動、リサイジングが自由にでき、表と同時表示も出来るため、保健指導の融通性が向上した。

⑥グラフ項目のセット化により、操作性が向上すると同時に指導がしやすくなった。また、集団の中での個人の状態が一目瞭然となったため、指導効果が高まることが期待される。

⑦VBを使うと画面の作成や変更が簡単である。また、Open Mはデータベースの作成や変更柔軟であり、検索スピードやコンパクトさも他に追従を許さない。OCXとテキストファイルによる相互通信は、汎用性、拡張性に優れており、処理速度も速かった。

6. おわりに

システムのGUI化は時代の流れであろう。また、医療や保健において、本人の理解を高めることや、本人との情報の共有が重要となってきている。携帯型端末による機動力（モバイルコンピューティング）や、表やグラフによる健康状態や変化の理解度の向上が、地域の保健活動の質的向上と、本人の健康へのライフスタイル変貌につながることを期待する。

Mテクノロジーも、常に時代の先端に位置し、広い世界との共存の時代になってきたことを喜ばしく感じている。

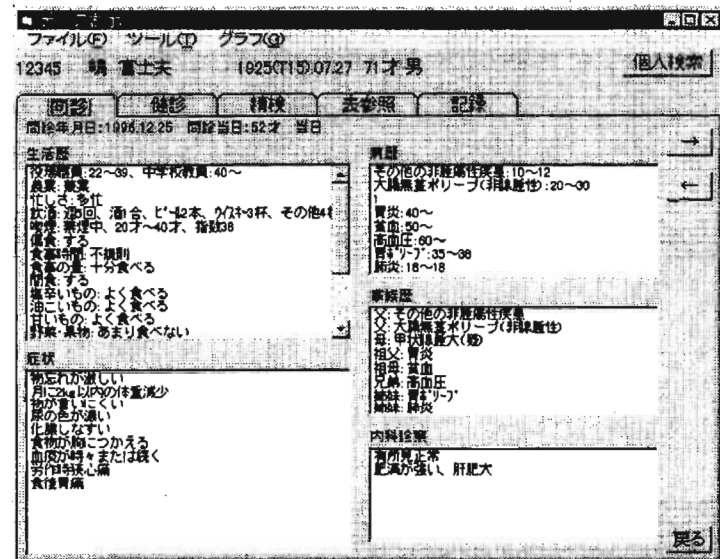


図1 問診結果表示画面

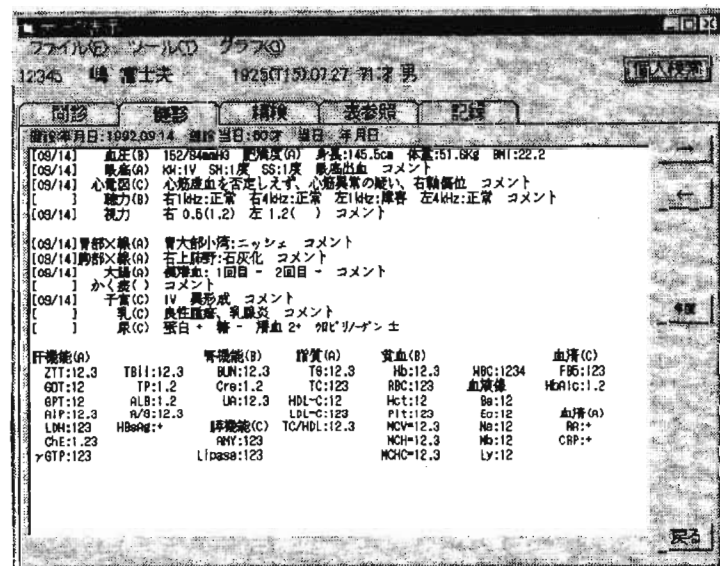


図2 健診結果表示画面

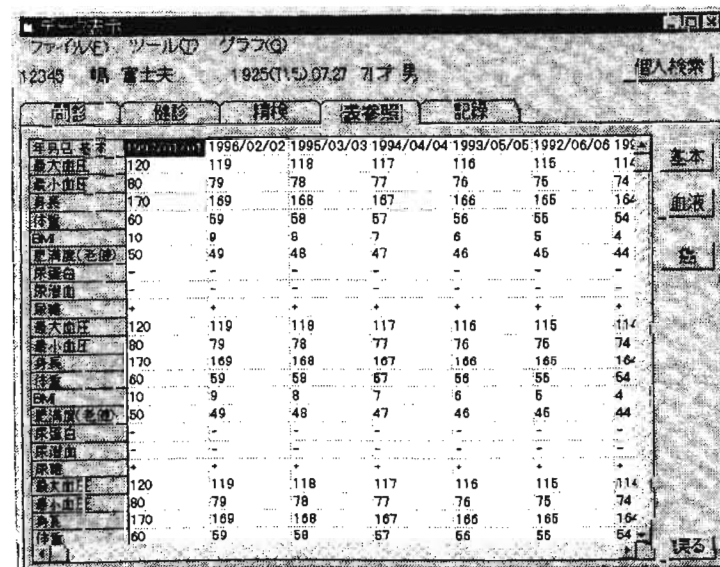


図3 表参照画面

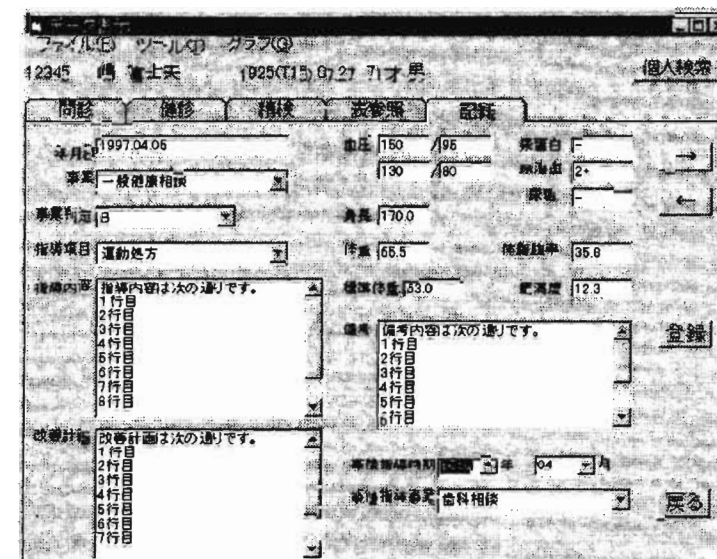


図4 記録登録・表示画面

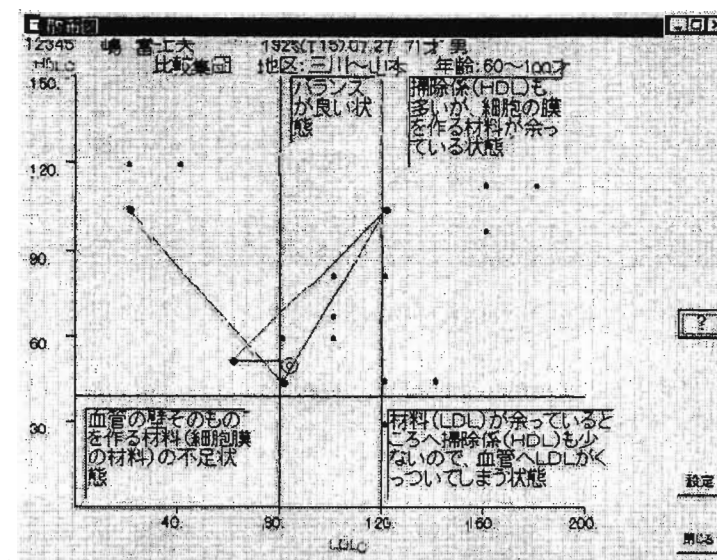


図5 折れ線グラフ画面

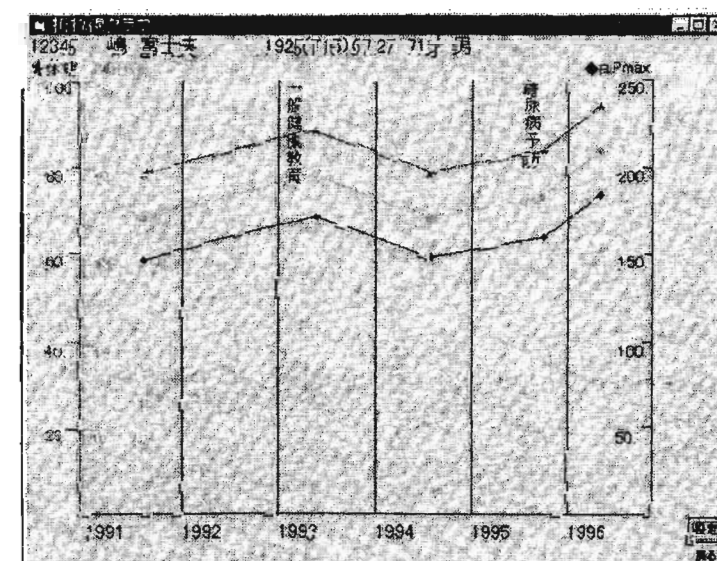


図6 散布図グラフ画面

MにおけるRPCの利用

Utilization of RPC in M

杉山 閑照*
Shizuteru Sugiyama

城崎 礼子*
Reiko Jouzaki

山下 芳範**
Yoshinori Yamashita

* 日本デジタルイクイップメント株式会社 西日本第1統合システム部システム3課
Digital Equipment Corporation Japan Network and Systems Integration Services
〒530 大阪市北区中之島2丁目2番2号ニチメンビル TEL(06)222-9211 FAX(06)222-9408

** 福井医科大学
Fukui Medical School
〒910-11 福井県吉田郡松岡町下合月23-3 TEL(0776)61-3111 FAX(0776)61-3114

キーワード: RPC, DCE, DSM, Windows NT, DDP

1 はじめに

リモートプロシージャコール(以降、RPCと略す)は、ネットワーク上の別CPUでのサブルーチン実行を、通常のサブルーチンコールと同様な呼び出し方法でプログラムできるメカニズムを提供する。RPCを利用すると、アプリケーションプログラマはネットワーク通信に関する特別なプログラミング無しに、クライアント/サーバ型アプリケーションの開発を進めることができる。このことにより、アプリケーション開発が進めやすくなり、開発コストを少なくすることが可能となる。また、RPCをうまく利用すると高性能のCPU間の通信処理を行うことができる。福井医科大学で、M環境の中でRPCとDDPを組み合わせたことにより効率のよいシステム間でのデータベース整合性を保つツールを構築したので報告する。

2 RPCを利用したツールの概要

福井医科大学では、DSM for Windows NTをクライアントアプリケーションに、DSM for OpenVMSをサーバアプリケーションにした電子カルテシステムを開発している。各クライアントシステムはサーバシステムとは独立に動くよう設計されている。したがって、サーバに蓄積されている患者情報の一部が、適時(予約患者など)、クライアントに送られ、そこで更新された後、サーバデータベースも更新される。同一患者情報がサーバ・クライアントで分散保持されるので、クライアントまたはサーバマシンや回線の障害を考慮したサーバ・(多)クライアント間のデータベースの整合性を保証するメカニズムが必要である。通常、システム間のMデータベースの整合性を保つためにポーリング方式のリモートグローバルアクセスDDPを用いるが、整合性を保つ必要のあるクライアントが200台と多いためその負荷が大きくなる。そこでM-RPCを利用し、DDP通信のネットワークトラフィックを減らした効率的なデータベース整合性保守ツールを作成した。

本ツールは、サーバ上にある患者電子カルテ関連のグローバル更新情報をもとに、その患者電子カルテ情報を同時保持しているすべてのクライアントに配信するためのツールである。また、このツールは、クライアントで更新された患者情報もできるだけすみやかにサーバ及び同時保持している(複数の)クライアントにも配信を行なう。各システムは独立して稼働・停止(正常または障害)するため、これらの情報の整合性・一貫性を常に保つのはできるだけ最小の時間遅れで保持するメカニズムの構築は結構難しい。また、その整合性保持のための処理に負荷をできるだけ少なくする設計・方式が必要とされる。以下、RPCとDDPを組み合わせた本ツールの概要を説明する。

各クライアント上には、

1. 事前に予約された患者の電子カルテデータ
2. その電子カルテデータがどのように変更されたかを記録する更新ログ

の2つのグローバルが置かれている。

サーバ上には、

1. 患者の電子カルテマスターデータベース
2. その電子カルテ情報をどのクライアントが読み出しているかを記録する配信患者テーブル
3. クライアントへの書き込みが、クライアントマシンの電源OFF等の理由で行われなかった場合に記録する配信ログ

の3つのグローバルが置かれている。

図1で示すように、本ツールは、次のような流れで処理が進む。

1. クライアントマシン(Windows NT)上の電子カルテアプリケーションは、ローカルに格納されている事前に予約された患者

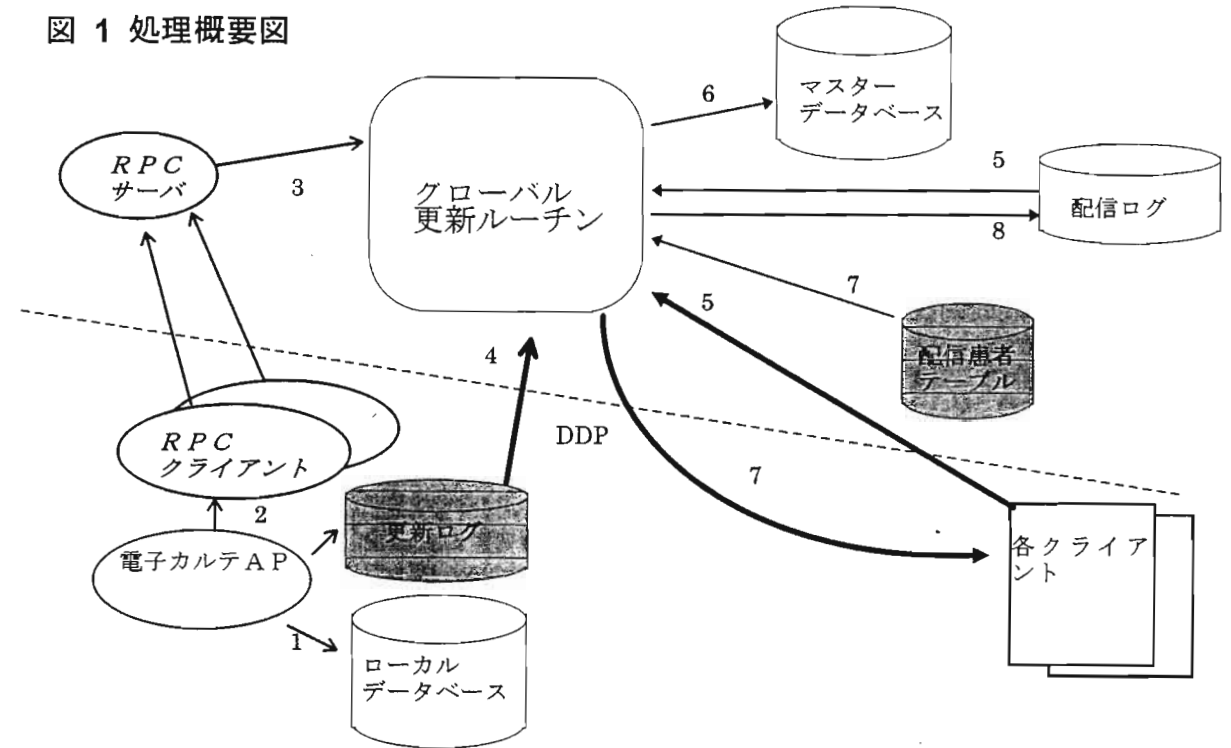
の電子カルテデータを元に動作し、その電子カルテデータに変更、追加が行なわれた場合、その内容を自システム内の更新ロググローバルに記録する。

2. クライアントマシン上の電子カルテアプリケーションは、適切なタイミングで、E CALLとして組み込まれているRPCクライアントを呼び出す。
3. RPCクライアントはサーバマシン(OpenVMS)上のRPCサーバを呼び出し、RPCサーバは、DSMのコーラブルインターフェースを介して、DSMのグローバル更新ルーチンを呼び出す。
4. サーバ上のグローバル更新ルーチンでは、まず、DDP経由で、クライアントの電子カルテアプリケーションが作成した更新ロググローバル(サーバに転送すべきデータ)を読み出す。
5. 次に配信ロググローバルをもとに、未配信のデータのあるクライアント(不整合を持つクライアント)から、DDP経由で更新ログデータを集める。

6. 4.と5.で集めた更新ログデータをタイムスタンプで比較し、最新の電子カルテマスターデータとして、サーバ上のマスターデータベースを更新する。
7. 配信患者テーブルグローバルをもとに、DDP経由で各クライアントに6.で作成した最新の電子カルテマスターデータを書き込む。
8. 何らかの理由でクライアントに書き込めなかったものについては、配信ロググローバルに記録しておく。

このようにして、クライアントは、RPCの呼び出しが正常に行なわれたことのみを確認した後、次の処理に移ることが可能になり、ネットワーク障害やサーバマシン障害のクライアントアプリケーションへの影響を極小化することができる。またサーバマシンを複数準備することによって、処理の集中しているサーバ処理を別のマシンに処理分散することも容易に行なえるようになり、システムの可用性が増した。

図1 処理概要図



3 RPCのメリット

本ツールの通信手段として、RPCを採用したメリットは、以下の通りである。

1. RPCを利用することによって、サーバアプリケーションがクライアントアプリケーションのグローバル更新のタイミングを認識できるようになり、そのタイミングでのみ、DDP通信を行なうようにすることによって、不必要なDDP通信のトラフィックを発生させることなく、グローバル更新を行なうことが可能となった。すなわち、サーバが多数のクライアントのグローバル更新状況を調べるようなポーリングの多いプログラムを行なう必要がなくなった。
2. 既存の通信手順を利用することによって、新たな手順の設計コストを無くし、プログラム開発面でも、クライアントのグローバル更新プログラムにE CALLを呼び出すようにするだけで、複雑なグローバル更新プログラムをサーバ上に一元化でき、クライアントプ

- ログラムを単純化することができた。
3. 実績の多い公開技術の利用により、異機種間でもデータのやり取りの複雑なテストなしにスムーズに実現することができた。

4 RPC利用時の注意点

RPCには、DCE (Distributed Computing Environment: OSFの提唱する分散コンピュータ環境) RPCとSunRPCの2種類があり、それぞれ相互には互換性がない。すなわち、DCE RPCを利用して作成されたクライアントアプリケーションとSunRPCを利用して作成されたサーバアプリケーションでは、通信が成立しない。どちらかのRPC規格に統一する必要がある。

福井医科大学では、Windows NTがDCE RPCを標準装備していることから、DCE RPCに統一し、サーバ上にDCE RPCを実現する基本ソフトウェアを設置して開発した。

5 考察

現状では、M言語にRPCの機能が装備されて

いないために、ECALLによりRPCを実現したが、ローカルプログラム呼び出しとまったく同じ手順で、リモートプログラムを動作させるというRPCの機構は、特別なネットワークの知識を必要としないことから、アプリケーションプログラムには理解しやすく、共同利用されるプログラムをサーバ上で一元管理したり、ネットワーク上のシステムリソースを最適な状態で利用する設定変更を容易にしたりすることで、システム管理者にとってみても、受け入れやすいものと思われる。

今後、M言語にも同様のRPCの機構が取り込まれることに期待したい。

参考文献

「DCEプログラミングガイド」(John Shirley, Wei Hu, David Magid 共著、中田温朗訳)
発行元：インターナショナル・トムソン・パブリッシング・ジャパン
発売元：株式会社 オーム社

図2 サンプルプログラム (クライアント側)

```
EMRURPC1(GNAME);
  N HOST,UCI,VOL,SSTATUS,TIMEOUT,PARAM
  S TIMEOUT=+$G(^EMRURTMO)
  S:TIMEOUT=0 TIMEOUT=8000 ; default
  S PARAM=+$G(^EMRURPRM)
  S (STATUS,SSTATUS)="
  ; ホスト名を得る
  S HOST=^EMRURHST
  ; UCI,VOL をセットする
  S UCI=$P($&ZLIB.%UCI,",1),VOL=$P($&ZLIB.%UCI,",4)
  ; 更新ルーチンを呼ぶ
  S STATUS=$&DCRPC.UpdateMasterDB(HOST,UCI,VOL,GNAME,TIMEOUT,PARAM, SSTATUS)
  S STATUS=STATUS_"_SSTATUS
  q STATUS
```

図3 サンプルプログラム (サーバ側)

```
EMRURPC2 ; ルーチンのエントリ
;
W $&ZLIB.%CDATASC($P($H,",1)," ",$&ZLIB.%CTIMASC($P($H,",2),2)," CLIENTからの要求",!
ZW UPDUCI,UPDVOL,UPDGBL
S STATUS=""
S S=$$UPDATE()
Q
UPDATE() ; update the master database
N (UPDUCI,UPDVOL,UPDGBL,STATUS)
;
; このプログラムはクライアントからのRPCで起動され、クライアントから取得した更新ログを使用して
; マスターデータベースを更新し、影響のあるクライアントへ配信するプログラムです。
; 入力:
;   UPDUCI   RPCを発行したクライアントDBのUCI
;   UPDVOL   RPCを発行したクライアントDBのボリュームセット名
;   UPDGBL   更新ログのグローバル名
; 戻り値:
```

```
;   0       失敗
;   1       成功
S STATUS=""
K ^EMRURULG($J)
; 0. 必要な情報のチェック
I UPDUCI="" D Q 0
.W "内部エラー: 変数'UPDUCI'が NULL です。"!
.S STATUS="内部エラー: 変数'UPDUCI'が NULL です。"
I UPDVOL="" D Q 0
.W "内部エラー: 変数'UPDVOL'が NULL です。"!
.S STATUS="内部エラー: 変数'UPDVOL'が NULL です。"
I UPDGBL="" D Q 0
.W "内部エラー: 変数'UPDGBL'が NULL です。"!
.S STATUS="内部エラー: 変数'UPDGBL'が NULL です。"
; タイムスタンプ (日付) のセット
S DATE=$P($H,",")
; 1. 呼び出したクライアントから更新ログを貰う
W "***更新ログの読み込み**"!
D GTUPDLG(UPDUCI,UPDVOL,UPDGBL,"",DATE,,STATUS)
Q:STATUS="" 0
; 2.1.更新ログで使用されている患者 No.と同じ患者 No.を持っているクライアントから更新ログを取り出す。(更新ログのロックが必要?)
W "***他のクライアントからの更新ログの読み込み**"!
S kn="" F S kn=$O(^EMRURULG($J,"UPDTIM",kn)) Q:kn="" D
.S vol=""
.F S vol=$O(^EMRUADWN(DATE,kn,vol)) Q:vol="" D
.. Q:vol=UPDVOL ; 発信したノードならば既に取り込んでいるので quit
.. S uci=$P(^vol,",")
.. D GTUPDLG(uci,vol,UPDGBL,kn,DATE)
S STATUS=""
; 3. 呼び出したクライアントの配信ログと 1、3 で求めた更新ログを時系列に
; 並べ、差分を求め、マスターデータベースの更新を行う。
; 配信ログの読み込み
W "***配信ログの読み込み**"!
S kn="" F S kn=$O(^EMRURULG($J,"UPDTIM",kn)) Q:kn="" D
.S vol="" F S vol=$O(^EMRURULG($J,"SCSNODE",vol)) Q:vol="" D
.. M:^vol)=0 ^EMRURULG($J,"UPDTIM",kn)=^EMRURDLG("FAIL",kn,vol)
; マスターデータベースの更新
W "***マスター更新**"!
S kn=""
F S kn=$O(^EMRURULG($J,"UPDTIM",kn)) Q:kn="" D
.S updtim=""
.F S updtim=$O(^EMRURULG($J,"UPDTIM",kn,updtim)) Q:updtim="" D
.. S uniq=""
.. F S uniq=$O(^EMRURULG($J,"UPDTIM",kn,updtim,uniq)) Q:uniq="" D
... D UPDDB; マスターの更新
Q:STATUS="" 0
; 4. クライアントの更新ログを削除する。
S uniq=""
F S uniq=$O(^EMRURULG($J,"UPDLOG",uniq)) Q:uniq="" D
.. uniq = <vol><uci><$p($h)><$p($h,2)> たとえば FKIU5323400123
.S uci=$E(uniq,4,6)
.S vol=$E(uniq,1,3)
.D KLUPDLG(uci,vol,UPDGBL,uniq)
Q:STATUS="" 0
;
(以下省略)
```

Mストアプロシジャを用いた自動再来受付機の開発

○鈴木隆弘¹⁾、山崎俊司¹⁾、高林克日己²⁾、中野正樹¹⁾、劉亜斌¹⁾、本多正幸¹⁾、
里村洋一¹⁾

Takahiro Suzuki, Shunji Yamazaki, Katsuhiko Takabayashi, Masaki Nakano,
Ahin Ryu, Masayuki Honda, Youichi Satomura.

¹⁾千葉大学医学部附属病院医療情報部、²⁾千葉大学医学部第二内科

Keywords: Mストアプロシジャ、M言語、Visual Basic

1. はじめに

グラフィカルユーザーインターフェイス (GUI) は最近のシステムにおいて必須のものとなりつつあり、M言語においてもGUIを構築するための拡張が各社より提案されている。我々は住友電工の開発したMストアプロシジャを用いていくつかのシステム開発を行っており、今回は最初のプロダクトとして稼働を開始した自動再来受付機を報告する。

2. 当院における外来待時間の状況

当院においても患者の待ち時間の長さは大きな問題となっている。アンケート結果によれば、来院は早い時間に集中する傾向があり、受付は開始直後が最も混雑する。病院への到着から受付までの待ち時間は平均で約45分、受付から診療までの待ち時間は約85分となっている。

自動受付機導入の狙いは第一に受付待ち行列の解消で、そのために処理の分

散を図り、位置的な分散として全部で10台を各科外来前に配置し、時間的な分散として早朝からの自動稼働を行った。

第二に早朝に来院する患者への対応として、手続きの簡易化、整理券の廃止を行い、患者が診療科前の待合室で待てる様にした。

3. 自動再来受付機の特徴

1) 職員による介助が不要

2) わかりやすい表示

大きな文字を用いる

絵・写真を併用する

音声による案内を行う

3) 簡単な操作

患者カードを差し込むだけ

4. システム構成

端末として富士通の FMV パソコンを用い、マイクロソフト社の WindowsNT を OS として、その上でマイクロソフト社の Visual

Basic で開発したクライアントプログラムを動かしている。このパソコンにはカードリーダーが接続され、専用のボックスに収納されている。サーバーにはサン社の UNIX ワークステーションを用い、住友電工の U-MUMPS をデータベースとして利用している。更に、サーバーからは医事会計用のメインフレームへ会計情報が転送される。これらのマシン間はイーサネット接続されている。

5. Mストアプロシジャによるプログラミング

Mストアプロシジャとはマイクロソフト社の Visual Basic によって作成されるアプリケーションからM言語によるデータベースへ処理を依頼し、結果を受け取る一種のクエリーである。これによってクライアントサーバーシステムを構築できる。

リスト1に Visual Basic のコードの一部を示す。変数の受け渡しには配列を利用している。リスト2にはM言語側のルーチンを示す。

処理速度に関しては満足できる結果が得られており、一人の処理は音声による案内の間に終了して患者を待たせることはない。サーバでは1分間に27人を処理した実績があり、それ以上になっても対応可能と考えられる。

ユーザーインターフェイスを作成する言

語として Visual Basic を利用することの利点は、マルチメディア機能を容易に実装できることや豊富な VBX 拡張コントロールを利用できることである。これによって開発にかかる時間と労力を節減することができる。今回の開発でもカードリーダーとの接続には市販の通信コントロール VBX を利用している。

しかしながら、Mストアプロシジャは現在のところの Visual Basic の Ver. 2.0 にしか対応していない。既にクライアントの OS が WindowsNT や Windows95 などの 32 ビット OS へ移行し、開発ツールも 32 ビット化・OCX 化が急速に進んで VBX のサポー Visual Basic がうち切れられ始めている現状では 16 ビットの Visual Basic 2.0 は時代に取り残された存在であり、32 ビット化への早急な対応が望まれる。

6. 導入後の状況と評価

自動受付機導入後に行ったアンケート調査では、表示の見易さに対しては、78%が見易いと答えた。操作の分かり易さに対しては 84%が分かり易いと返答し、操作を間違えた経験があると答えたのは 7%にとどまった。

待ち時間においても、到着から受付までの待ち時間が平均で約 15 分と 30 分短縮された。受付後の待ち時間は変化していないものの、これにより 58%が待ち

時間が短くなったと感じており、56%が待ち時間が楽になったと答えた。楽になった理由として最も多かったのは並ばなくても良くなったことで、処理の分散を図った成果が示された。

7. まとめ

・Mストアプロシジャにより Windows の

GUI・マルチメディア機能と既存のデータベースが連携した再来受付アプリケーションを短時間で開発できた。

・自動再来受付機は、概ね好評をもって受け入れられている

・自動再来受付機によって受付前の待ち時間は短縮され、患者の負担は軽減された

リスト1. Visual Basic のコード

Sub UKETUKE ()

Dim ANS, SendPnum, RecvPnum As Integer

'変数の宣言

SendPnum = 2

RecvPnum = 16

ReDim SENDPARAM(SendPnum) As String

'配列の宣言

ReDim RECVPARAM(RecvPnum) As String

SENDPARAM(0) = ID

'Mへ送る変数をセット

SENDPARAM(1) = CmdPara

ANS = M_SPDo(UCIName, VOLName, "A1^UAZOPRC", SENDPARAM(), SendPnum, RECVPARAM(), 10)

'Mストアプロシジャを起動

If ANS < 1 Then

'エラーが起きた場合

FmMain!LabelMes.Caption = "コンピュータと通信できません。玄関ホールの自動受付機をお使ください"

Elseif ANS = 1 Then

'返信が1個のみの場合

MDATA = RECVPARAM(0)

If MDATA = "YG" Then

Call YG '予約外の場合

Elseif MDATA = "TZ" Then

Call TZ '登録済みの場合

Elseif MDATA = "E1" Then

Call E1 '予約が10時以降の場合

Elseif MDATA = "E2" Then

Call E2 '予約より2時間以上早い場合

Else

End If

Else

'複数個の返信があった場合

If RECVPARAM(0) = "OK" Then

'登録成功の Flag があった場合

For SEQ = 1 To ANS - 1

SDATA = RECVPARAM(SEQ)

以下省略

リスト2. M言語のコード

A1 ; VBからデータを受け取る

S ID=%SPARAM(0)

S PCNO=%SPARAM(1)

;

A2 ; 各種パラメータのセット

S \$ZE="", \$ZT="ERROR" _\$T(+0)

S %OP="9999" D ^ZEPARADEM

D ^ZUD1 S TODAY=ZUZD3

S TIME=\$P(\$H,"",2)

中略

; VB用データをセット

S VDATA(SEQ)=PNAME_"_"_KAX_"_"_YTIMX_"_"_GRPX_"_"_KA_"_"_GRP_"_"_YTIM

; VBへデータを返す

S ANS=\$\$WRITE^%MSPSRV("OK")

以下省略

M グローバルをアクセスする Java ライブラリの試作

A Prototype of Java Library To Access M Globals

堀田 稔

Minoru Horita

日本デジタル・イクイップメント (株)
西日本第一統合システム部
Digital Equipment Corporation Japan

1 はじめに

近年の Internet ブームに乗って Java が注目を集めている。Java で開発される Web ブラウザ上のプログラム (アプレット) により、HTML/CGI のプログラミング上の制限を越えた、より高度な Web アプリケーションが構築可能になってきている。

さらに Java は、単なるアプレット開発ツールにとどまらず、マルチスレッド、ネットワーク・アクセス機能などをそなえた汎用オブジェクト指向言語であり、この点においても、プログラマの支持を得ている。

オブジェクト指向技術は、データとそれに対する操作のカプセル化や、継承などにより、ソフトウェアの保守性、柔軟性、再利用性を高めるものである。近年、M の世界でもオブジェクト指向技術を、さまざまな形で取り込む試みがなされている。

オブジェクト指向技術と、柔軟かつ高速なデータベースをもつ M とを組み合わせることは、高機能のアプリケーションを効率良く開発することを可能にし、M を核とするアプリケーションの幅を広げることに役立つと期待できる。

今回、Java から M のグローバルをアクセスするライブラリの設計・試作を行なったので報告する。

2 Java

Java は Sun Microsystems 社で開発されたプログラミング言語である。Java は、Web サーバからダウンロードして Web ブラウザ上で実行されるプログラムであるアプレット開発用の言語として注目されている。しかし、Java は単なるアプレット

開発用言語でなく、汎用プログラミング言語としても強力な機能を備えている。

Java の優れた点として次のようなことが挙げられる。

オブジェクト指向 Java はオブジェクト指向言語である。カプセル化、継承、多態性などをサポートし、柔軟で、堅牢なソフトウェアの開発を可能にする。

マルチスレッド Java はスレッドを言語レベルでサポートする。スレッド API はシンプルなものであり、マルチスレッドのアプリケーションを容易に開発することができる。

可搬性 Java はインタプリタ型言語である。ソースコードは、仮想機械 (Java Virtual Machine: JVM) 用のコードに翻訳され、JVM 上で実行される。したがって、JVM が実装された OS であれば「バイナリレベル」で可搬性がある。現在、Win32、Solaris、Digital Unix をはじめ、いくつかの OS で JVM が実装されており、Java によりマルチプラットフォーム対応のアプリケーションが容易に作成できる。

豊富なネットワーク関連ライブラリ Java の標準ライブラリには、豊富なネットワークライブラリが付属しており、ネットワークアプリケーションの開発をサポートする。特に、HTTP プロトコルが API レベルでアクセスでき、より複雑な WWW アプリケーションの作成を可能にする。

国際化アプリケーションのサポート Java の文字は 16 ビットの UNICODE で表されており、国際化アプリケーションをサポートする

基盤となっている。また、各言語に固有の情報ロケールという形でカプセル化され、特に日本語などの主要な言語のロケールは標準でサポートされているため、英語/日本語をともにサポートするアプリケーションなどの開発が容易に行なえる。

数々のベンダーがサポートを表明 数々の主要なベンダーが Java のサポートを表明している。このことは、今後 Java 関連のツール、パッケージなどが、次々に開発される可能性を示唆する。

Java には以上のような長所があるが、まだ開発されてから日が浅いことなどによるデメリットもある。次に短所を列挙する。

- まだ発展途上で変化が激しいため、API などが変わってしまい、アプリケーションの保守性に影響する可能性がある。
- ネイティブコードに比べプログラムの実行速度が遅い。
- 開発ツールが不足している。特に Visual Basic のようなビジュアルツールが今後開発されることが望まれる。
- グラフィック部品 (AWT) が未成熟である。Java では、グラフィックシステムに依存しない形で AWT を実装しているが、逆にプラットフォーム非依存性が障壁となって、各グラフィックシステムの洗練された機能が使えず、見栄えも貧弱である。グラフィック機能は、Java で最も変化している部分の一つである。

以上 Java についての概要を述べたが、昨今の業界動向と Java の持つ魅力とを考えれば、Java が今後重要な技術として発展していく可能性が高いであろう。

3 ライブラリの設計

3.1 概要

試作したライブラリは、Java のプログラムから、ネットワーク上に存在する M のデータベースにアクセスするためのものである。例えば、このライブラリを使用すれば、Java アプレットから M データベースが利用できる。

今回試作したのは、

- M サーバと Java プログラムが通信するためのプロトコル
- リクエストを受け付ける M サーバプログラム
- M サーバにリクエストを行なう Java API

である。次に各要素の説明を行なう。

3.2 プロトコル

今回実装したプロトコルは、典型的なクライアント/サーバ型のもので、クライアント (Java プログラム) から要求を出し、それをサーバ (M プログラム) で解釈・実行した結果を、返答メッセージとしてクライアントに返すものである。メッセージのやりとりは TCP プロトコル上で行なう。

例えば、あるグローバル変数の値を取得するためのリクエストのフォーマットは次の通りである。

```
"GBLGET" | [SP] | gname | [SP]
```

```
nsubs | len_sub1 | sub1
```

```
...  
len_subn | subn
```

gname: グローバル名 (文字列)
[SP]: 空白文字
nsubs: サブスクリプトの数 (2 バイトバイナリ)
len_subn: サブスクリプト n の長さ (2 バイトバイナリ)
subn: サブスクリプト n の値 (文字列)

これに対して、M サーバは次のような返答を返す。

```
code | vlen | value
```

code: エラーコード (文字列)
vlen: 値の長さ (2 バイトバイナリ)
value: 値 (文字列)

試作では、一つのリクエストに一つの返答という単純なモデルに基づいているが、実用的なアプリケーションでは、大量のデータをいくつかに分けてやりとりするなど、より実効/効率を考慮した設計を行なう必要がある。

3.3 Mサーバ

前節で述べた形式のリクエストを受け付け、得られた実行結果を返答メッセージの形で生成し、それをクライアントに送信するのがサーバの役割である。

一般に要求の実行自体は難しい処理ではない。しかし、文字列の形で得られたグローバル参照を、コマンドのパラメータとして指定する場合に、間接参照を利用するなど、M言語の利点を生かした実装が可能である。

3.4 Java API

Javaのプログラムでは、次のようなクラスを介してMサーバにアクセスする。

MEnv Mサーバが提供するMの環境を表す。Mサーバの実行されているIPアドレスと、ポート番号によりM環境を指定する。

このクラスは、次に述べるようなグローバル変数を表すオブジェクトを生成するメソッドの他、

- Mルーチンを実行する execute メソッド
- Mの式の値を返す evaluate メソッド

などを持つ。例えば、

```
menv.execute("D ^CHKNAME");
```

でCHKNAMEルーチンが実行され、

```
v = menv.evaluate("$H");
```

で、\$Hの値がvに代入される。

MGlobal Mのグローバル変数を表す。例えば、MEnvクラスの変数menv、MGlobalクラスの変数gにたいして、

```
g = menv.getGlobal("DOCTOR");
```

とすることで、gはmenvに接続されたサーバの^DOCTORを表すようになる。

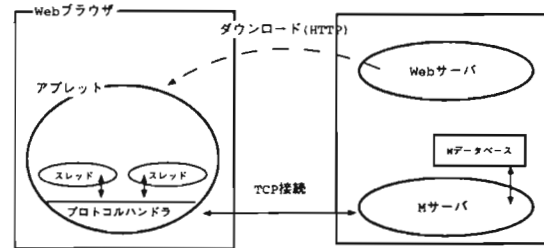
MGlobalTraverser グローバル変数のすべてのサブノードのようなグローバル変数の集合を、順に処理する際に利用する。

例えば次のコードは、^DOCTORのサブノードを順に処理するものである。ちょうど、M関数の\$ORDERを使った処理に対応する。

```
MGlobal g;  
g = menv.getGlobal("DOCTOR");  
MGlobalTraverser t = g.getSubNodes();  
while ( t.hasMoreElements() ) {  
    // tを使った処理  
    t.next();  
}
```

4 アプレットとMデータベース

今回試作したライブラリによって実現できるアプリケーションの中でも、代表的なものは、MデータベースにアクセスするJavaアプレットである。JavaアプレットがMデータベースにアクセスする際のプロセスを次に示す。



Webサーバからダウンロードされたアプレットは、ブラウザに実装されたJVMで実行される。そして、Mサーバに対するTCPコネクションを確立し、そのコネクションを通して、Mサーバと通信を行なう。

JavaアプレットによるWebアプリケーションは、

- 可搬性がある。すなわちアプレットは、Webブラウザが動く環境であればOSやハードウェアに関係なく動作する。
- アプレットはマルチスレッドで動作するので、データベースの内容を監視しながら更新するなど、リアルタイム性を持ったWebアプリケーションを作成できる。
- HTTPとは別のTCPコネクションを利用できるため、サーバとの通信を高い自由度で行なえる。

など、Webを基盤とし、かつ、HTML/CGIの制限に縛られないという、大きな利点を持っている。

例えば、米国・ボストン近郊の病院や大学が共同で行なっているThe W3 EMRS (World-Wide Web Electronic Medical Record System) Project

¹では、JavaアプレットによるWeb上で動作するモニタリング・システムのデモを行なっている。このシステムは、

- ICU(intensive care units)からのデータをリアルタイムに取得
- データをグラフなどのグラフィカルな形で表示
- Web上で動作するため、さまざまなプラットフォームで利用できる。

というもので、Javaアプレットが持つ機能を知ることのできる良い例である[1]。

このようなJavaアプレットがMデータベースを利用できれば、リアルタイムデータと静的なデータとの統合がはかれ、それらをWebブラウザ上でグラフィカルな形で操作するという、新しい形のアプリケーションの構築が可能となる。このことは、今回試作したJavaライブラリが目指すところの一つでもある。

5 課題

- ネットワーク通信によるパフォーマンス上のオーバーヘッドを最小化するような、プロトコルの設計が必要である。
- より抽象度の高いクラスライブラリとして発展させ、高い生産性を持つ開発ツールへとつなげていく。

6 まとめ

Javaはオブジェクト指向プログラミング言語であり、アプレットの開発や、ネットワークライブラリの充実など、Internet/Intranetをベースとするシステムの開発にも、威力を発揮する。Javaの強力な言語機能と、Mの高速で柔軟なデータベースとの組合せで、高機能のアプリケーションを比較的容易に開発できると期待できる。

今回試作した、Javaライブラリにより、JavaアプレットがMデータベースを利用することを可能にし、Mを核とするシステムをより新しい形のアプリケーションに統合していくことを目指した。また、MデータベースをJavaの提供するオブジェクト指向機能の中に取り込むことにより、Mとオブジェクト指向技術の融合を試みた。

¹ <http://www.emrs.org/>

今後は、実システムに耐え得る高パフォーマンスの確立や、より高レベルのツールの作成など、より良いシステムを高い生産性で開発するための基盤として、このライブラリを発展させていくつもりである。

参考文献

- [1] K. Wang, I. Kohane, K.L. Bradshaw, J. Fackler *A Real Time Patient Monitoring System on the World Wide Web*, http://www.emrs.org/publications/amia_icu.html
- [2] D. Flanagan 著, 日本サン・マイクロシステムズ株式会社 監訳, 永松健司 訳, *JAVA クイックリファレンス*, オライリー・ジャパン, 1996

[要旨]

販売管理システムを新規にGUI（グラフィカル・ユーザー・インターフェース）で構築するにあたり、従来のM言語アプリケーションとは異なる方法で開発を試みた。

すなわち、OpenM/SQL+ODBC+PowerBuilderというプロダクトの組み合わせでの開発である。

この方法を採用した主な目的は下記の通りである。

- ・GUI（グラフィカル・ユーザー・インターフェース）アプリケーション構築のツールとしてPowerBuilderとOpenM/SQLを組み合わせた場合の有用性の確認（VisualBasicとの比較を含む）
- ・OpenM/SQL+ODBCでの生産性/メンテナンス性の確認
- ・OpenM/SQL+ODBCでの処理実行速度の確認

今回の販売管理システム開発の結果、OpenM/SQL+ODBC+PowerBuilderの組み合わせは、非常に高い生産性/メンテナンス性が得られることが確認できた。

[現状]

従来の販売管理システム開発での問題点

従来の販売管理システムは、VisualBasic（マイクロソフト社製）+OpenM（InterSystems社製）の組み合わせで開発を行った。

VisualBasicでGUI部分を構築し、データベースとのデータ連携は専用のカスタムコントロール（VisualM.OCX：InterSystems社提供）を経由して実施する方式を採用した。

この方式では、データベース（グローバル変数）から任意のデータを抽出するためには、VisualBasic側とM側双方での処理の作り込みが必要である。（表1参照）

(表1)

VisualBasic側	M側
<pre> mvbl.p0=itemcd.text mvbl.code="set VALUE=\$\$getitemnm(P0)" mvbl.execflag=1 </pre>	<pre> getitemnm(itemcd) set iteminf=\$g(^item(itemcd)) set itemnm=\$p(iteminf,"^",2) quit itemnm </pre>
<pre> itemnm.caption=mvbl.value </pre>	

*mvbl.xxxは、VisualM.OCXのプロパティ項目

実際の業務では、大福帳型データベースシステムの形態を採用しているために大量のデータが抽出対象となる。抽出データが大量になると、これを1回のM呼び出しで受渡そうとするとデータエリアのデータオーバーフローが発生してしまう。これを避けるためデータを複数に分割し、全データをVisualBasic側に転送し終わるまでM呼び出しを繰り返す制御が必要になる。

また、上記のような制御を組み入れたとしても、非常に大量のデータを転送するとVisualBasic自体でメモリーオーバーフローが発生してしまうので、別の対策も講じなければならない。そのため、一層複雑なコーディングが必要である。

以上に述べた事例以外にも、VisualBasicとOpenMとの組み合わせにおいては、考慮すべき点を確認されている。

[新しい開発方法]

新販売管理システムは次の組み合わせで構築した。

OpenM/SQL+ODBC+PowerBuilder

- ・OpenM/SQLは、OpenMのグローバル変数に対して、データ定義（テーブル定義）を行い、SQLでアクセスするためのOpenMのサブセット。
- ・ODBCは、InterSystems社の提供するドライバーを利用。ODBC2.0規格。
- ・PowerBuilder（パワーソフト社製）は、Ver 5（Professional版）を使用

*実行環境

サーバー : WindowsNT4.0J, CPU=Pentium 60MHz, MEM=64MB

クライアント : Windows95, CPU=Pentium166MHz, MEM=64MB

OpenM上では、M/SQLを用いてデータ定義のみを行う。PowerBuilderはSQL文を自動生成するので、これをODBCを通じてOpenMに渡す。OpenMはSQL文をMプログラムに変換し、先のデータ定義に従ってグローバルをアクセスする。結果

的に、Mのコーディングはほとんど不要になる。

[結果]

下記に、OpenM+VisualBasicの組合わせ、OpenM/SQL+ODBC+PowerBuilderの組合わせにおける開発工数、及びレスポンスタイムを計測した結果を記す。

(表2) プログラム開発工数 (分析工数は除く)

	OpenM+VisualBasic (OCX)	OpenM/SQL+ODBC+PowerBuilder
工数	約3ヶ月	約2ヶ月

(表3) レスポンスタイム

	OpenM+VisualBasic (OCX)	OpenM/SQL+ODBC+PowerBuilder
部品名検索	約0.5秒	初回 : 約1.0秒 2回目以降 : 約0.5秒
受注データ 検索	約2.0秒	初回 : 約5.0秒 2回目以降 : 約4.0秒
出荷データ 検索	約2.0秒	初回 : 約5.0秒 2回目以降 : 約4.0秒

[考察]

今回の組合わせ (OpenM/SQL+ODBC+PowerBuilder) で確認した内容を下記に記す。

(1) OpenM/SQL を利用するメリット

OpenM/SQL を利用することにより、階層構造のMグローバルデータをRDBとして扱うことが可能になる。このことに多くの意味を含んでいると考えられる。

a. 開発に必要なMプログラマーが少数で済む。

通常のMプログラムを作成する場合、M言語の文法や各種関数の利用方法を熟知しなければならない。しかし、M言語にSQL文を埋め込む(Embedded SQL)方式にする場合、実際にデータアクセスを行う処理は非常にシンプルなものとなり、特にM言語を熟知していなくても、SQLの知識があれば処理プログラムを作成することが可能となる。

また、実際に使用するSQL文が正しいものかどうかは、ISQL(Interactive SQL Queries)を使用することによって可能であるために、生産性の向上、及び処理プログラムの品質の向上を見込むことができる。

b. インデックスデータ (例えば、部品マスタの部品名称索引) の作成はOpenM/SQLのストアード・プロシジャで処理することができる。

上記bの例のように、特定項目に対する検索が発生することが見込まれる場合、検索対象となる項目で索引用グローバル変数を作成することが考えられる。通常、このような索引用グローバル変数の作成も、業務処理プログラム中に組み込む必要があるが、OpenM/SQLに対してデータ更新(Update)命令が発行されたら索引用グローバルデータを作成する、といった処理をトリガーとしたストアード・プロシジャとして定義することができる。

従って、何らかの理由で作成すべき索引用グローバルの内容が変更になった場合でも、業務処理プログラムには一切影響を与えない。この事項は、システム構成要素(モジュール)の独立性の向上に寄与し、システム全体の品質向上を図ることが可能となる。

c. RDBのチューニングが不要

データベースシステムを構築する場合、処理速度を上げるために物理構造を変更する。しかし、一般のRDBでは、データベースの物理構造と論理構造が密接な関係を持っており、物理構造定義の変更は論理構造定義の変更が必要で、これはデータベースの非正規化を伴う。非正規化されたデータベースの整合性を保持するために、処理プログラムが複雑になるという問題も生じる。

OpenM/SQLの場合、Mグローバル変数に対して論理的にテーブル定義を設定するので、論理構造と物理構造の独立性が高い。従って、処理速度の改善のために物理構造(すなわちグローバル変数の構造)を変更しても論理構造の変更は必須ではない。

(2) ODBC+PowerBuilderによる高い生産性

上記(1)の内容は、M言語プログラムの中で、OpenM/SQLに対してSQLを発行する処理についての考察であったが、次に、M言語を使わずにシステム構築を行う方法についての考察を下記に記す。

OpenM/SQLは、ODBCドライバーを経由することによって、サードパーティの提供するシステム開発ツールと連携することができる。以下は、PowerBuilderをシステム開発ツールとして選択した場合の考察である。

PowerBuilderの特徴は、GUI設計とデータベースアクセス設計を明確に分離した形で開発を進められるツールとなっていることである。

「GUI設計」とは、画面上の各項目の配置やその各種属性(文字の大きさ、

使用するフォントの種類、等) やカーソルの動作を定義することと、複数画面間の遷移を制御する部分を設計することである。

このGUI設計においては、PowerBuilderのスク립ト(専用のプログラミング言語)を利用することにより、VisualBasicでのプログラミングと比較して遜色ないレベル、あるいはそれ以上の詳細な制御が可能となる。また、PowerBuilder スクリプトのプログラミング・スタイル、及び命令のシンタックスにおいても VisualBasic と互換性を持っているために、スク립ト習得にかかる時間を短縮することができる。

「データアクセス設計」とは、データを抽出するテーブルの指定、テーブル内の項目の指定、また、データ抽出条件を制御する部分を設計することである。

データアクセスについては、ODBC 経由となることから SQL の文法を習得する必要がある。しかし、PowerBuilder の提供するデータウィンドウ作成機能を利用すると、画面上で必要項目をクリックするだけで発行すべき SQL 文が自動生成されるので、SQL に対する知識がない開発要員(M言語プログラマー、等)でも十分開発を進めることができる。また、自動生成された SQL を変更することも可能であるので、SQL レベルでのパフォーマンス・チューニングも可能である。

これら、「GUI設計」と「データアクセス設計」は相互に依存することなく開発を進めることができるので、生産性の向上、品質の向上を見込むことができる。

(3) ODBC+PowerBuilder によるレスポンス遅延の問題

ODBC 利用に関しては、これまでも多くの指摘があるように、OCX での OLE 手法に比べて処理のレスポンスが遅延することが確認されている。(詳細は前述の図表3を参照)

このレスポンスの遅延に関しては、3つの項目について検討する必要がある。

a. ODBC ドライバーの問題

今回の販売管理システム開発で使用している ODBC ドライバーは、ODBC2.0 という規格に準拠したドライバーである。(InterSystems 社提供)

現在、InterSystems 社では、処理の高速化を図った新しい ODBC ドライバーの動作テストを行っているとの情報を得ている。これは、ODBC3.0 という規格に対応したものである。

この新しい ODBC ドライバーの処理が高速になれば、レスポンス遅延に関する問題が緩和される。

b. ハードウェアの問題

データベース・サーバーについて、CPU スピードの異なる3つのマシンで処理速度を計測した結果、CPU スピードが速くなるほどレスポンス時間も短縮されることが確認されている。(表4を参照) また、実装するメモリー容量もレスポンス時間に対する重要な要素でもある。(多くのデータをメモリー上に配置した方が処理速度は短縮できる為。)

(表4) ハードウェア別処理速度計測結果

ハードウェア構成		処理速度 (msec)
CPU	メモリー容量	(10回の平均値)
Pentium 60MHz	64MB	219
Pentium 90MHz	24MB	188
Pentium 166MHz	64MB	132

*OS: 全て Windows95

*処理内容: 部品マスタ(40,000件)から1件を抽出する処理

従って、実際に構築するシステムの規模に対して見合った性能(CPU速度、メモリー容量、等)のハードウェアを選択することが肝要である。

c. システム全体の問題

個別のレスポンス時間を検討すると共に、システム全体のスループットを低下させることなく、いかに維持するかが非常に大きな問題となる。

そこで、データ処理を行うアプリケーションサーバーを1台構成から複数台構成にして負荷分散を図る、あるいは、全データを検索する処理を極力避けるといった、システム全体での創意工夫が必要になると考えられる。

[今後の展望]

システムの構築、維持管理をいかに低コスト(安価、単納期、容易なメンテナンス)に、しかも高いパフォーマンスを得られるかを考えた場合、OpenM/SQLを中核に、開発ツールとして VisualBasic と PowerBuilder をうまく使い分けてシステム全体を構築していくことが肝要であると判断している。

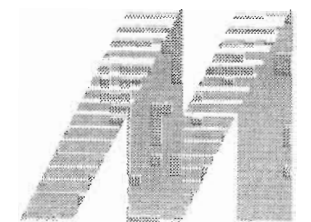
下記は各ツールの特徴をまとめたものである。

(表5)

	VisualBasic+VisualM・OCX	PowerBuilder+ODBC
納期	長	短
開発の容易さ	難	易
メンテナンス性	難	易

開発要員確保	難	易
処理スピード	高速	中速

*この表は、VisualBasic(OCX)とPowerBuilder(ODBC)間での比較結果である。



はやわかりデータベースシステム動向97

はじめに、データベース管理システム(DBMS)の標準規格である SQL3 の基本、SQL3 の SQL/CLI に準拠した ODBC、SQL3 の SQL/PSM に準拠したストアード・プロシージャを説明する。つぎに、インターネット世界で DBMS と関連する技術である CGI, WWW & RDBMS & Lotus Notes, Java, JavaDBC と ActiveX と展開する。最後に、素人にも視覚的かつマウス操作ひとつで、データベースを簡易に構築できるデータマネージャと Jet データベース・エンジン、プログラム・コードなしでデータベースの構築と操作ができるオブジェクト指向 API の DAO/DAC, さらにその技術をクライアント/サーバ環境に展開した RDO/RDC, OLE DB と ADO(ActiveX Data Objects)を紹介する。

Keyword :

- (1) SQL3, ODBC(Open Data Base Connectivity : SQL3/CLI), ODBC datatype, Storedprocedure (SQL/PSM), SQL/Object, middleware under Client/Server
- (2) Internet, CGI (Common Gateway Interface), WWW(World Wide Web), Lotus Notes, Java, Java Class library, DCOM (Distributed Component Object Model) JavaBeans, JDBC(Java Data Base Connectivity), J/SQL, JDBC-ODBC bridge, VBScript(Visual Basic Script), ActiveX
- (3) Visual Basic, Data manager, Jet database engine, DAO(Data Access Objects), DAC (Data Access Control), RDO(Remote Data Objects), RDC (Remote Data Control), OLE DB, ADO(ActiveX Data Objects)

1.1 SQL3

関係データベース管理システム(RDBMS)のための言語として SQL (Structured Query Language: 構造化問合せ言語) は、1970 年代に米 IBM 社の研究プロジェクト SEQUEL (Structured English QUery Language)に由来する。1981年に国際標準化機構(ISO)より規格化された(図1)。1987年に制定した国際規格では、

```

図1 SQL言語の問い合わせ形式
SELECT <値式>,<値式>
FROM <テーブル参照>,<テーブル参照>
WHERE <探索条件>
SELECT .....
    
```

RDBMS を用いる一般のアプリケーションが必要とする機能だけの言語仕様の開発を行い、これが SQL-89 となった。その後、①スクロール・カーソル②動的 SQL (実行時に SQL 文をサーバに送り、実行し、結果を得る方法) ③テンポラリ表 (セッションの間だけ存在するユーザ定義の表のこと、一時表とも呼ぶ) ④色々な組み込みデータ型の支援 (DATE, TIME, TIMESTAMP, BIT, CHARACTER, 他) 等が追加されて、1992年に開発が終了した。これが SQL-92 と呼ばれ、現在の最新版として SQL2 となった。SQL3 の今後の予定は、1997年に DIS (Draft International Standard) とすることを予定している。SQL3 は大きな仕様になるために9冊分冊の構成で、3分冊めの SQL/CLI、4分冊めの SQL/PSM と8分冊めの SQL/Object について説明する。

SQL3 の2つの焦点は、SQL2 の拡張機能とオブジェクト指向の導入である。SQL3 の主な拡張機能は、SQL/CLI(Call Level Interface) と SQL/PSM (Persistent Stored Module : 永続的格納モジュール)である。SQL/CLI は、従来の SQL がソースコードレベルであったのに対して、SQL の機能呼び出すためのインターフェースを標準化することによって、SQL の実行コードの可搬性を提供する。例えば RDBMS とアプリケーション・プログラムとのアクセス用のインターフェースを規定したもので、米マイ

クロソフト社の ODBC(Open DataBase Connectivity)や 米ボーランド社の IDAPI(Integrated Database Application Programming Interface)に相当する。またインターネットで使用される JAVA 言語から SQL 言語にアクセスする JDBC(Java DataBase Connectivity)が開発された。

SQL/PSM は、従来のクライアントが SQL を発行してサーバが対応していたものを、サーバに SQL をモジュールやルーチンとして格納し、作成・変更および削除を可能にした。モジュールの中で複数の SQL を含めることが可能になるほか、複合文(BEGIN...END)、一般の制御文・実行制御文(IF,CASE,WHILE,FOR)、関数呼び出し(CALL,RETURN)や代入文(SET)が可能となる。一般的に"ストアード・プロシージャ"として知られる技術であり、米オラクル社の Oracle7 の PL/SQL や米サイベス社の Transact SQL が相当する(図2)。

```

図2 SQL/PSM の事例
CREATE MODULE transaction
LANGUAGE SQL
SCHEMA catalog1,schema2.
DECLARE C1 CURSOR FOR...
PROCEDURE get_balance (SQLCODE,
account_# INTEGER
balance DECIMAL;
SELECT balance INTO balance
FROM accounts
WHERE account_#=account_#;
PROCEDURE withdraw (SQLCODE,
account_# INTEGER
amount DECIMAL(15,2);
.....
.....
END MODULE;
    
```

SQL/CLI と SQL/PSM 以外の分冊としては、SQL/Binding の主な機能は SQL モジュール、動的 SQL と埋め込み SQL(3GL のプログラム言語に直接 SQL を埋め込む方法)である。SQL/Transaction は、RDBMS が管理するトランザクションを、外部から制御するためのインターフェースを規定する。つまり、RDBMS と TP モニター (トランザクション専用のプロセス) とのインターフェースの標準である。

オブジェクト指向の導入は、従来の RDBMS の上に拡張として実現している点である。その概略は次の3つで、①オブジェクトモデル: 抽象データ型(ADT: Abstract Data Type)とは、利用者によるユーザ定義データ型であり、SQL は組み込みのデータ型とまったく同じく操作が出来る。抽象型を用いることにより、属性と操作を一体に隠蔽しカプセル化が出来る。②アクセス制御: C++同様に、PUBLIC(公開)、PRIVATE(限定公開)、PROTECTED(非公開)の3種類である。③クラス階層: 抽象データ型は、クラス階層を指定することが出来て、一つ以上の上位型(super type)の副型(subtype)になることが出来る。つまり型継承として、多重継承(multi-inheritance)がサポートされる。継承の内容は、属性からなる構造(attribute)と振る舞い(method)の両方である。多様性(polymorphism)は、呼び出しの方法は同じであるが、異なる操作が施されることを意味する機能もサポートされる。

1.2 ODBC は SQL/CLI に準拠

従来の SQL が、COBOL 言語などのソースコードレベル埋め込みであったのに対して、SQL/CLI は、SQL の機能呼び出すためのインターフェースを標準化することによって、SQL の実行コードの可搬性を提供する。ODBC のアーキテクチャーは、各種の DBMS の影響を受けないように、データベースアクセス方法をコンポーネント化することで、アプリケーションと DBMS を論理的に分離した。ODBC で作成された1プログラムで、異なるコンピュータにある複数の RDBMS がアクセスできる(図3)。

アプリケーションからは共通の ODBC インターフェース呼び出し

アプリケーション層は、VisualBasic 言語等のアプリケーション言語から ODBC インターフェース経由で情報をやりとりし、SQL 文を渡すために API 関数を呼び出して結果を受け取る。ODBC インターフェース層 (Version 2.10) が規定しているのは、SQL の文法、データベースに対する各種の API、エラーの種類とエラー・コード、データベースが扱うデータ型、文字列操作や算術計算などのスカラー関数等である。SQL 文法は 1992 年版をベースにしている。API は、SQL/CLI をベースにした関数群に、スクロール可能なカーソルや非同期処理機能などを追加した仕様になっている。データ型は、通常の整数、

浮動小数点、可変長の文字列以外に、拡張タイプとしてビット (BIT)、可変長バイナリ (VARBINARY、LONGVINARY)、日付・時刻 (DATE、TIME、TIMESTAMP) がある。スカラー関数は、文字列関数 (ASCII、DIFFERENCE、REPLACE、SUBSTRING、LENGTH 等)、数値計算関数 (ABS、COS、POWER、ROUND、SQRT、EXP、SIN 等)、日付・時間とシステム関数から構成される。

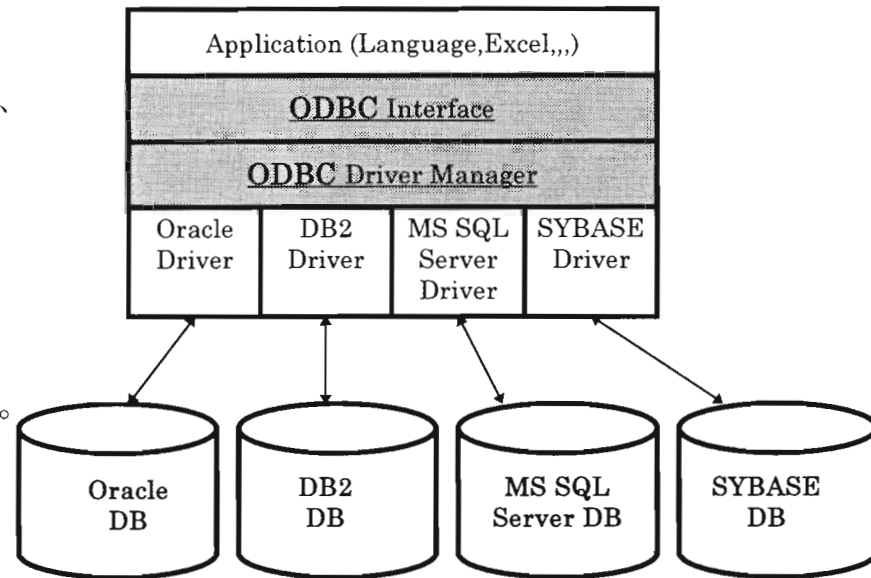


Figure 3: ODBC Architecture (without networking)

ODBC ドライバ・マネージャは各 DBMS への振り分け

ODBC ドライバ・マネージャは、マイクロソフト社がダイナミックリンクライブラリ (DLL) ファイルで提供し、複数の DBMS にまたがる処理要求を分割して各 DBMS 向け ODBC ドライバに送る処理をする。処理要求を分割する時は、実行環境設定も行う。ODBC ドライバ・マネージャによって、アプリケーションと各 DBMS 向け ODBC ドライバの結合が保証される。複数のアプリケーションのために、複数のドライバをロードし、またアンロードする。このロード手順は、アプリケーションがデータソース名にアクセスする必要がある場合、データソースのファイル名を探してロードされる。データソース名とは、ODBC ドライバの接続設定名と DBMS の名前や使用する OS、ネットワークのプロトコル名などの総称名である (図 4)。

ODBC ドライバは、ODBC のファンクションコールの実行や、データソースへの SQL の送信を行い、結果をアプリケーションに返す。必要に応じて、対象の DBMS の SQL 文法に合わせて変換する。ODBC ドライバマネージャによってロードされる。つまりデータソース名とやりとりする DLL である。クライアント PC に ODBC ドライバをセットアップする方法には、2 種類ある。Excel 7.0 や Access 95 などのインストール時にカスタム・インストールを選択して、ODBC ドライバをセットアップする方法と、Oracle 等の DBMS のクライアント専用のインストーラ等によってセットアップする方法がある。

```

/* ODBC program in C language */
#include < sql.h>
#include < windows.h>
#include < sqlext.h>
....
int WINAPI WinMain ( HANDLE hInstance , ..... )
{
    RETCODE rc ; /* return code for ODBC functions */
    HENV henv ; /* environment handle */
    HDBC hdbc ; /* connection handle */
    HSTMT hstmt ; /* statement handle */
    .....
    SQLAllocEnv (&henv) ;
    SQLAllocConnect (henv,&hdbc) ;
    SQLConnect (hdbc,"hello", SQL_NTS, NULL, 0, NULL, 0) ;
    SQLAllocStmnt (hdbc,&hstmt) ;
    SQLExecDirect (hstmt, "select * from hello", SQL_NTS) ;
    for (rc = SQLFetch ( hstmt ) )
    {
        SQLGetData(hstmt, 1, SQL_C_CHAR, szData,
            sizeof(szData), &cbData) ;
        MessageBox ( NULL, szData, "ODBC", MB_OK )
    }
    SQLFreeStmnt ( hstmt, SQL_DROP ) ;
    SQLDisconnect ( hdbc ) ;
    SQLFreeConnect ( hdbc ) ;
    SQLFreeEnv (henv) ;
    return ( TRUE ) ;
}
    
```

Figure 4: ODBC programming list in C

ODBCによるデータ変換と精度

C/S(Client/Server)の環境で、ODBC を使用していた場合に異なるソフトウェア間におけるデータ変換による精度が問題になる。データ変換は、最大3回考えられる。サーバー側のデータベース (RDBMS) とミドルウェア、クライアント側は、ミドルウェアと ODBC ドライバ、ODBC ドライバとアプリケーションである

1.3 ストアド・プロシージャ (Stored Procedure)

クライアント/サーバ方式 (C/S) で、クライアントから SQL をサーバに要求すると、① SQL の構文解析 ② SQL 文法チェック ③ SQL の妥当性チェック ④ 正規化 ⑤ 権限チェック ⑥ 最適化 ⑦ コンパイル ⑧ 実行、のような順で実行される。データの流れとしては、一つの SQL の検索要求に対して、クライアントからサーバ、サーバからデータベースをネットワークを経由してデータが流れ、結果は逆にデータベースからサーバ、サーバからクライアントと、検索要求回数分ネットワーク上をデータが流れる。また実行までには①から⑦まで毎回実施される (図 5)。

ストアド・プロシージャ (Stored Procedure) は、複数の SQL によって行う処理をサーバの DB (データベース) のディクショナリに登録したプログラムである。クライアントからの要求に対して①ストアド・プロシージャの呼び出し ②権限チェック ③パラメータ置換 ④実行、の順で実行される。明らかに SQL のみの方法より実行手順が少ないことが理解できる。データの流れとしては、一つのストアド・プロシージャ要求に

対して、クライアントからサーバは一回のみ、サーバからデータベース間を複数回ネットワークを経由してデータが流れる。

検索結果は逆にデータベースからサーバ間を複数回、サーバからクライアントは一回のみ、ネットワークを経由してデータが流れる。これによりストアド・プロシージャの利点としては4点ある。①逐次処理の効率化②ネットワークトラフィックの軽減③保守の向上④ロジックのカプセル化と再利用性である。有効的に機能する環境は3点ある。①更新対象や更新方法が一定②テーブルとテーブルの関係に一定のロジックが成り立つ③大量のバッチ業務である。

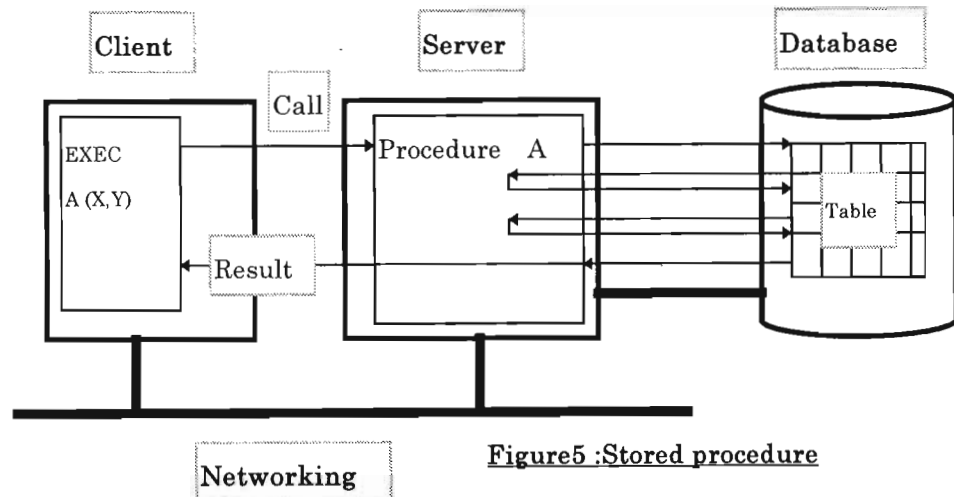


Figure5: Stored procedure

1.4 ミドルウェア

C/S環境におけるミドルウェアについては、添付A: "Client/Server environment"を参照。

2.1 CGI, WWW & RDBMS & Lotus Notes in Internet

CGI (Common Gateway Interface)

WWW(World Wide Web)クライアント(ブラウザ)の要求に応じて、WWWサーバがデータベースから目的のデータを検索するときは、WWWクライアント==>WWWサーバ==>インターフェース・プログラム==>外部プログラム==>データベースの順にデータが流れる。WWWサーバからデータベースにアクセスする方法である、インターフェース・プログラムには2種類ある。CGI(Common Gateway Interface)プログラムとAPI(Application Programming Interface)プログラムである。CGIプログラムをCGI・スクリプトあるいはゲートウェイ・スクリプトとも呼ぶ(図6)。

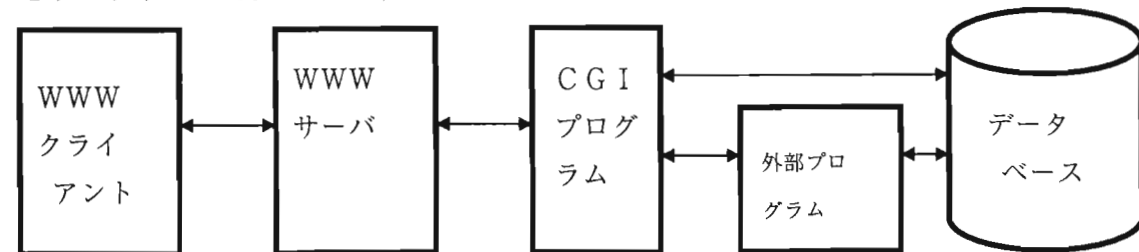


Figure 6: WWW & CGI

APIプログラムはWWWサーバ内専用APIとして、同一プロセス内で動作するためCGIプログラムより高速であるが、ベンダ固有である。製品としては、米ネット

スケープ社のNSAPI(Netscape Server API)、米オラクル社のOracle WebServer、米サイベース社のSybase Web sql、ジャストシステムのJust Office Server、NTTデータ通信のInterServなどがある。CGIプログラムはWWWサーバに標準で用意されているもので、連携アプリケーションの作成は簡易であるが、要求単位にゲートウェイ・プロセスを生成するために、処理速度が遅くなる。理由としては、WWWサーバはWWWクライアントとセッションを張る概念がなく、URL(Uniform Resource Locator)による要求を受け取るたびにホームページのHTML(Hyper Text Markup Language)ファイルを転送したり、CGIプログラムを起動するだけである。またファイアウォール(機密保護のための壁)経由のプロキシ(代理)ログインには同一利用者かどうか管理できないからである。CGIプログラムはUNIXにおいてはスクリプト言語(例えばPerl)やC言語やCOBOLやデータベース専用言語にてプログラミングができる。

CGIプログラムはWWWサーバと外部プログラムとの間に入ることにより、決められたデータ形式の変換に役立っている。つまりCGIプログラムを経由しないで外部プログラムを呼び出し、その結果をWWWクライアントに送ったとしても、そのデータが何なのかかわからない、文字なのか画像なのか、表示できない。CGIプログラムはWWWサーバ、WWWクライアントへとデータが渡されて、ブラウザに表示できるようにする。WWWクライアントからWWWサーバの呼び出しはURLで、WWWサーバからWWWクライアントへの出力はHTMLである。

```

Figure7: C言語によるCGIプログラム
#include <stdio.h>
void main()
{
printf("Contents-type:text/html|\n");
printf("\n");
printf("<HTML>\n");
printf("<Title>HelloWorld!</Title>\n");
printf("<BODY>\n");
.....
}
    
```

"Figure7: C言語によるCGIプログラム"を参照すると、printfの印刷内容は、WWWクライアントのブラウザにわかるようなHTML形式の標準出力で返却されている。HTML出力部は、HTMLヘッダ出力とHTML本文の出力に分けられる。通常のHTMLテキストならば右の例でよいが、GIF等の画像表示をする場合は、Content-typeを変えなければならない。

WWW & RDBMS

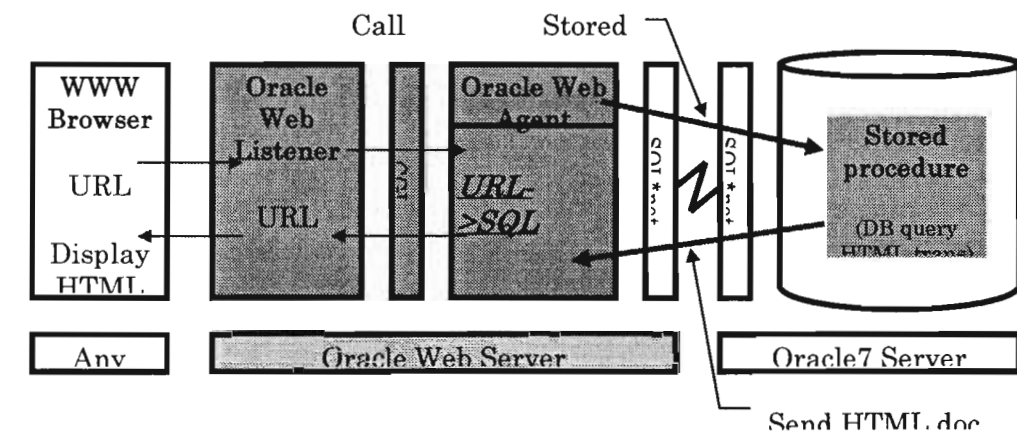


Figure 8: WWW-RDBMS (Oracle7)

Oracle WWW Listener:

DNS(Domain Name Service)を内蔵して、一般のWWWサーバと変わらない。

メモリ管理をマルチスレッド機能をもっている。

Oracle WWW Agent :

WWW Server と Oracle7 Server を連携するソフトウェア。

WWW Server とは CGI (Common Gateway Interface) を使ってやり取りする。

WWW Listner 以外の WWW Server の使用も可能。

WWW Oracle/Oracle7 Server が同一マシンの場合は SQL*net は不要。

URL (Uniform Resource Locator) を解釈して Oracle7 Server のストアード・プロシージャを呼び出す。

(例: http://www.novartis.com/ndp/owa/smss.pass?person=imaizumi novartis.com というサーバーの ndp という Oracle WWW agent サービスが、Oracle7 Server の mss.pass というストアード・プロシージャを起動する SQL 文を person=imaizumi という引数を付けて生成・発行をしてくれる)

RDBMS & Lotus Notes

Lotus Notes Client から RDB Server にアクセスする方法

① Client (フロントエンド・ソフトウェア) :

PC RDBMS の R D B で検索する操作を記述して、検索結果を Notes のフォームに貼り付ける。

② Client (Lotus Script + ODBC) :

Lotus Script でアプリケーションを記述して、ODBC 経由で R D B を検索する。

③ RDB server より Notes Server にバッチ転送 :

2.2 Java

Java の概要

インターネットの世界では、WWW (World Wide Web) の動きがさかんである。利用者が WWW サーバへアクセスする時は、Mosaic や Netscape Navigator (米国ネットスケープ・コミュニケーションズ社) 等の WWW ブラウザを使用し、URL (Uniform Resource Locator : 例えば http://www.novartis.com) を指定して、HTML (Hyper Text Markup Language) で書かれたファイルをロードし、表示する。処理はすべて WWW サーバが担当する。例えば利用者が WWW ブラウザで入力したデータのチェックは、WWW サーバでチェックしなければならない。HotJava は米国サン・マイクロシステムズ社が開発した WWW ブラウザであり、そのベースとなるのが Java 言語 (ジャバ) である (図 9, 10)。Java を組み合わせれば WWW ブラウザでチェック出来て、WWW サーバにアクセスしなくてすむ。

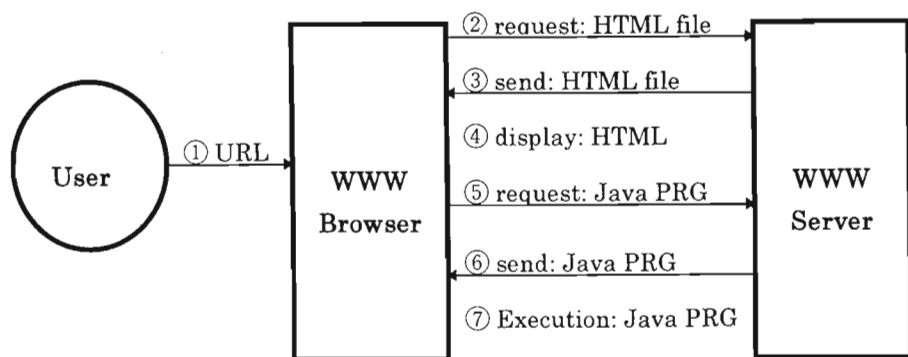


Figure 9 : Between HotJava and Java in WWW

Java はオブジェクト指向プログラミング言語で、1990 年に James Gosling , Bill Joy や Guy Steele らが開発され、1995 年 5 月に公開された。名前はジャバ (英語読みはジャバ) ・ コーヒーからとり、 “気楽に” という意味が込められている。

Java Class Library

Java のプログラム作成で標準的にクラスを定義したものである。

Java の最上位クラス (superclass) は、 “Object” クラスという名前である。

Java のクラスはパッケージという名前空間に属する。パッケージの実体は、ディレクトリと URL で記述されたネットワークの位置 (ディレクトリ) である。パッケージの名前は、ディレクトリの階層と対応した名前になっている。

ライブラリは全部で 8 種類ある。一番使用するライブラリは、java.applet (ホーム・ページを作成する時に使用)、java.awt (ウインドウ、メニューやボタンなどの GUI ツール、色やフォントなどの描画の属性、ダイアログ・ボックス)、java.applet.image (イメージ処理)、java.awt.peer である。その他には、java.io (ファイル処理)、java.lang (言語仕様に関するクラス集合)、java.util (便利なデータ処理)、java.net (ネットワーク処理) がある。

java.io (ファイル処理)、java.lang (言語仕様に関するクラス集合)、java.util (便利なデータ処理)、java.net (ネットワーク処理) がある。

Java Beans

異なるコンピュータがネットワーク間で接続されている時、各々の分散オブジェクトの相互連携技術は、米 Microsoft 社の分散オブジェクト DCOM/OLE (Distributed Component Object Model/Object Linking Embedding) とオブジェクト指向技術の標準団体である米 OMG (Object Management Group) の CORBA (Common Object Request Broker Architecture) 規格がある。米 IBM 社と米 Apple 社の OpenDoc 等は CORBA で実証済みである。異なるコンピュータ間で同じ名前のオブジェクトが存在しても、相互連携・再利用される分散オブジェクトがユニーク ID をもっていれば問題はない。例えば IP アドレスを付加すれば識別できる。分散オブジェクトではインターフェース定義言語である MIDL (Microsoft Interface Definition Language) と IDL を採用している。Java の世界でも当然分散オブジェクトへの規格が必要である。

Java 言語では、他の製品で作成されたオブジェクトと通信するために CORBA 準拠の Java IDL (Java Interface Definition Language) と、ネットワークに広がった Java アプレット同士の通信インターフェース Java RMI (Java Remote Method Invocation) 作成をした。Java IDL は通信を仲介するミドルウェアである ORB (Object Request Broker) を利用する API (Application Program Interface) である。ORB はディレクトリ・サービスを備えているので通信相手のオブジェクトがどのコンピュータで動いているかを記述する必要がない。Java における他の分散オブジェクトとの連携や再利用性の規格を “Java Beans” とぶ。 “Beans” はインターネット上のコンポーネント技術であり、 “the `write once , run anywhere` architecture of Java with a `write once , reuse anywhere` component model” と呼ばれる。 “Java Beans” に連携する主な米国ベンダーは、IBM (Visual Age for Java) , ORACLE (NCA) , Borland (Latte) , Netscape (Netscape ONE) , Informix (DataBlade Developers Kit) , Sunsoft (Java Workshop) ,,, である。

米マイクロソフト社は、OLE コントロールを ActiveX Control に名前を変えて、Java Beans と同じ機能をもっている。米 JavaSoft 社 (米サン・マイクロシステムズ社

```

<html>
<head>
<title>Circle applet</title>
</head>
<applet codebase="hpage" code="circle.class"
width=400 height=300>
</applet>
</html>

```

Figure 10: Java program in HTML
codebase: java program directory, code: java program name ,width/height: display dot size

Java 事業部が独立)は、JavaBeans 対応ソフトウェア部品と ActiveX 対応ソフトウェア部品を相互に呼び出し合えるようにするために、“JavaBeans ActiveX Bridge”を開発した。これは ActiveX のインターフェースを JavaBeans の形式に変換する。例えば、JavaBeans ActiveX Bridge を使用すると、Excel95 に ActiveX が JavaBeans 仕様の Java ソフトウェアを認識し、イベント等を受け入れる。

JDBC(Java Data Base Connectivity)

米 Sun 社が JAVA 言語からデータベースにアクセスする JDBC(Java DataBase Connectivity)を開発した。その方法には2種類あり、JDBCが従来の ODBC の役割をする方法、つまり Java アプレット→JDBC→各 DBMS のドライバ→各 DBMS のデータベースと、従来の ODBC を経由する方法、つまり Java アプレット→JDBC→ODBCブリッジ→ODBC→各 DBMS のドライバ→各 DBMS のデータベースがある(図11)。JDBCは複数のデータベースへの同時接続、トランザクション管理機能、ストアドプロシージャへの問い合わせ機能やコール機能を備えているが、SQL文をJavaに直接マップするわけではない。

米オラクル社は、Java からデータベースへのアクセスに3つの製品を発表した。①J/SQL: JDBC よりきめ細かく正確な構文を使って、Java アプリケーションからデータベースへアクセスをする J/SQL を開発した。SQL は知っているが JDBC を知らない利用者が書くのに便利なプリコンパイラである。②Thin-ClientJDBC: Java だけでインプリメントされた JDBC ドライバである。③Java Stored Procedure: SQL/PSM に準拠して、Java でストアド・プロシージャやトリガーを記述できる。

Microsoft の戦略

Java はマイクロソフト社の独占に対抗できる今世紀最後の言語と呼ばれることがある。そのマイクロソフト社は、インターネットの世界では少し遅れ気味なので、2つの路線を考えている。ひとつは、Java のライセンス導入を合意したことである。サン・マイクロシステムズ社、ネットスケープ・コミュニケーションズ社、マイクロソフト社の3社はインターネット標準言語として、自社の言語を、インターネットの標準に関する諮問機関である

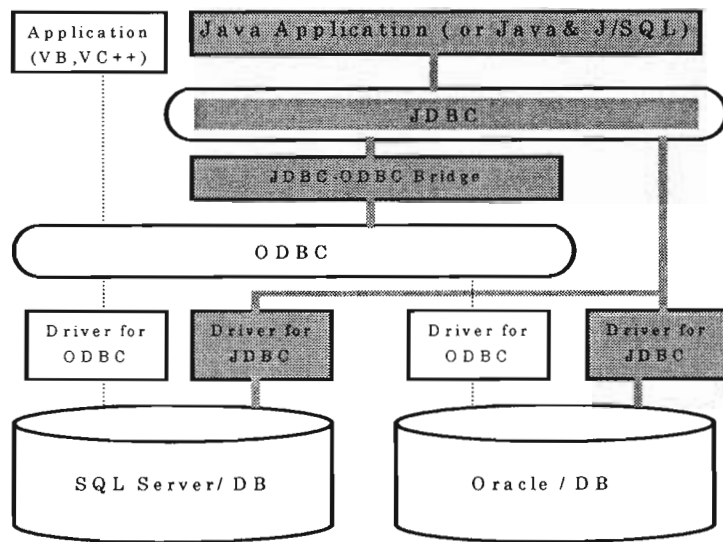


Figure 11 : JDBC Structure

W3C(World Wide Web Consortium)と IEFT (Internet Engineering Task Force)に提案している。

VBScript (Visual Basic Script)

もうひとつ JavaScript に対抗して、従来の自社の VB 言語を継承している VBScript (Visual Basic の略)を開発していることである。VBScript はブラウザでのプログラム実行、ActiveXcontrol や Java アプレットを制御できる。VBScript の特長は、ActiveXcontrol に対応している。VBScript の言語仕様は、VBA(Visual Basic for Applications)のサブセットになっている(図12)。WWW サーバでから外部を呼び出す

方法は2種類ある。①Win32API で記述した DLL (Dynamic Link Library) を呼び出すインターフェース ISAPI (Internet Server API)と②同じく Win32API を用意した OLE Automation である。

Table12 : Java, JavaScript & Active X control

Script Language		
JavaScript	Java applet	SunMicrosystem
Visual Basic Script	Java applet ActiveX control	Microsoft
Software parts		
Java applet	Java language	byte code
ActiveX control	C++ language	binary code

ActiveX Control

VisualJ++は名前から想像できるように VC++のビジュアル開発環境に似ている Java 言語向けと言える。VisualBasic は、V5.0 から従来の OCX(OLE コントロール)を ActiveX Control に名前を変え、RemoteOLE を DCOM (Distributed Component Object Model)に名前を変えた。Active X Control はインタープリタで IE (Internet Explorer)3.0 上で動作し、1996 年末には UNIX 版と Macintosh 版が販売される。クライアントには VB V5.0 のランタイム版が必要である。ActiveXcontrol はクライアント・サーバ型の概念である。つまり、あるひとつの動作要求(ボタンを押す)に対しての該当のプログラムは、必ずサーバの役目をするフォーム(コンテナとして実装)の制御のもとに起動・通信される。JavaBeans はネットワーク型の概念である。すべて対等のプログラムの部品として搭載されて、直接メッセージ等のやりとりをするフォームもひとつのプログラムの部品として扱われる。

3.1 プログラム・コードなしデータベース作成簡易ツール

データベースの理論・開発経験等がなくてもビジュアル(視覚的)にデータベースの構築とデータベースの検索ができる簡易ツールがある。米マイクロソフト社が提供する Visual Basic version4.0 言語(以降 VB)の機能をもとに、データベースを構築する時にプログラムコードを記述しないデータマネージャ、データベースの操作が容易である DAO (Data Access Objects)と DAC(Data Access Control)について記述する(添付B)。データマネージャ

データベースの構築には、①ACCESS などのデータベースのプログラムを書く。②DAO のプログラムを書く。③VB に用意されているデータマネージャ (DATAMAN32.EXE(32bit), DATAMGR.EXE(16bit)) の3つがある。ここではデータマネージャを記述する。データマネージャは米マイクロソフト社の Jet データベースエンジン(以降 Jet)という構造体を通じて、ACCESS(米マイクロソフト社の DBMS)のファイル形式*.mdb のテーブルが構築できる。構造体とは、複数のデータを集めて一つの変数のように扱える仕組みで、VB ではユーザ定義型変数と呼ばれる。構築はデータマネージャを選択後に、Jet を起動するとテーブル構築のためのダイアログボックス(窓)が現れ、一つ一つのデータ項目に対してフィールド名前・データ型・データのサイズを入力する。一つのテーブルの中でデータ項目が重ならないための主キーを決める(社員テーブルならば社員コード)、検索スピードアップするデータ項目の選択のインデクシングを決める。主キーは必須である。

テーブル定義ができたならテーブル名を指定して開くと、データ入力をするためのダイアログボックス(窓)が現れ、一件づつデータを入力できる。入力されたテーブルの内容を検索するのも、データマネージャの"SQL ステートメント"を起動して、SQL 文

1997年7月11日～12日 今泉 幸雄 (Yukio Imaizumi)

(Structured Query Language: 構造化問い合わせ言語)を記述することによって出来る。

データマネージャはデータベースの知識やプログラミング経験がなくても、表計算のデータを入力する操作性でデータベースの構築からデータ入力、検索までが容易にできる機能である。正に利用者向けの簡易ツールである。

Jet データベース エンジン (Jet)

DAO 経由の Jet を利用すると *.mdb 形式または Fox、dBASE、Excel、Btrieve、Paradox、区切り記号付きテキストをはじめとするその他の組み込み可能な ISAM 形式のデータベースを操作できる。また、Jet を使用して、米 Microsoft 社 SQL Server にアクセスしたり、同じ DAO のコードを記述して、ODBC ドライバ (Open Data Base Connectivity) でアクセスでき、その他のデータベースにもアクセスできる。Jet によって設定され管理されるセキュリティ設定を使って、1つ以上の *.mdb データベースへのアクセスを制限することができる。Jet には、WindowsNT/Windows95(32bit)で動作する Jet3.0 エンジン、Windows3.1Jet2.5(16bit)で動作する Jet2.5 エンジンがある。

DAO (Data Access Objects)

データ アクセス オブジェクト (DAO) を使用すると、プログラミング言語を使用してデータベースのデータにアクセスして操作したり、データベースまたはデータベースのオブジェクトと構造を管理することができる。

DAO では、データベース、テーブル、フィールド (カラム) などがオブジェクトになる。オブジェクトはプロパティと呼ばれる属性値とメソッドと呼ばれる実行するコマンドを持っている。オブジェクトの後ろにプロパティやメソッドを書く。例えば、fld には既に情報が格納されているとすると fld.Name、fld.Value という表現になる。あるオブジェクトのプロパティやメソッドが別のオブジェクトになることもある。コレクションは同じ種類のクラスが集まったオブジェクトである。コレクションに含まれている各オブジェクトを参照するには、添字あるいは名前を指定する。例えば、rst.Fields(1) あるいは rst.Fields!ISBN と書く。

Dim ステートメントおよび Set ステートメントを使用すると、新しいデータ アクセス オブジェクト変数を宣言し、任意のオブジェクトを参照することができる。たとえば、Database オブジェクト、ダイナセット タイプの Recordset オブジェクト、および Field オブジェクトを参照する (図 13)。

図 13: Sample of DAO programming

```
' define (object...)
Dim ws As Workspace
Dim dbs As Database
Dim Connect As String
Dim SQL As String
Dim flds As Field
Dim fldcount As Integer
' login database
Set ws = Workspace(0)
Connect = 'ODBC:DSN=' & Ptrx.svrname & ";UID=" & _
Ptrx.userid & ";PWD=" & Ptrx.password & _
& ";DATABASE=" & Ptrx.dbname
Set dbs = ws.OpenDatabase ("", False, False, Connect)

Function sql_get_isbn (iisbn As Long, book As Book,
rauthor() As String) As Long
Dim rst As Recorder
Dim qd As QueryDef
.....
' stored procedure
qd.SQL = "selbook3 " + CStr(iisbn)
Set rst = qd.OpenRecorder
' query
Do Until rst.EOF
Title = rst!TITLE
instock = rst!instock
sold = rst!sold
price = rst!price
rst.MoveNext
Loop
rst.Close
```

1997年7月11日～12日 今泉 幸雄 (Yukio Imaizumi)

データ アクセス オブジェクトには既定のコレクションまたは既定プロパティが割り当てられている。たとえば、Recordset オブジェクトの既定のコレクションは Fields コレクションで、Field オブジェクトの既定プロパティは Value プロパティである。

すべての DAO は、Jet を表わす DBEngine オブジェクトを基にしている。アプリケーションが起動すると、Jet は、Workspace コレクションを使用して参照できる既定の DBEngine.Workspaces オブジェクト (Workspaces(0)) を作成する。Workspace オブジェクトは、必要なだけ追加して開くことができる。各 Workspace オブジェクトには、関連するユーザー ID とパスワードがある

DAC (Data Access Control)

Data Access Control をデータコントロールと呼ぶことが多い。これを使えば、プログラム・コードをまったく記述しなくても、簡単にデータベース操作ができる。

データコントロールに連結された、表示するための連結コントロールは、現在のレコードや、現在のレコードの前後のレコードの集合についても、1つ以上のフィールドのデータを自動的に表示する。データ コントロールを使うと、現在のレコードでのすべての操作を行うことができる。データ コントロールに異なるレコードに移動するように指示すると、すべての連結コントロールから変更内容がデータ コントロールに渡され、変更内容がデータベースに保存される。データ コントロールは要求されたレコードに移り、現在のレコードのデータを、データを表示する連結コントロールに返す。

データ コントロールは、空のレコードセットの追加、レコードの新規追加、既存のレコードの編集と更新、ある種のエラー処理など、偶発的に発生する多くの事柄を自動的に処理する。ただし、複雑なアプリケーションでは、データコントロールが処理できないようないくつかのエラー状況をトラップする必要がある。

連結コントロールになる、DB リスト (DBList)、DB コンボ (DBCombo)、DB グリッド (DBGrid) の各コントロールは、データ コントロールに連結されている場合には、レコードの集合を管理できる。いずれのコントロールも、複数のレコードを一度に表示または操作できる。チェック ボックス (CheckBox)、テキスト ボックス (TextBox)、ラベル (Label)、ピクチャ (Picture)、イメージ (Image)、リスト ボックス (ListBox)、コンボ ボックス (ComboBox) の各コントロールも連結可能で、データ コントロールが管理する Recordset オブジェクトの単一のフィールドに連結できる。

データコントロールを使ったデータベースへのアクセスでは、次の手順でデータベース・アプリケーションとデータベースを接続する。①データベースの形式を設定する。(Connect プロパティ)②データベースを設定する。(DatabaseName プロパティ)③テーブルを設定する (RecordSource プロパティ)。アプリケーションを起動すると、VB はデータ コントロールのプロパティを使って選択されたデータベースを開き、Database オブジェクトおよび Recordset オブジェクトを作成する。データコントロールの Database プロパティと Recordset プロパティを使用すると、新しく作成された Database オブジェクトと Recordset オブジェクトを参照して、連結コントロールと連結しているかどうかにかかわらず、どのオブジェクトもデータ コントロールを使用せずに操作できる。

VB が Jet を使って Recordset オブジェクトを作成するとき、その動作が完了するまで他の VB の動作やイベントは発生しない。データ コントロールを使って Recordset オブジェクトを作成した場合、または Recordset オブジェクトをコードで作成し、データ コントロールに割り当てた場合、Jet は自動的に Recordset オブジェクトにデータを取得する。データ コントロールをマウスで操作して、レコード間を移動したり、レコードセットの先頭または末尾に移動したりできる。データ コントロールでは、マ

ウスを使って Recordset オブジェクトの両端を超えることができない。また、データ (Data) コントロールにはフォーカスを設定できない。

3.2 クライアント/サーバのデータベース作成簡易ツール

クライアント/サーバ方式(C/S)環境で、データベースの理論・開発経験等がなくてもビジュアル(視覚的)にデータベースの構築とデータベースの検索ができる簡易ツールを記述する。米マイクロソフト社が提供する Visual Basic version4.0 言語(以降 VB)の機能をもとに、データベースの構築・操作が容易である RDO(Remote Data Objects)と RDC(Remote Data Control)と OLE DB(Object Linking Embedding Data Base)について説明する。ACCESS(米マイクロソフト社のDBMS)や Visual Basic version3.0 に搭載されている Jet データベースエンジン(以降 Jet)を使うオブジェクトの DAO(Data Access Objects)と DAC(Data Access Control)を説明時に用いる(添付B)。

RDO (Remote Data Objects)

DAOはJet経由でローカルとリモートにあるデータベースを操作する仕組みになっているが、ローカルなデータベースを操作するのに適している。DAOでリモートにあるデータベースを操作する時は、JetからJetのクエリ・マネージャとリモート・マネージャを通して、ODBC(Open Data Base Connectivity)のレイヤーを経由しなければならないので、構造的にパフォーマンスが悪くなる。C/S環境では、RDOがDAOより3～4倍処理スピードが高い結果も報告されている。しかしODBCドライバの出来、不出来でパフォーマンスは変化する。RDOはその名もRemote Data Objectsといい、リモートデータベースに操作することを前提としたオブジェクトである。複数ユーザーからの要求の並列処理が求められるサーバとクライアントの構成を簡単に作成できる。

RDOは、DAOと同じくODBCを利用することが出来るオブジェクトとして提供されている。しかしDAOはローカルエンジンとしてJetをもっているが、RDOはODBC API (Application Program Interface)をオブジェクトの形で利用する仕組みである。したがって、リモートデータベースにアクセスする時、特長や制限はODBCに依存する部分がある。

RDOの記述形式はDAOと同じく、データベース、テーブル、フィールド(カラム)などがオブジェクトになる。オブジェクトはプロパティと呼ばれる属性値とメソッドと呼ばれる実行するコマンドを持っている。オブジェクトの後ろにプロパティやメソッドを書く。あるオブジェクトのプロパティやメソッドが別のオブジェクトになることもある。コレクションは同じ種類のクラスが集まったオブジェクトである。JetのDBEngineオブジェクトに代わるrdoEngineオブジェクトには8種類のコレクションしかない。rdoEngine以下の主要なコレクションは、JetのDAOのオブジェクトと似ている(図14)。

RDOの主な特長は、①非同期処理：処理の要求と結果の受け取りを別々に実行する。

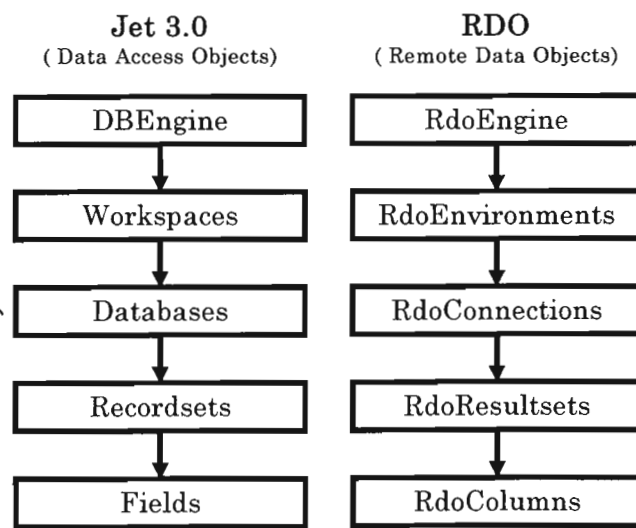


Figure 14: RDO objects

時間のかかりそうな検索要求をしておいて、プログラムは別の処理を続ける。②パラメータ付きの問い合わせとストアード・プロシージャ(複数のSQLによって行う処理をサーバに登録したプログラム)の処理。③DAOに比較して、必要なメモリ容量が少なく済む。

RDC (Remote Data Control)

RDCはRDOの機能をデータ連結OLEコントロール(DBGrid, DBList, DBComboなど)として利用できるように、VBのツールボックスのボタン登録をしている。このツールボックスを利用するので、プログラム・コードをまったく記述しなくてC/S方式のリモートにあるデータベースを、簡単に操作出来る。DACと外観がたいへん似ていて、多数のプロパティを共有している。RDCはRDOと同じくODBCを経由するので、RDCのパフォーマンスは、DAOとRDOの差に似ている。

OLE DB (Object Linking Embedding Data Base)

OLE DBは関係型データベース、表計算、電子メールやその他のファイル形式(Lotus Notes, ISAM, ...)等を、COM(Component Object Model)をベースに統一的に操作できるようにしたインターフェース仕様である。COMとは、米マイクロソフト社がオブジェクトを規格した仕様である。つまりアプリケーションからみるとOLE DBがLotus Notes, Word, ExcelやOracle等の関係型データベースが透過的にアクセスが可能になる。OLE DBの上位機能としてADO(ActiveX Data Objects)を用意している。ADOはオブジェクト指向データ・アクセスAPIとしては、DAOやRDOと似ているが、DAOとRDOは関係型データベースのみを対象としていた点が違う。

最後に 最近のデータベース管理システムの動向をとらえるに必要な基礎を話してから、つまみ食いになるかも知れないがわかってほしいのを書いてみました。最初の原稿は50ページを越えてしまったのでこの量に抑えるのに苦労しました。データウェアハウス、C/Sにおけるデータベース設計方法等は割愛しました。ここまで辛抱強く原稿への助言をしてくれた東京医科歯科大学の廣瀬先生に感謝します。

つくばと我孫子にて 今泉幸雄 (Yukio Imaizumi) June/8/1997

連絡先：茨城県つくば市大久保8番地 ノバルティスファーマKK 筑波研究所

tel : 0298-65-2205 Fax : 0298-65-2381

Client/Server environment

Client/Server environment		Client/Server environment		Client/Server environment		Client/Server environment	
Application / Client	Application (API)	Application (Visual Basic, Base SAS, SAS/ACCESS etc.)	Application (Java)	WWW browser	WWW server	Application	Application / Server
DDE	VBA Macro	ODBC interface	JDBC interface			DBMS API	Application / Server
KEY SQL	Excelent32	ODBC driver manager (ODBC.DLL)	JDBC Driver Manager			SAS dataset	DBMS/WWW
SQL*net	SQL*net or Excellent32	SAS ODBC driver	JDBC-ODBC bridge			ORACLE	MS/SQL Server SYBASE
SQL*net	SQL*net	Oracle ODBC driver	ODBC driver manager			SAS	DB-Library/CT-Library
SQL*net	SQL*net	MS SQL Server ODBC driver	ODBC Driver (each DBMS)			/SHARE*net	Net-Library
SQL*net	SQL*net	DB-Library	JDBC Driver (each DBMS)			SQL*net or Freeway/Sequelink	
SQL*net	SQL*net	Net-Library	ODBC driver manager				
SQL*net	SQL*net	CT-Library	SQL*net/DB-Library/CT-Library				
SQL*net	SQL*net	Net-Library	SQL*net/Net-Library				
TCP/IP/socket, IPX/SPX/socket, NamedPIPE							
TCP/IP, SPX/IPX, NetBEUI							
prg and prg com protocol stack							

Y.Imaizumi : Original(22/Feb/1997) . Revised(4/13/May/1997)

Windows95 : ODBC32.DLL or ODBC.DLL(16bit) WindowsNT : ODBC32.DLL or ODBC.DLL(16bit)+ODBC16GT.DLL+ODBC32GT.DLL

Original : May-10-1997 (Y.Imaizumi)
 Last saving : 06/19/97 12:11 (Y.Imaizumi)

添付 B

ODBC, OLE/DB, Jet (ADO, RDO, DAO)

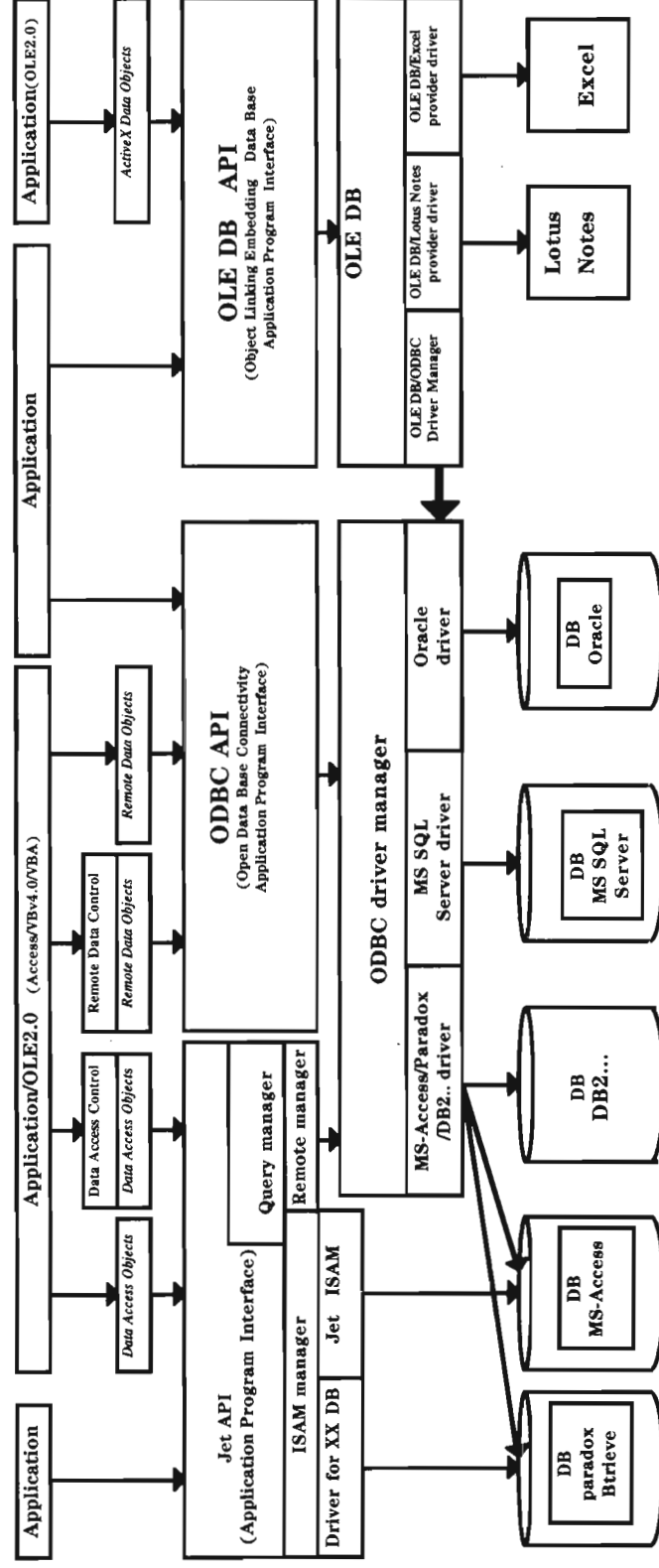


Figure : ODBC & OLE/DB & Jet

Jet database engine :

MS-Access database engine, ISAM file アクセスを提供, VB(Visual Basic) version 4.0 利用可能
 Windows3.1 (Jet2.5, Access2.0), Windows95/WindowsNT (Jet2.5, Jet3.0, Access2.0, Access95)

RDO (Remote Data Objects) & DAO (Data Access Objects) :

RDO は DAO の 3 ~ 4 倍は性能が高い。Excel や Word に、DAO ライセンスは含まれるが、RDO ライセンスは含まれない。
 VB v4.0 : Standard version はアクセス可能、Professional version は DAO ライセンスある、Enterprise version は RDO ライセンスある。

DAC (Data Access Control : ActiveX Control), RDC (Remote Data Control : ActiveX Control)

ADO (ActiveX Data Objects) :

ADO ライセンスは VB v4.0 で提供される

API (Application Program Interface) :

ODBC driver manager は C 言語、OLEDB は C++ 言語がインターフェース。

InterSystems ワールドワイド開発者向け戦略会議報告

A Report on the InterSystems Worldwide Developers Strategy Conference

今井敏雄

Toshio Imai

日本デジタルイクイップメント (株) 西日本第一統合システム部

Digital Equipment Corporation Japan Network and Systems Integration Services

〒530 大阪市北区中之島2丁目2番2号ニチメンビル TEL (06) 222-9211 FAX (06) 222-9408

This is a report on the 4th Annual InterSystems Worldwide Developers Strategy Conference which was held from April 13 thru 16, 1997 at Orlando, Florida. The main theme was WCTP --- World Class Transaction Processing. In the conference, InterSystems stated that their mission was to enable application developers to build and deploy WCTP systems and to do so faster and better than anyone else. I realized that InterSystems' strategy was based on understanding of the emergence of Post-Relational database and network-centric computing paradigm towards which we were now moving. In order for developers to develop WCTP systems, InterSystems strategically focuses on a high performance and scalable database for world class TP applications with rapid application development and its connectivity (interoperability & flexibility) using standard technologies such as GUI, SQL, Relational C/S, Web, OLE, Java, etc. Recently, they are importing Object technology into M to support new database access other than SQL and direct M. In this report, I summarized InterSystems' strategy presented in the conference, and tried to excerpt the Strategic Marketing's paper titled "The Emergence of Post-Relational Database", which was distributed at and for the conference.

キーワード: InterSystems, M Technology, MUMPS, Transaction Processing, Post-Relational Database

1 はじめに

1997年4月13日から16日まで、米国フロリダ・オーランドで開催された InterSystems 社の第4回年次ワールドワイド開発者向け戦略会議 The 4th Annual InterSystems Worldwide Developers Strategy Conference [1] に参加しました。同会議は、InterSystems 社が主に同社の V A R (Value Added Reseller) およびパートナーを集めて同社の製品戦略を訴え、それについて議論することを目的として年1回開催されている。以下、本年度の会議に参加して得た同社の製品戦略とトピックについて報告する。

2 メインテーマと戦略

今回のメインテーマは、WCTP (World Class Transaction Processing) でした。会議で InterSystems 社のミッションは、アプリケーション開発者がワールドクラスのトランザクション・システムを構築できるようにすること、それも他より、より効率的かつより良いシステムの構築を可能にさせることである、と述べていた。また、基本戦略として、M のデータベース・トランザクション機能を他の技術との差別化戦略要素として捉え、Post-Relational Database 技術の台頭の中で、複雑な現実世界をより効果的かつ実用的にモデル化する Multi-Dimensional データモデルの高トランザクション実現技術 (High Performance & Scalability) として M を位置付けている。

そして、ワールドクラスの TP システムを実現するために (すなわち、M 外の世界の中で M 技術を採用させるために)、M 内部からのダイレクト・アクセスに加え、データベース・エンジンへの M 外からの

Connectivity (Interoperability & Flexibility) を戦略的に重要視している。それを実現するアクセス手段として、GUI 技術/Visual M (VB, PB, Delphi, C++)、SQL 技術/M/SQL, Relational C/S (埋め込み SQL, ODBC)、OO 技術/Object M (OLE, Java, JDBC, Java Database, I S C Java Class)、Web 技術/WebLink (ISAPI, NSAPI) 等の Standard や De-facto Standard を取り入れようとしている。

特に最近 Object 技術を積極的に取り組んでいる。そこでは (Object M)、OLE や Java から M データベースを基にした Object データベースへのアクセスを実現すると同時に、M 内では、Class dictionary/Class Compiler を用意することで、M コードと内部データのカプセル化/オブジェクト化とその継承再利用を図ろうとしている。このことにより M 言語の簡潔さとインタープリタ性に頼った M プログラミングの生産性からの革新を試みようとしているように思える。

3 セッション

会議はオーランドの Omni Rosen ホテル内の三つのコンファレンスルーム (500 人収容可能なメイン会場と 300 名程の二つの分科会会場) で行われた。参加者は約 2 百数十名で、InterSystems 社から 50 名程が出席していた。13日 (日) の午後 6 時からの Welcome Reception で始まり 14、15、16 日の 3 日間にわたり計 31 のセッションが 3 つの分科会に分かれて開かれた。

セッションは 14 日午前 9:00 から 10:30 まで CEO Terry Ragon, COO John McCormick, VP Paul Grabscheid による Opening Session: World Class Database Strategy --- Post Relational Database for WCTP で始まった。そこでは、2 で述べた同社の戦略と、次世代の M として同社が現在開

発中の Open M/NextGen に対し、新しい技術、新しい製品に対して新たなイメージを与えるため、同製品を Caché (キャッシュ) と命名したことを発表した (Caché is a Post Relational Database)。その後、同社技術者からよりブレイクダウンしたセッションがもたれた。いくつかセッションのタイトルを拾うと、以下の通り。

- Getting to World Class by making The Move to Caché
- Intro to Object Technology
- Intro to Web Technology
- Tooling up for WCTP
- Caché product update
- Caché Road Map
- Diagnosing performance problems with Caché/ViewPoint
- Leveraging Microsoft's Active Technologies
- Global Solution marketing
- Panel Discussion: DTM, ISM, DSM Update
- Relational Development with Caché - High Performance SQL
- Visual M - Tips and Techniques
- InterSystems' New Post-Relational Positioning Strategy
- Advanced Open M/WebLink
- InterSystems Caché and Java
- Java and Caché/Object M

また、ゲストスピーカーとして以下の招待講演がありました。

- マーケティング調査会社 IDC (International Data Corporation) Dan Kusnetzky, Director: "The Changing Mission of Computing"
- IBM から Andress Dieckow: "The Network Computer" と Paul Tomlinson: "IBM RISC System/6000 and AIX update"
- Microsoft から Brian Walcker: "Microsoft Windows NT Server" と David Webster: "Visual basic 5.0"
- 雑誌 DOC (Distributed Object Computing) の編集長 Marie A. Lenzi: "Beyond Basics - Objects and Distributed Object Computing"
- Classic System Solutions, Inc. CEO James Hobart: "Migrating Client/Server Systems to the Web"
- DEC から Gregg Laird, Manager, Worldwide Windows NT Technical Marketing: "A New Dimension In digital - Windows NT Computing Power"

3 日間の Breakfast (午前 7:00-9:00) と Break (午前 11:00-11:30) は、IBM、DEC、Sun、HP、Microsoft の各社がスポンサーとして提供した。IBM と DEC がそれぞれスポンサーとなった Breakfast では、IBM から Ron Gordon、DEC から Gary Gorden, Director of healthcare: "Healthcare and the Internet" のプレゼンテーションがあった。

セッション会場とは別の部屋で Solution Lab という名前で、InterSystems 社の Solution Providers (Atlas, Sea Change systems, ESI, Connex, TRAK Systems) のデモ、および、InterSystems 社の Object M のデモ、Caché (Open M) UNICODE/日本語ワールドテスト版のデモ、性能管理ツール Caché ViewPoint (Kaiser Permanente) のデモが行われていた。Solution Lab のすべてのハードウェア (40 台程の PC) は DEC がスポンサーとなって提供していた。

場所がフロリダであり参加者は半袖のラフな服装で参加していた。セッションの雰囲気もフランクで質疑応答も活発に交わされていた。どの講師もまず "Question is 何々" という形で質問内容の確認と質問が聞き取れなかった人達への配慮を行っていたことが印象的であった。

4 Post-Relational Database

今回のコンファレンスのひとつのキーワードは、Post-Relational Database でした。InterSystems 社はマネージメント・コンサルティング会社 Strategic Marketing 社と協力して会議で Richard Carrier, Chairman, Strategic Marketing: "The Emergence of Post-Relational Database" [2] の論文を配布した。同論文では、第3 (後述) のコンピューティング・パラダイム (Network-centric computing) へのシフトに伴い DBMS 技術における変化、すなわち、リレーショナル・データベースに代わる次のデータベース・システムが必要とされだして来ており、また登場し始めていると述べている。そして、その Post-Relational Database (PRDBMS) が対峙し解決すべき課題 (Challenges)、その特殊技術、PRDBMS システムの分類、および、post-relational 製品・ベンダーの選択指標について見解を述べている。

同論文には、M のデータベースについては全く触れられていないが、今後の PRDBMS のもつべき第一番目のキー指標 (network-based) performance & scalability "to support enterprise network access (GUI/client/server, Web-browser Internet/ Intranet etc.) transaction processing with complex application and data base requirements" は今後の M の目指すべき方向性がこのあたりにあることを示唆している。InterSystems 社は明らかにこの方向で WCTP 戦略を考えている。

以下、同論文の一部を要約する。

第1章 Post-Relational データベースの出現

Relational データベースは頂点に達した。もうすぐ Post-Relational データベースが前面に現れて来るだろう。RDBMS が衰えている理由は簡単で、現在開発者が直面しているもっとも重要な課題のいくつかは RDBMS が適合しないからだ。

RDBMS は簡単になくなるのか?

もちろん、"RDBMS の衰え" を正しい視点で捕らえなければならない。膨大な RDBMS ベースの既存アプリケーションが一夜にして置き換わるはずもないし、RDBMS ベンダーがすぐに衰える訳でもない。リレーショナル・データモデルの将来とは関わりなくデータのリレーショナルな見方 (View) は予見可

能な将来において残り続けるだろう。少なくとも、ほとんどの PRDBMS システムは SQL アクセスを提供し続けるだろう。

純粋 RDBMS 技術の販売の相当の落ち込みが明らかになるまでには何年もかかるだろう。なにせ、おそらく 10 年以上も前に RDBMS に置き換えられたと思われる RDBMS に先行した IMS や IDMS のようなメーカー特有の DBMS システムが、最も要件のきびしく、かつ重要な “ミッション・クリティカル” なアプリケーションで、実業務処理として今なお使われているくらいだ。

リレーショナル・データベースは次第に衰えて行くかもしれないが、2つの主要技術の利点は残るだろう。それは、**Standard** と **Portability** だ。今マーケットに参入を図っている発生期の PRDBMS システムでは、しばしばこれらの利点のひとつあるいは両方共をばぶいてしまっているが、明らかに、この2つは Post-relational の最前線で残り続けるだろう。コンピュータのエンドユーザは Standard と Portability を高く評価している。エンドユーザは Standard と Portability を、ベンダーの気まぐれやユーザ自身の組織の中での変動からシステム投資を保護するための第一の武器として捉えている。

VAR は Standard と Portability にさらに一層価値を置いている。それらは彼らの技術投資を保護してくれるばかりでなく、コストとリスクを下げながらマーケットを広げることから資すからだ。

RDBMS は役割をはたしたか？

リレーショナル・データベースの諛い文句は：

1. End-user Queries and Reports

レポート作成と検索の要件に、より敏速に答え得るという技術上の主張が RDBMS の普及に火をつけた。しかしその約束は達成されたか？

当初、SQL はエンドユーザによるレポート作成兼検索言語として促進されたが、今日では誰も SQL をエンドユーザ技術であるとまともには考えていない。SQL の限界もさることながらリレーショナル・モデルの限界のほうが大きな問題だ。複雑なデータ構造をリレーショナル・モデルで実装すると数十、数百にもおよぶテーブルが必要となるのが普通だ。こうした複雑な環境ではリレーショナル・モデルはプログラミングと性能の両方の問題で苦しむことになる。RDBMS ベンダーはデータクラスターリングなどのデータベース管理者が影で性能改善ができる様々な仕掛けや技術を提供しているが、大規模で複雑な実アプリケーションで高性能を出すという目標は未達成である。

VAR にとってはトランザクション・システムの性能問題はより深刻である。というのは、開発者は顧客サイトでのデータベースの管理についてコントロールできないからである。

2. Standard and portable

RDBMS は “portable standard” か？一般的に、普通信じられているより遥かに否である。大規模で複雑なシステムでは性能のために広範囲の stored procedure コーディングが強いられたり、ベンダー固有の技術を使っている。システム管理やチューニングの複雑さも真のポータビリティの妨げとなっている。

ポータビリティ上のわずかに成功したと言えるものは低性能の ODBC 結合で、レポート作成と検索に通常用いられる。それは、DBMS ベンダーではない、いわゆるサード・パーティのツールベンダーのほとんどが共通に利用している。それゆえ、ベンダー独立のデータベースを探している VAR やエンドユーザがアプリケーション完成時に通常悟るのは、ポータビリティとは捕らえどころのないものだ、ということだ。

Post-Relational へ導くトレンド

pure-relational から post-relational へデータベース技術をゆるやかであるが着実に移行させている底流を理解するために、データベースの要件とマーケットを基本的に動かしているコンピューティング・パラダイムをまず理解する必要がある。

データベース技術を動かしてきた基本的なコンピューティング・パラダイムが3つある。

technology-centric computing user-centric computing network-centric computing

現在、我々は第2の user-centric computing パラダイムから第3の network-centric computing パラダイムへ向かって移りつつある。数年前 Scott McNeall が洞察した “the network is the system” が今日の主要なコンピュータ・トレンド、つまり、明らかに network-centric computing である。client/server、Internet、World Wide Web および Intranet への爆発的な関心はこのパラダイム・シフトのあらわれである。

変化するパラダイムの底流

コンピューティング・パラダイムの変化は、技術的経済性 (technology economics) と、ビジネス要求の基本的な変化の結果である。

technology-centric computing パラダイムではハードウェアが高価で人件費は比較的安かった。したがってこの時期は技術生産性 (technology productivity) が強調された。

user-centric computing の第2のパラダイムはテクノロジー価格の急落で始まる。集積回路の発展やマイクロプロセッサの出現は主な計算コストを引き下げた。逆に人が相対的に貴重となり、人的生産性 (people productivity) が重要視される。最初、コンピュータは畏敬と恐ろしい雰囲気を漂わせていたが、コンピュータを恐れなくなると、より多くのサービス、特にレポート作成と検索を要求するようになった。IS 部門はエンドユーザのそれらの要求にすぐに追いつかなくなった。

リレーショナル・モデルと SQL はエンドユーザのレポート作成、検索要求を簡素化する技術として登場した。データ構造を行と列に単純化することで、簡単な非手続型言語 SQL の開発を促し、それがより強力な一般的なレポート機能の開発を助長し、ユーザの手に初めて検索パワーをもたらした。しかしながら、RDBMS 技術は高性能のトランザクション処理 (そこでは、色々な面から単純化したデータ構造はそぐわない) の必要性から生まれたものではない。

今日その勢いを急速に増している第3の network-centric computing パラダイムでは、企業生産性 (enterprise productivity) が今強調されている。統合がそのキーである。network-centric computing では、企業内および企業間でシステムを統合することによって、企業生産性が高められる。

第2章 データベース技術とコンピューティング・パラダイムの変化

DBMS 技術の盛衰はあれこれのコンピューティング・パラダイムの変化に呼応する。

technology-centric computing のためのネットワーク型・階層型 DBMS

創成期の DBMS システムは高価なハードウェア技術を最小にする必要があるという環境下で、有用な情報を提供することに注力していた。コンピュータ処理効率を最大にするということは、データモデルは利用形態と相似形にすることを意味する。したがって、この technology-centric computing の時期、階層型とネットワーク型のモデルが主流を占めた。

階層型とメーカー固有 (proprietary) 両方を同じ軽蔑

の目で振り返って見る傾向があるかも知れないが、それには同意しかねる。階層型データモデルが不都合であるというのは、何も確かなことではない。世の中の情報の多くは階層型で自然に発生しているということ、そして、リレーショナル・モデルの多くの性能問題は、階層型情報を正規化データ構造で処理しようと最適化を試みていることに直接起因していることはきわめて明らかである。

user-centric computing のための relational DBMS

リレーショナル・データベースは、standard、portability および レポート作成でのユーザパワー強化の利点を公開宣伝することで、その伸びが顕著となった。しかし、不幸なことに、この利点は低性能と高資源消費という RDBMS の明らかな欠点で相殺される。

リレーショナル・データベースの伸びは大きなビジネス要件の変化によっても促進された。グローバルな競争激化は user-centric computing パラダイムへの原動力となった。パワーを与えられたユーザは情報システムに多大な柔軟性を要求する。リレーショナル・モデルは柔軟性への突破口を約束している。第3正規形で貯えられているデータは、思いがけない特殊などんな要求に対しても柔軟に検索ができる。リレーショナル・データモデルは、自立的に動く情報強化された個人、あるいは小集団のユーザを支援するには理想である。

network-centric computing のための post-relational DBMS

競争激化とビジネス・リストラクチャは一時的な現象ではなく、むしろ現在ビジネスの常態である。生産性向上要求は個人の人々の生産性向上の総和では追いつかない。必要とされるものは、企業や企業間レベルの生産性を飛躍させるもの、高性能情報処理システムである。ビジネス競争優位は、情報システムの複雑さと守備範囲の広さで測られる。より多くのユーザがより多くのアプリケーション機能に繋がれば、よりビジネスの競争力が増す。今日 IT の焦点は、“どのように検索し、レポートを作れるか？” というのではなく、むしろ、“どのように、大量のトランザクションを処理するか？” である。

user-centric computing の必要性から生まれたリレーショナル・データベースは、network-centric computing と企業生産性が要求する職務には適しないということを我々はますます実感してきている。

第3章 Post-Relational データベース

PRDBMS が解決すべき難題 (special Challenges)

規模 (scale) の問題は、network-centric computing の最前部にある問題である。主に3つの次元の規模の問題がある。

- トランザクション量 — network-centric computing では、ユーザ数、トランザクション数は幾何級数的に増加する。明らかなビジネス・トレンドとして、システムの成功は、繋がったユーザ数に直接比例する。
- データサイズ
- アプリケーションの複雑さ

リレーショナル・データベースはますます増える network-centric computing 固有の問題を扱うために生み出されたものでは決してない。それは、大規模なユーザ数とトランザクション数をもつ大規模データベースへ拡張するには不都合である。レポート機能の柔軟性というリレーショナルの利点は、卓越した性能ととほうもない拡張性の要件と比較すると色褪せてしまう。

PRDBMS のための特殊技術

成功する PRDBMS は RDBMS のもつ顕著な3つの機能、つまり、standards、portability と flexibility

を保持し続けると信じている。post-relational の構成要素として際立つものが2つある。それは、Object Oriented(OO) 技術と Multi-Dimensional データベース(MDD)技術である。

• Object Oriented — 複雑なデータ構造と複雑なアプリケーション機能を扱う際、柔軟性と利用の容易さを提供するための突破口となる。

• Multi-Dimensional データ — MDD モデルの適切な実装は、実アプリケーションでの性能面での際立った突破口となりうる。

同論文では、さらに PRDBMS システムのカテゴリ、勝利する post-relational 製品とベンダーの見つけるための指標が述べられている。同論文を読むと InterSystems 社の WCTP 戦略がよく理解できる。

5 ソフトウェア開発責任者

昨年秋に InterSystems 社に Will Anderson 氏と Joe DeSantis 氏が入社し、両氏が同社のソフトウェア開発の責任者 (co-directors) に就任した。Anderson 氏は、19年間 DEC の開発部門で AI や RDB の上位アプリケーション開発ツール RALLY (ラリー) の開発などを行った後、2年間 Oracle 社で director of software development を経て InterSystems 社にいられた。DeSantis 氏は1986年に Object 技術のソフトツール会社 (Oakland Group) を共同設立、その後同社を Liant software に売却、そこで Object 技術の調査・開発 VP として働いていた経験の持ち主。

今後 InterSystems 社が、M をベースにした WCTP を目指すためには Oracle 社との競争は必至であり、その際 Anderson 氏の経歴とポジショニングは今後 M にとって有利に働くと思う。また、M への Object 技術の導入で DeSantis 氏の役割も大いに期待できる。M の世界へ他からこうした人材が入って来ているということを知って心強く感じた次第です。

6 おわりに

コンファレンスに参加して、M の今後の方向性とそれへの取り組みの息吹き・エネルギーを感じました。

リレーショナル・データベースの創始者であるエドガー・コッド博士自らが、ビジネスデータの解析処理を OLAP (On Line Analytical Processing) と命名した論文の中で、“リレーショナル・データベースは多次元解析として定義されているデータの合成・解析・整理統合のための強力な機能を提供しようとした訳では決してない” と述べている [3]。リレーショナル・データベースの神話が崩れつつある今、multi-dimensional な post-relational データベースとして高性能のトランザクション処理にフォーカスした M の戦略はひとつの回答であると思う。

文献

- [1] <http://www.intersys.com/wctp/infowww.html>
- [2] Richard Currier : The Emergence of Post-Relational Database --- Changes in DBMS Technologies Occurring in Response to a Shift to the Third Computing Paradigm, Strategic Marketing, April 1997
- [3] TM1WhitePaper, <http://www.applix.com/tm1/tmwhites/tm1wppr.htm>

北米MTA参加報告

上戸 隆¹⁾、北尾 真一²⁾

- 1) 住友電気工業株式会社 情報通信システム事業部 MEシステム部
- 2) 住友電工システムズ株式会社 応用システム事業部 ソフトウェア技術部

〒543 大阪市中央区南船場 4-11-28
TEL: 06-258-5519 FAX: 06-258-5520
kamido@apsys.ssy.sei.co.jp

1. はじめに

1997年5月18日から5月22日まで、Boston Hynes Convention Centerにて、第26回北米M学会に参加する機会を得たので報告する。全体としては、300名程度の参加者と推測され、去年と同程度の参加人数であった。日本からは、私共住友電気関係者2名、日本ダイナシステム1名、群馬ワークステーション1名の計4名が参加した。

2. 第26回大会のイベント

第26回大会は、大きく下記の7つのイベントに分けられる。

- 1) Solution Center
 - 2) 基調講演
 - 3) プロダクトデモンストレーション
 - 4) チュートリアル
 - 5) ホットトピックス・ディスカッション
 - 6) 施設見学
 - 7) データベース&クライアント/サーバ・ワールド・エキスポ
- 1) は展示会である。2) は、基調講演。3)、4)、5) に関しては、3会場で並列に行われたため、その全ては聴講できなかった。6) は、今回は、Brigham and Women's Hospital 見学である。7) に関しては、同じ建物で実施される100社が

展示する大規模な展示会である。M学会参加者は、このショーに対しても、無料で参加できる特典がある。以下、各イベントごとにその内容を述べる

3. 各イベントの概要

3. 1 Solution Center (出展社: 注1)

ベンダーの展示会および人材募集を実施する場所を Solution Center と呼び、大きな会場を提供していた。中央にコーヒー等のセットも用意され、参加者の交流の場にもなっている。ベンダーおよびソフトウェアハウス9社および人材募集4社の展示コーナーが用意されていた。展示はどちらかというところ、データベース&クライアント/サーバ・ワールド・エキスポに各ベンダーは力を入れていたが、M学会参加者にとっては、こちらの方が、じっくり相談できると思われた。

3. 2 基調講演

2000年問題に対する基調講演(The Year 2000: Are You Ready?) が、Coolidge System 社社長 Saulnier 氏により、2日目に行われた。この分野は、米国においても大きな問題になっているが、その割には着手が遅い。理由は、コストがかかることや関係者の認識が甘いことが指摘されていた。

3. 3 プロダクト デモンストレーション (項目: 注2)

各ベンダーが1時間程度で製品の紹介を行う場である。InterSystems 社、Micronetics 社に加えて、オブジェクト指向Mの ESI Technology 社、Case Tool の Atlas 社、SQLの KB Systems 社の発表を聴講した。大手2社は本大会でも発表があるので、それに譲るが、後半の3社もそれぞれの分野で、興味深い発表を実施していた。

3. 4 チュートリアル (項目: 注3)

有料のMに関するプログラミング等の講習会であり、クライアント/サーバ、WWW、GUI、HTML等の新技術への取り組みを中心としたテーマで行われていた。しかし、今回は参加していないので、ここでは演題だけの紹介にとどめたい。

3. 5 ホットトピックス・ディスカッション・セッション (項目: 注4)

WWW、Client/Server、オブジェクトオリエンテッド、GUI、2000年問題等の最新的话题を提供するセッションがあった。聴講した中で、印象的だったオブジェクト指向関連について述べる。

Javaのセッションが、Sun Microsystems 社からあった。概要の紹介であったが、聴講者の関心が高く、MがJavaに対して取り組もうとする意気込みが感じられた。これと同様に CORBA (分散オブジェクト指向) のセッションがあり、ESI Technology 社から紹介があった。

3. 6 施設見学

Brigham and Women's Hospital の見学が第3日目の午後行われた。学会会場よりバスで往復した。アプリケーションシステムの内容の説明が主体で、実際のハードが設置してある場所が見学できなかったのは残念であった。GUIを用いた新システムも開発中とのことであった。また、現在5,000台の端末をサポートしているが、新システムでは、ボストンの5病院を統合した、50,000台の端末が稼動するものとなるとのことである。

3. 7 データベース&クライアント/サーバ・ワールド・エキスポ

M関係のベンダー7社も展示を行っている。Hynes Convention Center の3フロアを占有する大きなショーであった。Mはこのなかで、Multi Dimensional Database で高速であるということテーマに展示を行っていた。InterSystems 社、Micronetics 社が20名程度の参加できるデモ会場を備えた展示を行い、参加者の目を引いていた。

3. 8 懇親会

新参加者に対する懇親会(第1日目の午後6時(CompuSearch 主催)) および、MTAの懇親会(第2日目の午後6時)があった。北米ばかりでなく、スイス、イギリス、オランダ等のヨーロッパの参加者も多く見られた。ヨーロッパは、会計システムのアプリケーションにM言語を使用していることが多いことを再認識した。アジアからは、韓国から4名参加していた。

4. おわりに

今回、北米MTA学会に参加してみて、特に感じたのは、Mが新しい世界(WWW、オブジェクト指向)を取り入れようとする意気込みであった。また、基調講演である2000年問題に対する関心も、北米では高い。その反面、実施に関しては、日本と同じように遅いことが危惧されていた。M言語の高速な検索・優れた文字列処理・コンパクトなデータベースという従来の特徴を活かしつつ、新たな技術との親和性をもった、新しいM言語に期待したい。

※参考 HOMEPAGE

1. M Technology Association

<http://www.mtechnology.org/>

2. 第26回北米M学会関連

<http://www.mtechnology.org/conference97/index.html>

<http://www.mtechnology.org/newsletter/index.html>

注1) Solution Center 出展者

1. ベンダ

Atlas Development Corp.
AudioCARE Systems
CyberTools, Inc.
ESI Technology Corp.
George James Software, Ltd.
InterSystems Corp.
KB Systems, Inc.
Micronetics Design Corp.
Sea Change Systems, Inc.

2. リクルータ

Henry Elliott & Company
IDX Systems Corp.
Partners HealthCare System, Inc.
Systems Resource Corp.

注2) Product Demonstrations 項目

1. IDX Systems Corp.: IDXtend
2. Micronetics Design Corp.: Focus on Client/Server: Developing and Distributing M Applications

3. Atlas Development Corp.: Freedom
4. Micronetics Design Corp.: Focus on the Web: For the M Developer: Introducing the Microneitics Web-centric Database Solution
5. InterSystems Corp.: How Do You Web-Enable My Application?
6. Atlas Development Corp.: Web Freedom
7. KB Systems, Inc.: Adding SQL to M: What It Takes and What You Get
8. CyberTools, Inc.: CyberTools (R) and M: Build-It-Once for Windows, Java, Web, and Character Terminals
9. KB Systems, Inc.: Transforming M into an "Open" System
10. CyberTools, Inc.: CyberTools' FileMan GUI Client/Server/Web Solution!
11. Sentient Systems, Inc.: Web Development Tools
12. ESI Technology Corp.: EsiObjects™
13. InterSystems Corp.: What Database Offers World-Class Transaction Processing?

注3) チュートリアル項目

1. Effective Web Pages
2. Database Design
3. Delivering Value Through Information Systems
4. Introduction to Information Modeling
5. KB_SQL (Intermediate SQL)
6. Introduction to HyperText Markup Language (HTML)
7. Guidelines for Choosing Web Technology: Accessing Your Existing M Data
8. Event-Driven Programming for GUIs
9. Java and M: The Similarities, Differences, and Opportunity for Synergism
10. VA FileMan for Programmers
11. Using M to Develop Applications for the World Wide Web
12. Developing GUI Applications
13. Object-Oriented M Programming
14. Guidelines for Choosing Web Technology: The Role of the Database
15. Advanced M Programming with Style and Discipline
16. Features of the New M Standard
17. Introduction to Web Site Analysis, Design, and Management
18. Introduction to M Programming

19. Web Site/Intranet Management
20. Technology Insight: Open M/WebLink
21. Introduction to Delphi
22. Intermediate M Programming
23. Developing and Distributing M Applications with MSM-Workstation
24. Multiuser and Multidevice M Training Techniques
25. Looking Ahead to the "Millennium M Standard"
26. Transaction Processing

注4) ネット・トピッカ・ディスカッション・セッション項目

1. Building World Class Web Applications
2. Applications Conformance Clauses
3. Year 2000: Will Your Software Stand the Test of Time?
4. Specifications: From Business to Systems
5. MDC (MUMPS Development Committee) Update
6. How to Improve Your Marketability and Value within the Information Technology Industry
7. NextGen and Object Technology
8. VISTA/DHCP'S Client/Server Environment
9. Answering the Question "Why M?"
10. Sun on Java
11. FORMAT: How the World Looks at Money
12. M Behind the Scenes: Evolving M Legacy Systems to Support Modern Industry Standards
13. Merging Disparate IS Systems
14. Web Scripting in M
15. The Year 2000 Bug: It Could Be a Bigger Problem Than You Think!
16. GUI vs. Browsers
17. Implementing a Three-Tier Architecture
18. VA FileMan Update
19. Focus on the Web: Setting Up a Web Environment
20. Report Generator Shootout II
21. Year 2000: Will Your Software Stand the Test of Time?
22. Developing for Visual Interface Independence

23. Fast Time to Market: Using Visual Basic and NT to Develop and Deploy a Ground-breaking Hospital Information System in Six Months
24. Software Tools Review
25. Brigham & Women's Lecture/Demo
26. Accessing the M Database with OOPS
27. Telephony and Partners Healthcare
28. Migrating M Databases to the World Wide Web
29. Getting to World Class by Making the Move to Open M NextGen
30. Opening VISTA/DHCP for Access with Commercial SQL Products
31. Year 2000: Information System Problems and Solutions
32. M at the Crossroads
33. What Is the World Wide Web and Why Should I Care?
34. A Vision for the Future of M: "M Technology Can and Will Survive"
35. Construction of the Hardhats.org Web Site
36. GUI Shootout III
37. Apps of Steel: How to Create Bug-Free Systems
38. Linking to VISTA/DHCP with non-Delphi Client/Applications
39. Integrating M and Web Technologies
40. A Trio of Teletraining Technologies
41. Front-Ending M with the WWW: An Application Perspective
42. OOPS and M: Internetting to Score Your Goal
43. Focus on Client/Server: Creating Client/Server Applications with Popular Tools
44. The Secret of Learning M, Gaining a Major Edge in Job Effectiveness
45. CORBA, Objects, and Your M Database
46. Interfacing M to the Web: Demonstration of Live Intranet Applications
47. What Makes Good M Code?
48. The Year 2000 Bomb: Issues and Non-Issues

M言語による構文解析における英文のスケルトン構造

～ 構文解析におけるMソリューション ～

高橋 亘

関西福祉科学大学

〒582 大阪府柏原市旭ヶ丘3-1-1

TEL 0729-78-0088 (内 781) FAX 0729-78-0377 E-mail: Wataru_Takahasi@msn.com

Abstract: Some methods of pattern matching in parsing algorithms of English are examined. Special pattern image of English sentence called skeleton structure is proposed. Adopting this pattern image parsing algorithm become very simple.

Keywords: MUMPS, parsing method, pattern matching, skeleton structure of sentence

1. はじめに

1970年に、コンピュータによって英語の構文を解析する世界で初めての試み(ATG; 1970)がなされてより今日に至るまで、四半世紀を越える歳月が流れている。しかしながら、近年市販されている英和翻訳ソフトなるものを試用してみるにつけ、英語構文の認識のあまみや和訳された日本語の語彙の貧困さが目立つものがあまりに多く、翻訳家の手助けになるようなものはあまりに少ない。

25年もの遅れを感じながらもあえて英構文解析のアルゴリズムを問い返してみようという意欲を筆者に抱かせたものは、このような不完全な市販翻訳ソフトの氾濫である。

インターネットで世界のコンピュータが繋がれた今日、世界の情報交流を阻んでいるものは言語の差異そのものである。したがって、安価で有能な翻訳ソフトが囑望されていることは論を待たない。しかしながら不完全な翻訳ソフトの氾濫はかえって文化の交流に弊害を与えこそすれ世界市民を利することは何もないであろう。

2. 構文解析の新しい試み

1970年代から1980年代になされていた自然言語解析に関する多くの仕事(DCG; 1978-1984)では、LISPやPLOGと言った論理型プログラミング言語が主として使われた。これらは構文木を、その各ノードに対応した構文規則をもとに、根から葉の方向に、言わばトップダウン方式によって、論理的に組み上げようとするものであった。

論理型推論を用いて、出来得る限り少ない構文規則によって多様な英語の構文を組み立てようとするのは、数学的な推論の美しさからすると十分魅力的なものであるが、このような研究が進むにつれ、論理的推論とヒトの脳が行っている構文判断の仕方との食い違いが次第に明確になってきた。

ヒトの脳がやっている構文認識は論理的というよりは直感的で断定的なものである。素早く、誤ることのない脳の構文認識の方法をコンピュータの推論の方式として表現できないものだろうかと言いつつも考える。

この論文では、英語構文を解析するにあたって、ヒトの脳がどのように文の特徴を捉えるのだろうかと言いつつも整理し、これを類型化する

ことによって、コンピュータの推論のモデルを確立することを試みたい。

3. 開発言語について

ヒトの脳の特質の一つは、幼児期からの経験により膨大な構文規則を大脳皮相に蓄えているということである。これを模倣するには膨大な構文規則をデータベース化し、構文規則の各段階で経験則を生かすような方式を取ることが必要であるが、膨大なデータを扱うことは1970年代のコンピュータでは難しかったと考えられる。年月を経て、コンピュータの性能の進化した今日では、このような方式が技術的に可能になってきた。このような時期に、ハードディスク上の膨大なデータを効率的に活用することの出来るプログラミング言語が囑望されるのである。

言語のデータは不揃いなものが多い。これをランダムアクセス出来る形でハードディスクに蓄えておき、必要とあればデータに文字列処理を施してパターン認識に資する。このようなプロセスを経済的、かつ自然な形で行うことの出来る言語はMUMPSである。MUMPSの効用については十分議論されているので、ここでは詳説は省略する。本質的なアルゴリズムは、本来それを表現するプログラミング言語に依らないはずであるが、これを最も自然な形で表現するのに最適の言語が存在することも事実である。

4. 構文規則の分類と獲得

次に考えられるヒトの構文判断の特質は、その判断能力、言い換えれば判断に必要な構文規則の体系、を学習により獲得しているという点である。ヒトは学習により、幼児期のいわゆる赤ちゃん言葉から次第に複雑な構文へと、構文規則を修得していく。この論文ではヒトの言語習得過程に重点を置いたアルゴリズムの構築を図るため、英語の構文規則を分類するに当た

って、英文そのものを易しい英文から、難解な英文に並べ直し、易しい英文によって構文規則のカテゴリーを決める。そして可能な限り既成のカテゴリーの部分的拡張によって、より複雑な英文の構造をとらえるように配慮する。そしてこのような拡張に困難が生じたとき、新しい構文のカテゴリーを用意するという方式を取ることにする。

日本の英語教育は中学に始まるので、本来中学英語の教育過程は学生の知的発達段階を熟考して組まれているべきであるが、現在の中学の教科書にこの配慮が十分なされているか、という点については是々非々の議論がある。しかしこのような議論は英語学者もしくは英語教育の人にまかせることにして、たとえ不完全であるにせよ、筆者の手元にあるテキストとしては中学の英語の教科書しか無いので、とりあえず東京書籍の教科書「NEW HORIZON English Course」から例文を取り、まずは1学年のレベルを解析出来ることを目標にモデル化を進めることにする。

5. 構文判断とパターン認識

今ひとつ取り上げたいヒトの構文判断の特質は、その判断の仕方にある種のパターン認識によっているという点である。ヒトの脳は、目を通して文を文頭から文尾へ順に読み込むがこのプロセスの中で文の各部分の詳細についての情報を読みとると同時に、文の全体の構造への漠然とした類型イメージを描いていく、そしてもしこの類型イメージとあわないような点が見つかったら、もう一度読み返して文の詳細の認識と類型イメージとの食い違いを検証し、これらのいずれかを修正し正しい認識へとたどり着く。

このような認識過程は短文ではほとんど気づかないが、少し複雑な構造を持つ文章を読むときに経験することである。そして、この漠然とした類型イメージこそ、幼児期からの学習を

通して獲得した構文規則のカテゴリーに他ならないのである。

6. スケルトン構造

中学の第一学年の英語の教科書の例文をもとに、ヒトが持つ漠然とした構文の類型イメージをコンピュータの言葉で表現してみようというのがこの論文の目的である。

この論文では例文より英文の詳細構造をある種の規則により捨象した構文パターンを抽出する。この構文パターンは、先に述べたヒト

の脳が英文に対して描く類型イメージに対応するコンピュータの上での表現である。

数学のグラフ理論においては、グラフの詳細をある意味で捨象した図形をスケルトンと呼んでいるが、グラフ理論に習って我々の構文イメージを「英文のスケルトン構造」と呼ぶことにしたい。詳説は略すことにするが、スケルトン構造を中継することにより、次の例のような直接的構文判断が可能となる。

文	スケルトン	構文
(1) Come in.	→ vp.	→ vp+.
(2) Sit down.	→ vp.	→ vp+.
(3) Is she from America?	→ vcp#\$\$?	→ vcp+np+prep+?
(4) She's from Canada.	→ #vcp\$.	→ np+vcp+prep+.
(5) I'm from America.	→ #vcp\$.	→ np+vcp+prep+.
(6) I'm in the seventh grade.	→ #vcp\$.	→ np+vcp+prep+.
(7) Is soccer popular in Japan, too?	→ vcp#&\$,adv?	→ vcp+np+adj+prep+,+adv+?
(8) How about you?	→ how\$?	→ how+prep+?
(9) What time do you go to bed?	→ what#vd#vp\$?	→ what+np+vdo+np+vp+prep+?
(10) About ten thirty.	→ #.	→ np+.
(11) It's about noon on Sunday.	→ #vcp#\$.	→ np+vcp+np+prep+.
(12) The dog is under the table.	→ #vcp\$.	→ np+vcp+prep+.
(13) Look at this picture.	→ vp\$.	→ vp+prep+.
(14) I listen to music.	→ #vp\$.	→ np+vp+prep+.
(15) I go by bus.	→ #vp\$.	→ np+vp+prep+.

7. 考察

英文の構文認識のプロセスの中間段階に、英文のスケルトン構造と称するイメージパターンを用意することで、構文解析の効率を高める方法を考察してきた。今までの議論を構文解析のアルゴリズムとして要約するなら、次のようなことになる。

- 構文解析の第一段階では、与えられた英文より英文のスケルトン構造を解析する。
- 次の段階で、このスケルトン構造に符合する、構文木の上位構造に対応する構文規則を見つける（この段階では前置詞句という中間段階の構文認識が残っている）。

このような2段階の方法により構文の概形を決定した後、最終段階として動詞の種類と句

の並びから前置詞句の文法的役割を特定する。

我々の方法では英文のスケルトン構造を活用することにより構文認識の各プロセスが直裁的になり、そのため高速な構文決定が期待される。

8. 結論

我々はこのようなアルゴリズムを採用した構文解析のプログラムをMUMPSというプログラミング言語により実現した。しかし、この論文での議論はMUMPSの特性に依らない形で表現しているため、同様のプログラムをその他のプログラミング言語によって構成する事も可能であると思われる。

今回の議論では中学一年程度の英語に限定したため、関係代名詞や、不定詞、分詞構文等に関連したアルゴリズムは除外されている。これらのより高度な構文判断の議論は、現時点では時間的制約のため将来の問題として残さざるを得ない。しかしながら、英文のスケルトン構造というものと想定し、これを構文認識に活用する方式は、より高級な英文の解析にも有効であることが十分に期待できよう。

決断過程の構造化分析

～ マネージメント支援システムの新しいありかたへ向けての考察 ～

廣瀬 康行

東京医科歯科大学歯学部歯科麻酔学講座, 同歯学部附属病院情報処理委員会企画調整部会長

〒113 東京都文京区湯島1-5-45

Voice: 03-5803-5551 Fax: 03-5803-0206 E-mail: y.hirose.dane@dent.tmd.ac.jp

1. DSSの経緯と限界

1.1. 歴史とドグマ

意思決定 (decision making) は、現状から目標状況に至る経路の発見的探索として捉え、経路上の分岐点をノードとし、ノードから他のノードへの状況変化は作用素 (または規則) によって行うものとして特徴づけられた。そしてこれは、人工知能研究や意思決定支援システム (DSS) の基礎となった。

これは「限定的合理性」のプロセスであり、ノードにおける経路選択は、形式的な・しかも限定された規則を、極めて狭視野の部分情報に適用することによって行われている。

この手法について良く知られている限界はノード展開数の制約すなわち「組合せ爆発」である。しかしそんなこと以前に、もっと重大な欠陥もしくは現実との乖離が含まれているのである。

1.2. 疑義

上記の妥当性への疑義を以下に列挙する；

- ・選択肢集合はどのように決めるのか、そしてそれは限定的なのか。
- ・選択肢の優先順位は (どう生成するか) ?
- ・そもそも選択肢を生成する規則それ自体が、真に限定的か。いわゆる意思決定ではなく「決断」は、新しい選択肢を創り出すことに向けられる場合もありうる。

実は、選択肢から選択するという形式のみで問題空間を定式化すること自体、すでにバイアス要因となっていないか。

逆に言えば、選択肢空間が全ての合理的可能性 (そして実際の有用性) を尽くしていると考えすることは重大な誤りではないのか。

- ・状況への被投性: ヒトという存在は時間等の制約の中で決断を迫られているということである。

すなわち、全てのノードの展開 (当然ながら各ノード選択肢集合展開を含む)、ならびに選択肢選択の詳細検討の余裕はあるのか。

ここでは「組合せ爆発」以上のことをいっている。つまり、ヒトは「合理的な選択」ではなく、「決断」を行っているのではないか。

- ・背景 (文脈) の重要性。被投性による選択限界性のみならず、背景と文脈への顧慮が欠如していないか。

そもそもある行動の選択や新しい選択肢の生成には、それに先立つ背景や文脈により制約されている。とすると与えられた文脈に関連した選択肢集合を形成することは本当に為しうるのか。

- ・マネージメントでは、組織内や自分自身との調整と・それに付随する会話を含んでおり、これによって問題や目標が「浮かび上がる」。すなわち背景や文脈の中で形成されるわけだが、これが対自もしくは対他の言語活動を

基盤としている以上、「会話構造」と解釈の問題を避けて通ることはできない。

- ・真に難しいのは決断過程における「問題解決空間」の定式化そのものである。

伝統的な手法ではいわゆる「問題空間」が既に明確であることが前提とされ、仮定されている。しかしそれが「見えている」のは誰か。プログラマーなのであろうか。

そして先に挙げた疑義と関連して、対象とすべき定式化空間は閉じているのか、あるいは閉空間としてよいのか。

1.3. 構造構築のパラダイム

つまり DSS では、解決すべき問題 (=DDS でいうところの目標) ならびに関連する状況や選択肢集合すなわち「問題空間」全体は完全に「見えて」おり、その意味で実は「決断後」の作業なのである。しかし実世界においては、ビジネスシステムでも医療システムでも、真に困難かつ支援の必要な状況とは「決断」の過程である。

そして DSS での問題空間は、その下敷きにコンピュータ処理指向があることによって完全な構造的タスクを求めており、よって閉じた世界として構成されている。

一方、実世界における決断過程を重視する・ここで謂うところの「問題解決空間」は半構造的タスクを目標とするものである。したがって規則 (作用素) を常に完全に規定できるわけではない。しかし DSS のパラダイムに起因して、DDS 構造内に本質的に内在する盲目性からは解き放たれているとともに、システムがヒトの決断を支援しうる領域なのである。

2. 目標

まずは、問題の生成から目標の設定を経て、解の策定に至る思考のプロセスを定式化した。この枠組みにより、時間と文脈、思考ステップ

間の論理関係、決断状況の記述、外界と内的世界とを統一的に扱うことのできる「問題解決空間」を定式化した。

- 思考や決断の過程を、種々の文脈ならびに状況情報とともに統合した形式によってシステム内に記録蓄積すること
- システム内におけるマネージャの挙動に、マネージャ自らが・その合理性と妥当性を明示的に付与しうる環境を整備すること

3. 方法

実世界での問題の発見と定義から、目標設定と解の策定までの過程について、内省と共同研究者間の議論によって、以下のことを行った。

- 要素構造への分離分割
- 構造間の論理関係の規定
- 時間推移における外界の変化に応じた内的状態の変遷関係の規定

4. 結果

4.1. 問題解決空間の概要とその性質

問題の生成と、それに引き続く目標の設定ならびに解法の策定に至る思考、すなわち内的な運動過程あるいは知の挙動の範囲を、ひとつにまとめて問題解決空間と呼ぶことにする。この思考のプロセス、そしてそのサブ空間は a priori に存在しているのではなく、知の挙動によって徐々に、あるいは直截的に形成されるのである。

三個のサブ空間では、絶えず対象状況への参照運動が生じている。この参照結果に如何によっては、それぞれの空間の大きさや内包する要素も変動することになる。換言すれば、個々の空間の形成過程・あるいは・一つの空間から一つの空間への写像、そしてそれらの要素は、いずれもが対象状況からの修飾を受けることになる。

さらに時間経過による外界状態の変化によって、一つの空間から一つの空間に対して・あるいは・その空間の再構成過程に対して、言わば逆順の修飾作用も存在している。この修飾作用によって、個々のサブ空間内部の部分構造も変遷を余儀なくされることとなる。

4.2. 思考プロセスの分解と相互関係

まず、各空間の生成過程、ならびに空間から空間への写像過程を、複数の思考ステップに分解した。またこれを思考ブロックとして、明示的に表現した。同時に、各思考ブロックもしくはブロック間における知の挙動範囲や、ブロック間の参照関係や論理関係を規定した。これらの知の働きや相互関係は、そのまま思考のプロセスと捉えることができる。

なお紙幅の関係上、この概要については学会大会にて説明せざるをえない。また詳細については、稿を改めて記したい。

4.3. サイクルと巡回性の排除

次に、時間経過による逆順の修飾作用を扱いやすくするために、逆順の作用を形式的に排除することとした。すなわち、ある時刻点での思考過程を一つの思考サイクルとみなして、時間経過にしたがってサイクルを回転させることとした。このサイクルは、マネージメントサイクルと呼ぶことにする。このことによって、逆順の修飾作用は、時刻 t のマネージメントサイクルから時刻 $t+1$ のマネージメントサイクルの要素への修飾と捉えられ、巡回しないプロセスとして扱うことができるようになる。

なお思考の焦点は、時間経過によるマネージメントサイクルの回転の中で、思考空間と対象空間との間をスパイラルに動いていくこととなる。

4.4. 時間による変遷の明示

さらに、時間経過におけるマネージメントサイクル間、あるいはマネージメントサイクル内

の対応する思考ブロック間での変遷関係や相互関係の希薄さを惹起しないために、二点を考慮している。

一つは、マネージメントサイクル自体が、明示的にはプロブレムリストの変遷(非変遷を含む)・もしくは・ゴールを中心軸として回転していることである。そしてもう一つは、マネージメントサイクル間での対応するブロックの変遷関係を明示し、規定していることである。なお変遷関係には、順位や優先度の変化も含まれている。

上記三小節で述べたことは定式化のための処理ではあるが、問題解決空間における知の挙動を、技巧的な要因によって変質させてはいない。むしろ、1) ある時刻から次の時刻へと時間が経過した際には、外界ばかりでなく内的世界も同一ではないこと、2) そうは言っても過去の歴や文脈を無視しえないこと、の二点を考慮すれば、かえって自然な形であると言えよう。

5. 考察

5.1. 決断過程の重要性と現実性

これまで論理性を付与するための構造化という側面を強調してきた。しかし我々は、一方では、文脈のある程度の論理的な構成と、いわゆる科学的な合理性とを峻別していることを明言しておきたい。

現実世界をふりかえれば日常において、いわゆる「意思決定的手法」が採用されることなど、コンピュータの支援があろうがなかろうが、まずほとんどありえない。現実世界においてはむしろ、問題空間も目標空間も解空間も十分に形成しない(できない)うちにも、何かしらの対応や処置を為しつつ、自らと自らをとりまく状況とを未来へ投げかけているのである。

換言すれば実世界では「意思決定的手法」は採用しておらず、不完全な問題解決空間の形成

のままに「決断」を行っているのである。価値観や倫理的な議論は別にして、これこそが現実であろう。

ということは決断過程の記録支援環境の方が、現実的には、より重要性な意味を持つものと思われる。定式化した問題解決空間はフルセットで応用することを前提として構築したもの、サブセットで応用することを拒むものではない。むしろ決断というキーワードを軸としてサブセットの応用を考慮することのほうが、より現実的かつ応用範囲を広くすることになるであろうと考えている。

5.2. Heuristic な方法への対応

現実性という側面でもう一つ挙げておかなければならないことは、良くも悪くも heuristic な方法が必要である、ということである。

言い換えれば、限定的かつ狭小な問題空間の仮定によって、対象空間へのアクション数とともに、思考空間におけるプロセス数をも極端に減少させることで、マネージメントサイクルの効率的な回転をもくろむ、ということである。

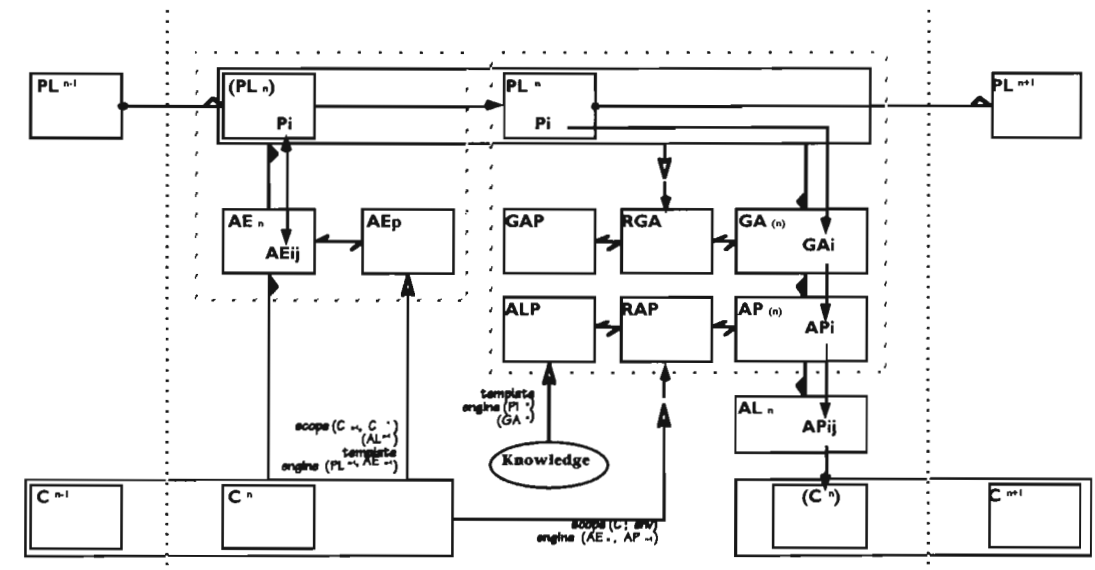
Heuristic な手法は効率的である反面、誤謬や pitfall の危険性ははらむものである。しかし最適化された heuristic な手法は、その限界

と危険性さえ認識していれば極めて有用であることに反論は無かろう。

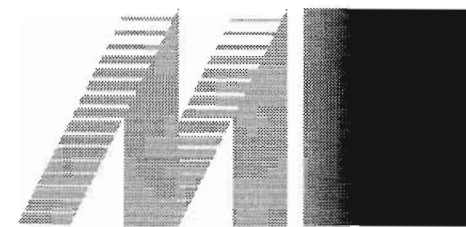
では如何にして最適な手法を見出していくのか。その環境も、定式化した問題解決空間としてその実装と応用とによって、近い将来には提供できるものと思われる。

文献ならびに資料

- 1) 問題指向電子診療録を構築するために必要な中核構造と操作環境の研究開発：大山喬史，廣瀬康行，佐々木好幸，木下淳博，水口俊介，藤江昭．平成7年度 東京医科歯科大学 民間等との共同研究．
- 2) Problem Oriented Medical RecordBase の構造ならびにハンドリングに関する一考察：廣瀬康行，佐々木好幸，木下淳博，水口俊介．第15回医療情報学連合大会論文集 569-570. 1995.
- 3) 診療論理情報ベース：廣瀬康行，佐々木好幸，木下淳博，水口俊介．平成7年度国立大学医療情報部門連絡会議 65-68. 1996.
- 4) 問題解決空間の定式化に関する考察：廣瀬康行，佐々木好幸，木下淳博，水口俊介．医療情報学 17 (3) Suppl : 185-192. 1997.
- 5) 思考過程に視座をおいた病歴記録形式におけるプロブレム変遷記述言語に必要な述語群に関する試考：廣瀬康行．Seagaia Meeting '97 HandOut . 1997.



広告



● 高性能な新世代データベース

● 高速アプリケーション開発環境

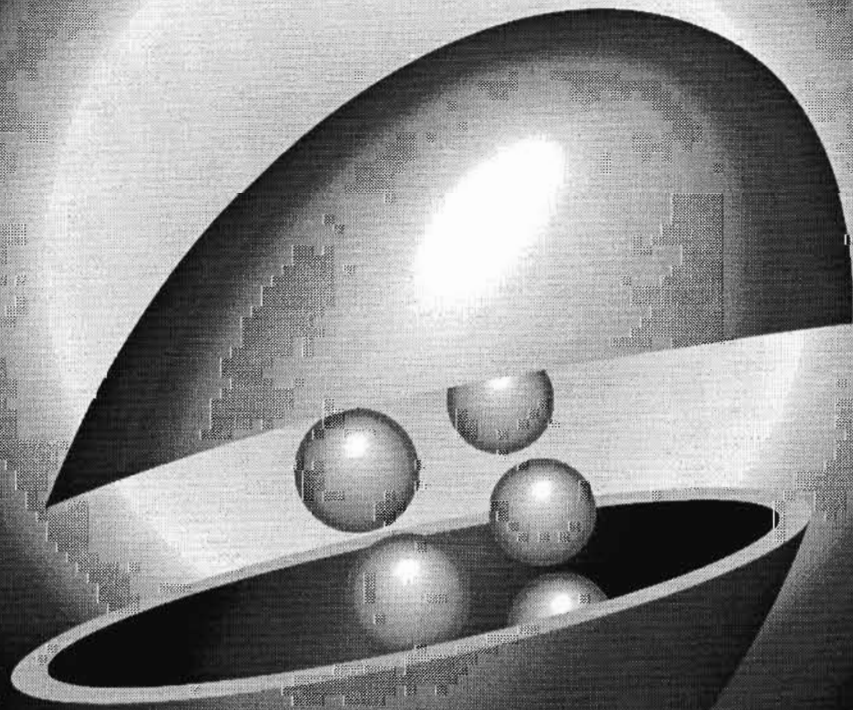
● リアルタイムトランザクション処理環境

Caché

Cachéとは、高性能な新世代データベース（ポストリレーショナルデータベース）＋高速アプリケーション開発を可能とする統合システム大規模データベース処理やリアルタイムトランザクション処理に最適！

機能的特長

- 高速で強力な処理能力
 - JIS / ISO標準規格のM言語システムをフルサポートしており、例外の多い複雑なデータや処理に容易に適合
 - 分散キャッシュプロトコルによってネットワーク上の分散データベースへの高速アクセスが可能
 - 数千レコードから数百万レコードのデータベースも応答時間はほとんど不変
 - 高速なネイティブ32ビット処理
- 拡張性
 - PC1台から数千台のクライアント / サーバシステムまでサポート可能
 - システム構成を拡張・変更してもアプリケーションは修正不要
 - サードパーティのアプリケーションからSQL, ODBC, OCI経由で、データベースへアクセスするリレーショナルクライアント / サーバ機能
 - 様々な開発ツール (Delphi, Visual Basic, Visual C++, PowerBuilder等) のグラフィカルユーザーインターフェイス開発環境からデータベースへ高速アクセスするOCX / ActiveXコントロール
 - ブラウザからインターネット、イントラネットを通して、データベースへアクセスする通信インターフェイス
- 安全性
 - ジャーナリング / ミラーリング / シェアードリソースなどの障害時対策
 - 分散トランザクション処理も装備
- 快適に短期開発
 - 使いやすく豊富なツール、ポイント&クリックを特徴とする強力なアプリケーション開発環境
 - 従来のRDBMSのような、工数のかかる複雑なチューニングは不要
- 容易な保守
 - データ項目の追加 / 挿入 / 削除も容易で、ユーザーの要求変化に柔軟な対応が可能



事例紹介

- 国内では、数百台のクライアントからなる総合病院情報システムや、レコード数数千万件の物流システムがPCのみで稼働中
- 米国では、総クライアント5万台という巨大なオンラインリアルタイムトランザクションシステムを構築するプロジェクトが進行中

Cachéの動作するオペレーティングシステム

- Microsoft Windows NT
- Microsoft Windows 95
- 各種UNIX

Japan DynaSystems Inc.

日本ダイナシステム株式会社

〒460 名古屋市中区新栄2-1-9 雲竜ビル東館5F
TEL 052-242-5441
FAX 052-242-5984
URL <http://www.jdynasys.com/>

日本ダイナシステム株式会社はCachéおよびM言語専門のソフトウェア会社として、米国InterSystems社 (URL <http://www.intersys.com/>) と協力 / 提携し、日本国内での販売 / 技術サポートに専心しています。

Cachéは、米国InterSystems Corp.の登録商標です。その他、記載された社名および製品名は、一般に各社の商標または登録商標です。

digital



Lサイズには、Mがいい。

大容量のデータも、すばやく快適レスポンス。
DECのMは、着心地のいいデータベースです。

データベースは、いつも成長期。データが大きく育った日に、満足なパフォーマンスが得られないようでは困ります。DECのMは、たとえ大規模なデータベースに成長しても、高速のレスポンスが自慢。使い心地で差のつくデータベースです。クライアント / サーバ環境に適し、プラットフォームにも依存しません。その上メンテナンスも簡単で、保守でお悩みの方には朗報です。あまり知られてはいませんが、世界中で約100万人、日本でも約10万人という数多くのご利用実績にも、そのアドバンテージは表れています。DECは、豊富な経験や実績を活かし、さらにMの専門会社InterSystems社との技術提携 / 販売提携を行いながら、今後とも強力なサポートを提供します。どうぞ安心しておまかせください。

データベースの
隠れた
ベストセラー

M

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

お問い合わせは、統合システム本部 ☎0120-499150

Alpha AXP CALIS

《キャリス》

高機能/広域ネットワーク対応の情報図書館システム

CALISは、情報文化資源の共有・共同利用をシステムの基本理念に、徹底した“標準志向”を重視しながら丸善が、開発したオリジナルな広域ネットワーク対応の情報図書館システムです。

CALISは、高機能のネットワーク・プラットフォーム上に、①データベース構築機能②高速情報検索機能③相互貸借機能④電子メール機能⑤複数部署から複数業務の同時処理

機能⑥データベースの相互検索機能などを有機的に結合し、統合したシステムを実現します。

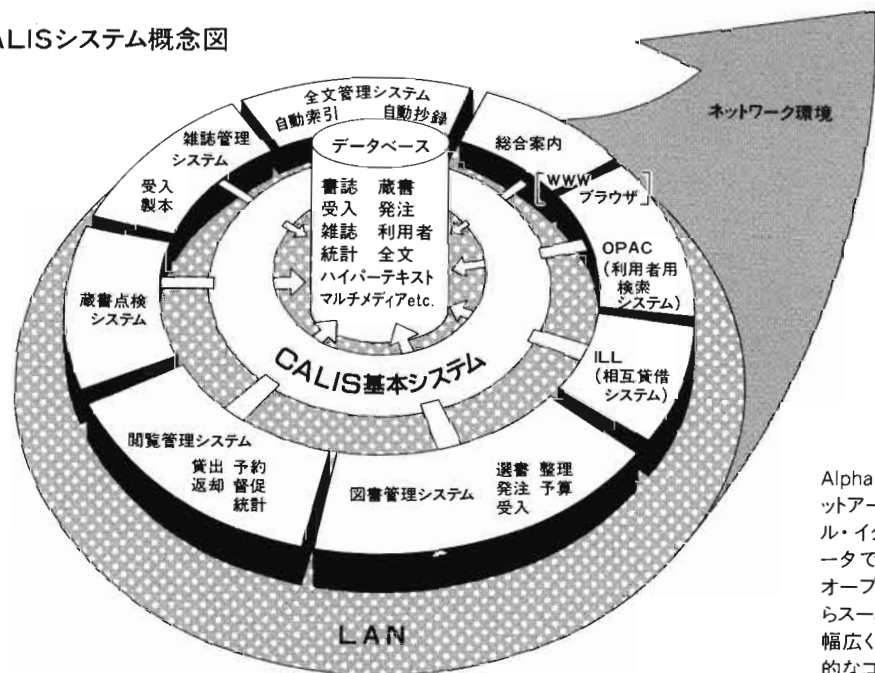
また、マルチメディアなど最新の情報媒体のデジタル化も標準で対応し、電子図書館を可能にします。

CALISは、数多くの大学図書館、公共図書館、企業・専門図書館など情報文化施設に広く採用され、高い評価を得ております。



- 標準のデータベース環境:MUMPS(JIS X3011)
- 標準のネットワーク環境:TCP/IP, Z39.50, DECnet
- 高度なNACSIS(学術情報センター)接続機能:VTSS, TCP/IP
- 容易なシステム管理機能:データベース集中、業務分散
- 高信頼性を実現するセキュリティ管理、及びデータベース保全機能: DOD C2 レベル
- 柔軟なデータベース構造:あらゆるMARCに対応
- 業務に密着した図書管理、雑誌管理などの業務支援機能
- 先進の情報発信機能:Internetに対応
- オンライン書籍受発注システムとのリンク:ちよいす君

CALISシステム概念図



Alpha AXPは、新時代を画する64ビットアーキテクチャーの日本デジタル・イクイップメント社製のコンピュータです。オープンアーキテクチャとして、PCからスーパーコンピュータにいたるまで、幅広くサポートし、長期にわたり理想的なコンピューティング環境を実現します。

情報図書館システム営業部

〒103 東京都中央区日本橋3-9-2 第二丸善ビル
 ☎(03)3272-0336 FAX(03)3271-8680
 E-mail: calis@maruzen.co.jp
 http://www.maruzen.co.jp/ils/ils-index-j.html

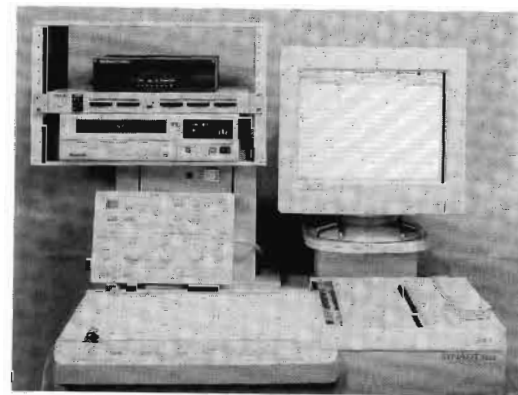


マルチメディアのNEC

NEC

近未来型脳波検査システム デジタル脳波計

EEGネットワークシステムやてんかんモニター等に於けるビデオ同期収録、ダイポール解析、マッピング解析などさらに医療現場のニーズにお応えします。



ビデオモニタリングユニット

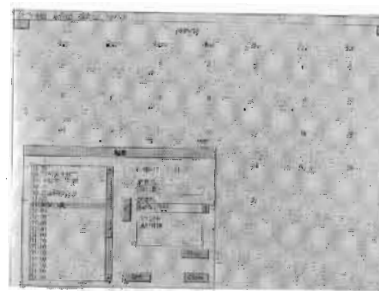
●VTRに収録する映像データと脳波計に収録される脳波データが同期されて収録されます。その後の解析に威力を発揮します。



デジタル多用途脳波計

サイナフィット5000

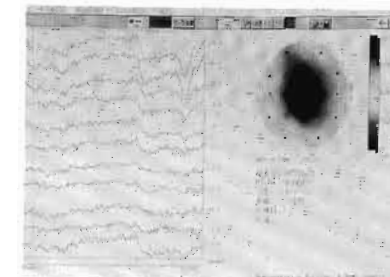
EE5114/EE5514 (ファイリング機能付) 記録器14ch型
 EE5118/EE5518 (ファイリング機能付) 記録器18ch型
 EE5121/EE5521 (ファイリング機能付) 記録器21ch型
 医療用具承認番号07B第0705号



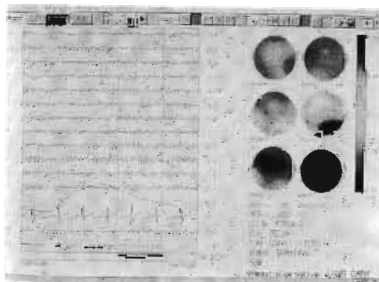
検査予約画面
1年間検査予約管理が可能。



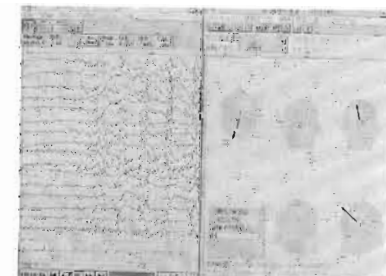
鮮明波形表示
高解像度CRTによる鮮明表示。



誘発電位(CNV) マッピング表示
一つひとつの誘発波形を確認しながら加算が可能。



周波数マッピング表示 (ATALAS)
スパイクの焦点やその周波数のパワーを把握。



ダイポール解析
スパイクの発生源や誘発電位の発生の推定を行うことが可能。

NECメディカルシステムズ

東京都文京区本郷3丁目42番6号(NKDビル) 〒113 ☎03(5684)1411
 NECメディカルシステムズのホームページ
 http://www.necms.co.jp/



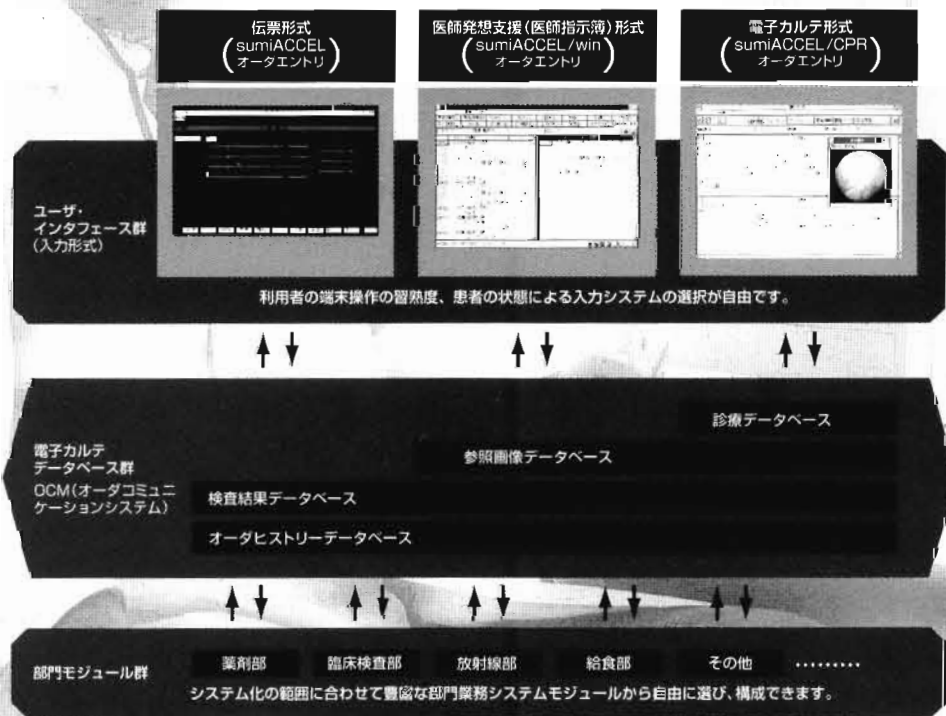
操作性も一段とアップ。 スムーズに電子カルテへ発展する sumiACCEL/win。

効率的な病院経営を支援し、患者サービスを向上。

スマアクセルWinは、病院全体で情報を共有・活用し、各部門の業務を大幅に効率化します。同時に、医療の質的向上、経営の効率化、患者サービス向上にも大きく貢献します。拡張性に優れたオープンなシステムにより、貴院の計画次第で、分院、外部医療機関などと結び、より総合的な医療情報ネットワークへと拡大することもできます。コミュニケーションを積極的に支援する住友電気独自のオーダコミュニケーションシステムを中心とし、電子カルテシステムへ最短距離をたどれる国内屈指の病院総合情報システムです。病院のインフラとして、成長するシステムをお選びください。Windows対応で操作性も一段と高まりました。

医師の思考過程に沿った診療支援機能が充実。

病院情報システムは、膨大な患者情報を体系的に管理し、多角的に分析・表示する「電子カルテシステム(CPR: Computerized Patient Records System)」に進んで行くことが予想されます。医師や看護婦の日常診療や業務をリアルタイムでサポートする電子カルテシステムが導入されると、医療従事者の作業効率の向上や医療レベルの安定化等がはかられ、医療品質のレベルアップが促進されます。sumiACCEL/winは、オーダエントリーから電子カルテシステムを連続してカバーする当社独自のオーダコミュニケーションシステム(OCM)と電子カルテシステムDB群により、オーダエントリーから電子カルテへスムーズに移行することができます。



卓越したユーザー・インタフェースとデータ共有性。

各部門の業務の流れに沿った画面デザインで、初めての人でも簡単に操作することができます。ユーザー・インタフェースは、使う方にとって最適な入力形式が選べるように、<伝票入力形式> <医師発想支援(医師指示簿)形式> <電子カルテ形式(sumiACCEL/win)形式>の3タイプが準備されていて、併用も可能です。電子カルテにもスムーズに移行することができます。

病院総合情報システム
sumiACCEL Win

EVER ADVANCE IN HEART 無限の可能性は人の心に――。



情報戦略のベストパートナー

私共はソフトウェア技術をもって、お客様の信頼できるパートナーとしてお客様のかかえる課題の解決に協力し、お客様がその競争相手に対して優位に立てるよう貢献致します。私共はこの貢献を通して私共自身を鍛え、よりよい生き甲斐と豊かさの実現を目指します。

高度な技術と豊富なノウハウをベースに、業種・業界を問わず幅広いニーズにお応えします。

[M言語開発実績/医療総合情報システム]

- ・処方オーダーリングシステム
- ・注射オーダーリングシステム
- ・食事オーダーリングシステム
- ・診療予約システム
- ・検査予約システム
- ・臨床検査システム
- ・生体検査システム
- ・入退院システム
- ・放射線システム
- ・薬剤部システム

NAS 株式会社
日本アドバンストシステム

- 本社 東京都港区芝5-13-14 アネックス三田ビル 〒108
TEL: 03-3455-2373 FAX: 03-3455-2927
- 大阪事業所 大阪府大阪市中央区北新町2-6 ツムラビル 〒540
TEL: 06-941-3211 FAX: 06-946-9746
- 沖縄事業所 沖縄県具志川市字州崎5-1 トピカルテクセンター内 〒904-22
TEL: 098-939-8412 FAX: 098-939-8406

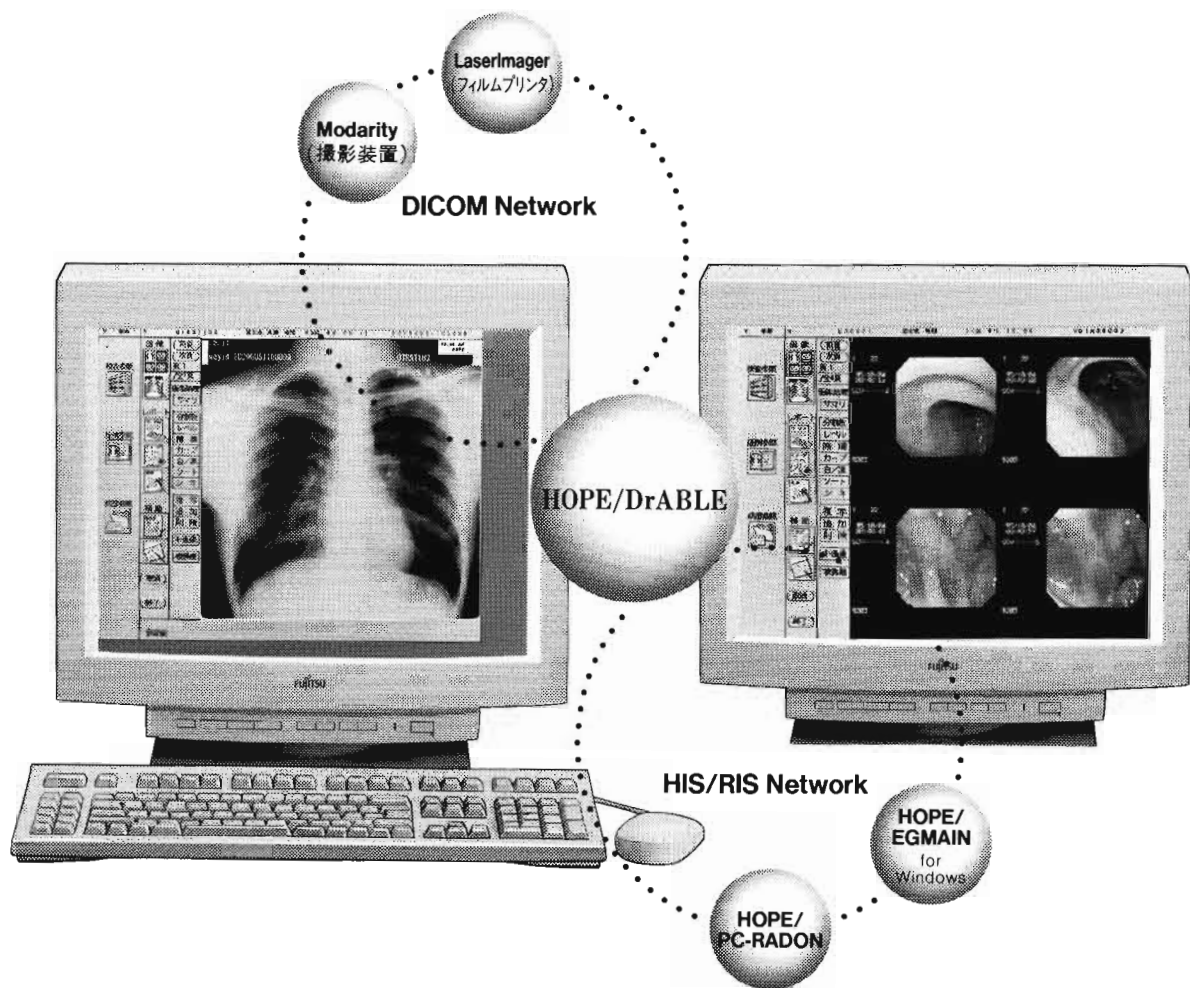
住友電気工業株式会社
情報通信システム事業部 MEシステム部システム営業課
住友電気システムズ株式会社
応用システム事業部

- 東京事業所 東京都港区元赤坂1-3-12 〒107 ☎(03)3423-5880(代)
- 大阪事業所 大阪府大阪市東淀川区4-11-28 サン船場ビル 〒542 ☎(06)258-5510(代)
- 中部事業所 名古屋市中区栄1-1-6 住友ビル 〒461 ☎(052)963-2755(代)
- 九州出張所 福岡市博多区博多駅前4-2-1 住友海上福岡ビル 〒812 ☎(092)441-1791(代)

住友電工 WWW ホームページ URL: "http://www.sei.co.jp"
または、BSC ネット URL: "http://www.bsnet.or.jp"

※sumiACCELは住友電気工業株式会社の登録商標です。※Windowsは米国Microsoft Corporationの米国及びその他の国における登録商標です。

画像データをネットワークで活かす!



HOPE/DrABLEは、CTやCRなどのモダリティから提供される画像データを一元管理し、読影およびレポートに必要な情報を即時に提供致します。

また、放射線部門システムとの連携による事務効率化や、オーダーリングシステムをはじめとする診療部門システムへの画像・レポート情報の提供も実現可能です。

HOPE/DrABLEの多彩なネットワークが、画像診断部門のオープン化を強力に推進します。

DICOM3.0対応

標準規格DICOM3.0に対応し、多くのモダリティ、フィルムプリンタとの接続が可能です。

柔軟なシステムを構築できる豊富なコンポーネント

- DrABLEコンポーネント
DICOM対応画像サーバ
DICOM対応読影ワークステーション
レポートPC(読影レポート作成システム)
ビデオIF(ビデオ画像収集システム)
- 放射線部門システム
HOPE/PC-RADON
- オーダーリングシステム
HOPE/EGMAIN for Windows

超高速、高画質を実現

従来の専用PACS以上のスピード、画質を汎用超高速64ビットワークステーションで実現しています。

多様なネットワークに対応

ATM、FDDI、FastEthernet、ISDNなどに対応可能です。

拡張性にすぐれたデファクトスタンダードのシステム環境

サーバ、読影ワークステーションにはSunSPARC station互換のSファミリー、レポートPC、ビデオIFにはPC/AT互換パソコンのFMVと、国際標準プラットフォームを採用したシステム環境は3DやTeleradiology、Internet、Macintosh接続などへの発展が容易です。

Windowsは、米国Microsoft Corporationの米国、およびその他の国における登録商標です。その他、記載の製品名などは、一般に各社の商標、または、登録商標です。

富士通医用画像情報システム

ホーフ/ドクターエイブル
HOPE/DrABLE

富士通株式会社 第二医療システム営業部 〒100 東京都千代田区丸の内1-6-1(丸の内センタービル) ☎(03)3216-9752(直)

協賛者 一覧

日本ダイナシステム株式会社

Voice: 052-242-5441 Fax: 052-242-5984

〒460 名古屋市名東区新栄2-1-9 雲竜ビル東館5F

日本デジタルイクイップメント株式会社 西日本第一統合システム部

Voice: 06-222-9211 Fax: 06-222-9408

〒530 大阪市北区中之島2-2-2

日本モトローラ株式会社 医療システム本部

Voice: 03-5487-4190 Fax: 03-5487-4199

〒141 東京都品川区西五反田4-32-1

丸善株式会社 情報図書館システム営業部

Voice: 03-3272-7027 Fax: 03-3271-8680

〒103 東京都中央区日本橋3-9-2 第二丸善ビル

NECメディカルシステムズ株式会社 東東京支店

Voice: 03-5684-1420 Fax: 03-5684-1423

〒113 東京都文京区本郷3-42-6 NKDビル

日本光電東京株式会社 本郷第一営業所

Voice: 03-3815-9431 Fax: 03-3815-9433

〒113 東京都文京区本郷2-27-20

アトムメディカル株式会社 クリティカルケア事業部

Voice: 03-3815-2491 Fax: 03-3812-3144

〒113 東京都文京区本郷3-18-15

住友電工システムズ株式会社 応用システム事業部

Voice: 06-258-5510 Fax: 06-258-5520

〒542 大阪府大阪市中央区南船場4-11-28 サン船場ビル

株式会社日本アドバンストシステム 営業部

Voice: 06-941-3211 Fax: 06-946-9746

〒540 大阪府大阪市中央区北新町2-6 ツムラ谷町ビル

富士通株式会社 医療システム事業部第二医療システム営業部

Voice: 03-3216-9752 Fax: 03-3216-3258

〒100 東京都千代田区丸の内1-6-1 丸の内センタービル

アイ・エム・アイ株式会社 東京営業所

Voice: 03-3816-4411 Fax: 03-3816-4420

〒113 東京都文京区本郷3-4-5

ラジオメーターレーディング株式会社 北関東営業所

Voice: 03-5704-6556 Fax: 03-5704-8080

〒153 東京都目黒区三田1-12-23 MT2ビル

日本Mテクノロジー学会

■ 日本Mテクノロジー学会 役員

学会長	河村 徹郎 (鈴鹿医療科学技術大学 医用工学部医用情報工学科)
庶務財務担当幹事	山本 和子 (島根医科大学医学部 医療情報学講座)
国際担当幹事	若井 一朗 (マンブスシステム研究所)
流通担当幹事	本多 正幸 (千葉大学医学部附属病院 医療情報部)
広報担当幹事	小林 泰道 (日本デジタルイクイップメント株式会社)
雑誌担当幹事	岡田 好一 (東海大学医学部 医用情報工学系)
ネットワーク担当幹事	木村 一元 (獨協医科大学 総合研究施設ME室)
標準化担当幹事	山下 芳範 (福井医科大学 医学情報センター)
JIS/ISO 担当幹事	大楠 陽一 (東海大学医学部 医用情報工学系)
会計監事	里村 洋一 (千葉大学医学部附属病院 医療情報部)

■ 日本Mテクノロジー学会 事務局

島根医科大学医学部 医療情報学講座 内

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

Fax: 0853-25-2764

Voice: 0853-23-2111 (ext.2942)

〒693 島根県出雲市塩冶町89-1

編集後記

このたびは第24回日本Mテクノロジー学会にお越し戴きまして誠に有り難う御座います。急速に進展する computing 環境のなか、Mの真価が問い直されている一方で、その底力を示すことのできる絶好の機会がおとづれてきたとも言えるでしょう。

巻頭言の問いに対する答えとして、各方面から斬新な論文が寄せられたことは、大会長として歓びにたえません。これらの成果ならびに大会での生産的な議論を礎として、Mのさらなる発展と普及とが実現することを願ってやみません。

本大会は様々な方々の御協力と御協賛によって組織運営されました。本大会の組織委員はもとより第3回Mテクノロジーフェア実行委員ならびに本学会事務局の皆様ほか、本大会の準備ならびに運営に携わってくださった方々、ならびに御協賛くださった各企業の皆様には、この場を借りまして厚く御礼申し上げます。

末筆とはなりましたが皆様の益々の御隆盛を祈念しつつ、筆を擱かせて戴きます。

平成九年初夏 廣瀬 康行

発行者 第24回日本Mテクノロジー学会大会 組織委員会
大会長 廣瀬 康行
事務局 東京都文京区湯島 1-5-45
東京医科歯科大学歯科麻酔学講座 内
03-5803-5551

企画意匠 廣瀬 康行
発行年 平成9年7月11日
編集調整 R.Moon 03-5532-8856
印刷製本 株式会社 立花製本 03-3591-5545