

カルテキーパー Ver.2 (電子カルテシステム Ver.3) 開発環境

筑波技術短期大学視覚部一般教育等¹⁾ 同鍼灸学科²⁾

村上佳久¹⁾ 上田正一²⁾

概要：視覚障害者に対応した電子カルテシステムを開発するに当たって、従来の開発環境ではない、新しい手法で開発を行った。これらは、視覚障害者向けのソフトウェアを開発するに当たり非常に有効な手法であると思われる。ここでは、その開発手法について解説する。

キーワード：Windows 開発環境、視覚障害補償、電子カルテ

1. はじめに

2003年より実施される文部科学省の新しい学習指導要領では盲学校の専攻科理療科、保健理療科などで「理療情報処理」「保健理療情報処理」等が実施されるが、学習指導要領では、電子カルテシステムや地域医療情報システムなどの医療情報システムを実習を通じて学ばせることと記している。

しかし、視覚障害者が利用できる電子カルテシステムなどは非常に少なく、また、そのシステムを実際導入している医療機関や治療院などは、極少数である。鍼灸学科では、実用的な視覚障害者用の電子カルテシステムを導入したが、利用し易さから言えば決して適切なものとは言えなかった。あくまでも紙のカルテを電子化したために実際の運用では不必要な部分が多く、電子カルテに入力されなかった経緯がある。

その反省を基に、主として簡易に利用できる「カルテキーパー」が開発され実際に運用されている。「カルテキーパー」は現在、鍼灸学科の実習室実習において、ボランティアのカルテを収録しており、この作業を通じてカルテ管理や電子カルテの実習を行っている。しかし、この簡易版電子カルテシステムである「カルテキーパー」は、MS-DOS版であり、旧式のNEC製PC-9801シリーズ上でしか動作しないと言う欠点があった。PC-9801シリーズがメーカーよりサポート停止の発表がなされ、また、GUI画面のWindows全盛になり新しいバージョンが求められるようになった。

ここでは、「カルテキーパー」の新しいバージョンの開発環境について解説し、視覚障害者向けのソフトウェアの開発環境や音声化への対応などをどのように行うのかについて、研究開発を通じて得られた知見を報告する。

2. Windows 開発環境 (各OSの決定)

Windowsの開発環境は事実上、Microsoft社の販売する開発言語シリーズを利用せざるを得ないのが現実である。しかし、販売されている様々な開発環境用ソフトウェアには学習用と試験開発用、製品開発用の3種類に分

けられる。一見すると同じような構成であるが実際には内容が大幅に異なり、また、利用するData BaseやGUI (Data Baseの見かけ上の利用メニュー)によっては、他の開発環境用ソフトウェアを合わせて利用せざるを得ない場合もある。今回の「カルテキーパー」の開発環境を整備するに当たり、様々な検討を行い実際に実験を行いながら最適な環境を見つけるべく検討を行った。ここでは、3つのOSを決定した。

2.1 開発OS

事実上、プロテクトモードで動作するWindows環境はWindows NT またはWindows 2000しかなく、安定度から言えばWindows NT Service Pack 6a がよいが、開発環境のモジュールの整合性と安定度を評価したところ、特にネットワーク関係でWindows 2000の方が安定なことがわかったために、今回は開発環境OSをWindows 2000とした。

2.2 端末OS

端末のOSを決定する要素は幾つかある。はじめに、Data Baseの利用方法について、今回の「カルテキーパー」では、クライアントサーバ方式を採用し、サーバ上のデータを端末が利用し、最終的にサーバ側で全ての管理を行うことを想定した。そのために、端末として必要な要件は次のようなものとなる。

端末OS

画面読み合成音声ソフトウェア

画面拡大ソフトウェア

Data Base GUI

ネットワーク通信環境

日本語入力環境

バイオメトリックス (生体認証) システム

この中で、 は、電子カルテシステムの秘守義務に関わる部分について関係者以外に情報を与えず、視覚障害者でも簡易にできる認証方式として採用した。他の生体認証方式として、虹彩認証は利用できない眼疾の学生が

多いため利用不可とした。また、音声認識は現時点では指紋認証よりも個人認識率と他人排他率が劣るために利用不可となった。

本学特有の問題として の視覚障害補償は不可欠な部分である。OSの安定度と共に重要な部分となる。

さらに、Data BaseのGUI画面をMicrosoft社のOfficeシリーズのAccessを利用して設計したが、音声化が不可能なため、Visual C++で書き直した。実際に使用するData Baseエンジンは別途決定し、DAO(Data Access Objects)を利用する。この場合、DAOは、Data Baseのデータのやり取りのみを担当する。これらの条件を基に幾つかのOSについて検討した。

安定度から言えば、UNIX系のOSの方が安定度はよいが、画面読み合成音声や画面拡大などの視覚障害補償の関係から、Windows系のOSに絞って検討した。

Windows 98/Meなどの非プロテクトモードのOSでは、上記の ~ までのどれかにトラブルが起きてもOSの機能を失う。Windows NT/2000では、プロテクトモードで動作するために ~ のタスクは保護されるので、その意味では安定動作となる。しかし、特に画面読み合成音声ソフトの画面読み部分が、Windows 98/Meに比べて十分ではない問題がある。そこで、現在販売されている画面読み合成音声ソフト3種類とWindows 98/Me/NT/2000で、テストを行ったところ、1つの組み合わせで非常に安定動作を示したので、Windows 2000を端末OSとした。

2.3 サーバOS

クライアントサーバ方式では、サーバOSは、別にWindows系でなくともよい。特にData BaseサーバではUNIX系の方が遙かに安定で実績もある。また、個人の利用者環境を保存する「ディレクトリ・サービス」もWindows系でもUNIX系でも運用可能である。しかし、専門の知識を多数必要とする見かけ上複雑なシステムよりも見かけ上単純なシステムの方が運用し易いであろうと言う安易な理由から、Windows系のWindows 2000サーバを選択した。但し、見かけ上簡単に見えるだけであって実際の保守運用は非常に高度な知識を要求される。また、デフォルトで設定される各種のセキュリティや初期設定事項は、電子カルテのような秘守義務を伴う厳密なセキュリティを要求される場合には、あまりにも不都合が多く、その設定には、MSCE(マイクロソフト認定システムエンジニア)クラスの技術者が不可欠となるので注意が必要である。

3 . Data Baseエンジンの決定

「カルテキーパー」の中心的存在が、Data Baseエンジンである。今回の「カルテキーパー」では、

1日当たりの患者数：約50件

1週当たりの総数：約100件

年間データ総数：5000件

運用期間8年

総データ数：20000件

のData Baseを想定する。

Data Base GUIよりのクエリー要求をData Baseエンジンで処理して、端末のGUIに結果を転送する。これがクライアントサーバ方式の特徴であるが、Windows系のData Baseエンジンには、現在次のようなものがある。

FoxPro (dBASE 互換)

SQL

JET (Access 互換)

MSDB (SQL 互換)

この中で最も強力なのが、SQLである。しかし、SQLは高価であり、またData Baseの規模から言ってオーバークオリティである。そこで、今回は と の両方の混在環境で両者の特性を生かした設計を行った。即ち、Data Base GUIの整合性から を利用し、Data Baseの安定性から を包括する。この両者は、サーバ上でデータをクエリーでデータのやり取りし、変換される。将来的にData Baseが非常に大きくなった場合(本来あり得ないが)を想定してSQLへの移行が容易なように考慮した。

4 . 開発環境の整備

はじめに利用者端末と開発環境の両方を整備する。何故なら開発されたソフトを常に検証するために利用者端末の存在は欠かせないものとなる。

両者ともOSはWindows 2000であるが、両者のバージョンは異なる。また、開発環境の様々な開発ソフトウェアには全てDeveloper版を導入した。

利用者端末：Windows 2000 Professional 版

開発環境：Windows 2000 Developer 版

Developer Studio Enterprise 版

Office 2000 Developer 版

また、画面読み合成音声ソフトは、2000Reader(95Reader Ver.4)を導入し、音声出力用のサウンドボードは、YAMAHA製のYMF744サウンドIC搭載のサウンドボードを改造したのものを利用し、サウンドボード用のデバイスドライバも一部改造したのものを利用した。これらの改造は、安定動作のためと音声の多重出力を可能とするためである。

注意すべき点は、開発環境のソフトウェアのインストールの順番である。これを間違えるとモジュールの一部がバージョン違いとなり後に影響を与える。従って、上記の順にインストールを行い、更にOSのService Packを導入する。この開発環境用のソフトウェアを全て導入するまでは他のソフトウェアは一切導入すべきでない。我々が行った実験では、一部のモジュールに影響を与え、特に合成音声の出力が行われにくくなるので注意が必要である。

利用者環境は、開発環境の一部のモジュールをインストールする必要がある。これは、モジュールのバージョンの整合性を取るためである。それ以外には大きな注意点は無い。但し、統合型マザーボードチップセット（例えば、Intel 810など）の場合には、Windows 2000に対応したBIOSのアップデートが必要不可欠である。

今回の開発環境では、Intel製のチップセットは利用せずにサーバ系のチップセット（Server Works LE III）を利用して、高性能化を図った。

開発環境の仕様は、次の通りである。

チップセット：Server Works LE III
CPU：Intel Pentium III 733MHz x2
RAM：Registered ECC SDRAM 256MB x1
HDD：SCSI 160 10000rpm 18GB x2
NIC：100BASE-T (32bit PCI), 1000BASE-T (64bit PCI)

最新の超高速ハードディスクは発熱量が多いためケースには合わせて6つのファンがついており内1つがハードディスク専用である。また、超高速ネットワークに対応すべく、1000BASE-Tのネットワークボードを導入している。このネットワークボードは64bit PCIバス対応なので、対応したチップセットとマザーボードが不可欠である。

さらに、2CPUにしてプログラム・コンパイル時の負荷を出来るだけ軽減できるようにした。なお、メモリはこのチップセット専用のレジスタードメモリと呼ばれる特別の物である。

5. 開発の工程

今回の開発環境では、「カルテキーパー」の仕様よりも、Data Baseの開発環境とGUIの合成音声対応に重点を置いた。つまり、視覚障害者対応のソフトウェア開発を最重点項目としている。

一般的には、開発環境で様々なソフトウェアを開発するが、今回の場合は逆に、目的別開発環境を先に整え、開発環境に沿った視覚障害者対応のソフトウェアを開発

している。これは、一見すると奇妙に思えるかもしれないが、今回の開発時間が著しく短くできたのは、プログラマーの努力と共に開発環境整備に妙がある。

実際の工程は次の通りである。

端末用Data Base・GUI作成
初期Data Base項目決定
視覚障害補償用画面設計
Data Baseエンジン決定
Data BaseエンジンとData Baseフロントエンドの通信
試験サーバ上での運用テスト
バグ取り

ここで、とに不思議に思われるかもしれない。先に画面構成を決めてから、視覚障害補償用画面設計を行っている。これは、従来の手法では技術的問題からData Baseと画面設計が同一事象で行う必要があったものが、Data Baseの設計と画面構成の設計の2つが独立した事象で行うことが可能であるため、両者を一致させるために、先にGUIを作成している。実際に作成されたGUIに従って作成されるクエリーを利用する視覚障害補償用画面設計を再度設計し直しているのである。

一見すると無駄が多く感じるかもしれないが、Data Baseそのものが直接データを参照するのではなく、クエリーを利用するというクライアントサーバシステムの本質に関わるからである。

の視覚障害補償用画面設計は、プログラマーの設計に任せましたが、画面拡大などの処理は視覚障害補償用ソフトに任せることにして、画面読み合成音声ソフトの音声出力に合致させるべく試行錯誤を繰り返した。基本的にはテキストボックスについては、音声化は可能であるがカーソル移動に伴う位置情報の再帰と音声出力の整合性が一番の問題となるが、Windows系でもWindowsNT/2000とWindows 98/Meとでは同じでないのが一番苦慮する点である。今回の場合は端末のOSをWindows 2000としているため、これに合わせた設計を行っているために他のOSに対する視覚障害補償は保証出来ない。

でData Baseエンジンは、前述のようにJetエンジンを主体にMSDBを包括するため、先に画面設計を行い、後にサーバとの通信実験を行った。その為、いくつかの点で問題点が発生したが、基本的には、Data BaseエンジンとData Baseフロントエンドの通信がクエリーで行われるために、他のデータの排他処理と最終マージ処理だけが問題となる。この部分は、SQL等では、自動的に行われる部分であるが、合成音声出力を行っている

る場合には、その分遅延が発生するために、排他処理には注意を払った。

は、サーバの横でHUBで接続された環境と、ネットワークケーブルで大きく離れた場所で行われる環境とでは通信環境が異なるために、実運用の条件下で何度も実験を行い、遅延に対する対策を講じた。

は最大の問題である。バグのないプログラムはないと言われる位に開発にバグは付き物であるが、今回の開発環境では、そのような事態が、出来るだけ起きないように開発環境を非常に狭めたものになっている。実際、大きなバグは見あたらなかったが、GUI 画面構成などで幾つかのバグが発生している。

6 . おわりに

今回の事例は、カルテキーパー(電子カルテシステム)の開発について、その環境をどのように整備するかによって、データベースの開発、視覚障害者向けのソフトウェアの開発という、2点について非常に短時間で効率的な開発が出来たと自負している。

今回の開発環境では、他のソフトウェアの開発を行う場合には、それように開発環境を変更する必要がある。しかし、変更する場合も、はじめに開発目的をはっきり

させ、どのような仕様を実現する開発環境にするのかという従来の開発環境とは逆の発想で行えば、開発工程や開発時間を著しく短縮することが可能となろう。

今回の開発環境の整備の副産物として、Windows で、視覚障害者自身がソフトウェア開発を行う場合の環境を整備することが可能となることが判明した。

開発環境の整備とは、様々なノウハウの積み上げでもある。一朝一夕に簡単に出来るわけではない。

視覚部の開学当初にみられた、素人によるプログラム開発が如何に無謀で無利益なものあるのかを認識出来ない場合など論外であり、システムエンジニアとプログラマーによるプログラム開発が重要あることは言うまでもない。

今回得られたノウハウは次の新しい開発環境の開発に役立つであろうが、このようなノウハウの蓄積と公開は、視覚障害者の高等教育機関としての本学の使命の1つでもある。

参考文献：

MSDN 開発環境, Microsoft, Microsoft Press, 2000

DAO 開発環境, Microsoft, Microsoft Press, 2000

Development Environment for Clinical Record Keeper Ver.2 (Electronic Clinical Record Ver.3)

MURAKAMI Yoshihisa ¹⁾, UEDA Shoichi ²⁾

¹⁾Department of General Education, Division for the Visually Impaired, Tsukuba College of Technology

²⁾Department of Acupuncture/Moxibustion, Tsukuba College of Technology

Abstract: We developed a new technique, which differed from a past development environment when the electronic clinical record system corresponding to the visually impaired person was first developed. These seem to be very effective techniques for software development, for the visually impaired person. We explain the development method.

Key Words: Windows development environment, Visual disturbance amends, Electronic clinical record