

# 3層構造データベースの 実験モデル構築と教育利用

森 本 雅 博

## 1. はじめに

今日のインターネットの普及は情報化の進行を加速度的に増大させ、わたしたちはこれまでになかった新しい技術や多くの情報を簡単に手に入れることができるようになってきた。教育分野においては、このような時代へ対応するため、全国100校程度の小中高校にコンピュータを設置し、高度化したコンピュータ技術、ネットワーク技術を活かしながら、従来の枠組みを越えた教育・学習の可能性を実証することを目的としたいわゆる「文部省100校プロジェクト<sup>1)</sup>」を推進している。また、「インターネットの教育利用」というテーマのもと、あらゆる教育現場でその取り組みの研究、実践活動が行われている。一方企業においては、インターネットによるビジネス活動の拡大やグループウェアによるビジネス・スタイルの変革、インターネットコマースによる新しい商取引技術の開発等々、インターネット技術の応用分野への取り組みが盛んに進められている。

本稿は、このような環境で、今日的技術であるイントラネットの教育分野への導入という観点に立脚して、従来の学内クライアント／サーバシステムによるデータベースシステムを WebSystem による3層構造のデータベースシステムへ移行するための実験モデルの構築を試み、その教育的利用を考察する。

## 2. 従来型データベースモデルの変遷

### 2.1. ホストスレイブモデル

フォン・ノイマン型コンピュータの出現以来長らく、基幹システムはメインフレームと呼ばれる大型のホストコンピュータを中心に据え、ノンインテリジェントの端末機が接続されたシステム (Host-Slave System) がコンピュータ・システムの一般的な形態であった。(図1)このようなシステムでは、ホストコンピュータがすべてのデータを管理し、端末機からのすべての要求を処理する。同時アクセスに対して、TSS (Time Sharing System) という方式により、リアルタイムの応答も可能としたのである。しか

し、メインフレームによるシステムでは、端末機からのすべての要求をホストコンピュータが中央集権的に処理するため、接続される端末台数の増加に伴ってレスポンスが低下してしまう。その上、ホストコンピュータの一元管理による弊害も徐々に出現してきたのである。すなわち、ホストコンピュータは端末機からの要求に対して処理するものであるが、その処理はあらかじめ定められたものに限られたものであった。

そのため新たな要求に迅速に応答していくことは困難であり、部門からの要求が蓄積

されてきた頃をみはからってシステムの入替えやバージョンアップを施すという運営であった。また、大型汎用機の価格は数千万円するのが当たり前で、なおかつ運用、管理にも非常にコストがかかるものであった。大型コンピュータの運用メリットは、結局のところ大量のデータを一元管理のもとで処理できるということにあり、多様な要求や即応性という点においては限界があったと言わざるをえない。

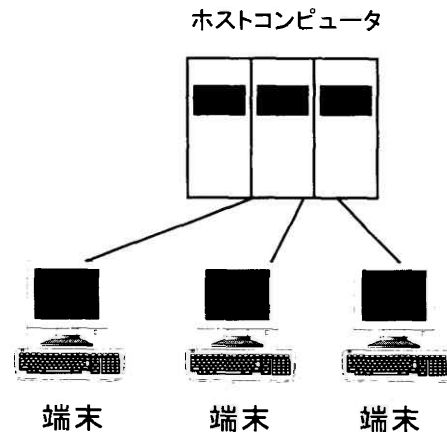


図1 ホストスレイブモデル

## 2.2. クライアント/サーバシステムモデル

コンピュータ開発の進展は急速に進み、大型コンピュータを凌ぐパフォーマンスを有する中型、小型の安価なコンピュータが出現してきた。パソコン（PC）と呼ばれる小型の個人使用のコンピュータは、現行モデルにおいて1980年当時の大型コンピュータをはるかに凌ぐものである。ノンインテリジェントな端末機は、やがてインテリジェント端末機へと移行していった。ホストコンピュータである大型コンピュータもワークステーション（WS）や高性能パソコンに取って代わられるようになっていった。この方向性を“ダウンサイジング”と呼んだのである。

ダウンサイジングの中心に据えられたシステムがクライアント/サーバシステムであった。（図2）ネットワーク上のクライアントであるインテリジェント・マシンすなわちPCが情報処理のキーマンとなり、サーバと呼ばれる高性能マシンがクライアントからの要求に対して迅速に応答するというシステム形態である。このシステムによって情報の部門所有が可能となり、基幹業務と部門業務の住み分けも実現できるようになった。いわゆる情報の分散処理である。部門独自の多様な要求に対しては、部門ごとに自己解決がはかれ、迅速な処理が可能となったのである。コスト面においても莫大な費用を要する大型コンピュータと違い、高性能で安価なPCの導入によるかなりのコストダウン

が実現できた。

### 2.3. EUCとコストの問題

PCの普及により、専門技術者であったコンピュータ部員は小人数で足り、代わりにシステム・アドミニストレータの存在が注目を浴びるようになってきた。彼らは部門ごとの情報化推進者であり、専門技術者と一般ユーザ (End User) との接点として双方の要求を円滑に処理する任務を負うものである。定型的な情報処理操作を行っていた一般ユーザが自己の判断のもとに非定型的処理を行おうとするとき、的確なアドバイスを与える人材でもある。しかし、以

前指摘したようにこの人材の養成が遅れているため、エンド・ユーザー・コンピューティング (End User Computing) の実現には程遠い現状が<sup>2)</sup> 厳然とある。

また、クライアントの増加は、クライアントごとに挿入するアプリケーション・コストを肥大化させることになる。バージョンが上がれば、それに対応すべく新たな教育も必要になる。次々と新しい技術や方法が現れるたびに、アップグレードを施していくことは管理や教育のためのコストに大きく影響を及ぼすのである。

### 2.4. インターネットの普及

1994年頃からのWWW (World Wide Web) ブームにより、研究者のためのものから大衆者のものへと変容してきている様は、図3の数値とグラフから読み取れるだろう。インターネットに接続するコンピュータの台数は、1981年8月当時わずか213台であったものが、5年後の1986年2月には10倍になり、1987年12月には2.8万台となり100倍を越えた。1994年1月に220万台に増加し、1995年1月に585万台、1996年1月に1,435万台、1997年1月に2,182万台、1998年1月に2,967万台と倍々ゲームのごとく伸びてきた。1998年8月現在で3,674万台となっている。最近のインターネット上の統計よりグラフ化したものを下図に示す。

今年度の文部省調査による公立の小、中、高校および特殊学校の接続状況 (表1) は、長崎県の4.1%から岐阜県の99.0% (全国平均18.7%) までかなりの開きがある。各自治体の教育対策が一様ではないので、このような地域格差に表われているという現状である。教育現場にますます情報化の波が押し寄せて来るとい状況下で、早急に解決しな

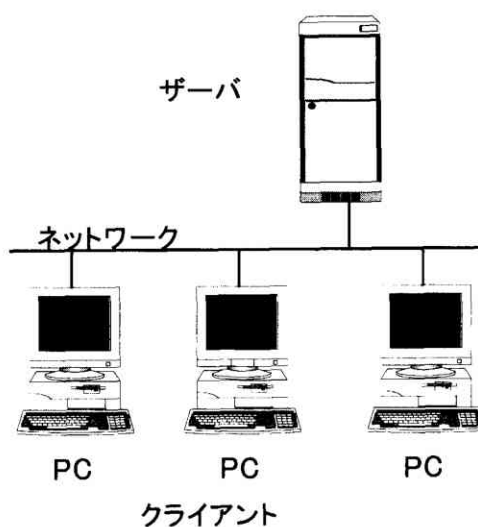


図2 クライアント/サーバシステム

3層構造データベースの実験モデル構築と教育利用

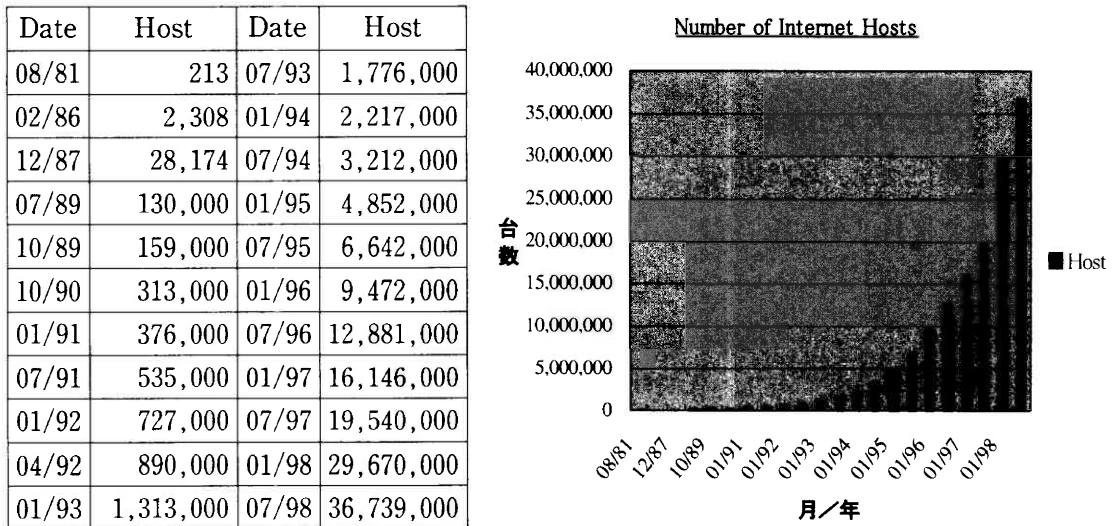


図3 Number of Internet Hostsの統計資料をグラフ化<sup>3)</sup>

表1 公立の小、中、高校および特殊学校の接続状況<sup>4)</sup>  
文部省調査（平成10年3月末現在）

北海道	8.5	石川	35.5	岡山	12.4
青森	19.2	福井	26.4	広島	10.2
岩手	8.9	山梨	27.0	山口	10.2
宮城	13.8	長野	30.5	徳島	26.9
秋田	35.1	岐阜	99.0	香川	43.8
山形	20.7	静岡	14.2	愛媛	14.0
福島	7.6	愛知	15.1	高知	74.9
茨城	19.8	三重	15.1	福岡	11.5
栃木	10.2	滋賀	12.5	佐賀	70.0
群馬	25.0	京都府	16.3	長崎	4.1
埼玉	11.4	大阪府	27.3	熊本	23.8
千葉	9.6	兵庫	31.8	大分	6.8
東京都	5.1	奈良	10.3	宮崎	10.8
神奈川	19.0	和歌山	13.8	鹿児島	10.8
新潟	19.0	鳥取	21.2	沖縄	21.4
富山	27.7	島根	17.5	全国	18.7

都道府県別インターネット接続状況 (%)

なければならない問題は、生徒に適切な情報処理教育を実施できる人材の養成である。ハードの導入そのものは、予算があれば簡単にできるが埃をかぶるような状況にさせてはならないのである。しかし、小中高教員の県内留学や教育系大学の指導者養成への取り組みは徐々に結実されてきている。現職の半数近くの教員がパソコンを理解し、1/10は

指導できるという回答が新聞社アンケートでも寄せられている状況を考えると、近い将来この数字が増加方向で変化していくことを期待する。

### 3. イン트라ネットとWWWアプリケーション

#### 3.1. イン트라ネット

従来、インターネットのような外部のネットワークへのアクセスと、内部のネットワークへのアクセスには別々のハードウェア、ソフトウェアを使用し、全く別のネットワークとして使用するのが一般的だった。インターネットへのアクセスにはWWWブラウザやメールクライアントを使用するが、社内のネットワークに存在する情報へは専用のクライアント・ソフトウェアやファイルの種類に応じたアプリケーションを使用していた。そのため、クライアントに要求される機能が肥大化し、また、ユーザもいろいろなソフトウェア、ハードウェアの使い方を習得する必要があった。

インターネットの発展にともなって、T P C / I P、WWWをはじめとするインターネット・テクノロジーを内部ネットワークに流用することが考えられた。インターネット技術を応用した内部ネットワークは、「Inter (間の) -net」と区別するために「Intra (内側の) -net」と呼ばれ、近年の企業情報システムのトレンドとなっている。イントラネットでは、インターネットと同じ通信プロトコル、同じアプリケーション、同じファイルフォーマットを使用するため、クライアントはネットワークの内外を問わず、透過的にアクセスすることが可能になる。<sup>5)</sup>

#### 3.2. WWWアプリケーション

イントラネットでは一般的に、電子メールやネットニュース、WWWなどを使って内部システムを構築する。特にWWWはグラフィカルに情報を伝えられるため、効果的な情報伝達が可能である。もともとWWWは、HTML (Hyper Text Markup Language) というマークアップ言語を使って記述された静的な情報を配信するシステムである。そのため内部データベースへの情報の照会や登録、更新といった動的な情報操作を行うことはあまり得意ではなかった。しかしここ数年、WWWとデータベースを連動させてリアルタイムに動的な情報を提供する仕組みが急速に整備され、情報システムの中心的役割を担うプラットフォームとしても注目を集めている。このようなシステムでは、CGI (Common Gateway Interface) やJ A V A (Sun Microsystems社が提供するスクリプト言語) などの機能を利用することで常にユーザの要求に応じたHTMLファイルを生成し、クライアントに提供する。クライアント側では特に新しいソフトウェアをインストールすることなくWWWブラウザの機能だけで新しいシステムを利用できるため、管理コストや教育コストを削減することが可能である。

## 4. 3層構造データベースモデルとは

### 4.1. 単層モデルから2層モデルへ

上述のようにホストスレイブでは、すべての処理はホストコンピュータ側で行われる。つまり〈単層モデル〉である。こうしたメインフレームと端末とによるコンピュータ利用の在り方を大きく変えたのが、ミニコンピュータやワークステーションを使ったダウンサイジングの登場であり、クライアント／サーバシステムであった。クライアント／サーバシステムでは、ユーザーが操作するクライアント側と、その要求に応じて処理を行うサーバ側を分離し、その間はSQLに代表される標準化されたコマンドやデータをやりとりする。両者が異なるプラットフォームで実行できるし、ユーザーインタフェースも自由に構築できるようになった。このようにクライアント／サーバシステムは、コンピュータの利用方法に大きな転換をもたらしたのである。

従来のクライアント／サーバシステムは〈2層クライアント／サーバシステムモデル〉と呼ばれるものである。このモデルでは、アプリケーションをクライアント層とサーバ層に分け、両者をネットワークで接続した構成になっている。クライアント層はユーザーの操作によって情報の入力および出力を担当する。端末エミュレーションと異なり、マウスなどのポインティングデバイスを使ったGUI (Graphical User Interface) が多用される。ユーザーの操作によって発生した処理要求は、ネットワークを通じてサーバ層に送られ、サーバ層は要求に応じた処理を行い、その結果をクライアント層に戻す。主にサーバにデータベースを置いて、複数クライアントからのデータ検索・更新・登録などの要求に対応する。このシステムではサーバ層とクライアント層の機能の振り分け方により以下のようなパターンが考えられるが、いずれのパターンにも問題点が存在する。

表2 サーバ層とクライアント層の機能の振り分けパターン

	クライアント	サーバ	問題点
1	ユーザーインタフェースとアプリケーション機能を持つ	データベース機能のみを持つ	クライアント側の機能が肥大化
2	ユーザーインタフェースとアプリケーション機能の一部を持つ	残りのアプリケーションとデータベースの検索機能を持つ	実際には構築されない
3	ユーザーインタフェースのみを持つ	アプリケーションとデータベースの機能を持つ	クライアントの増加によりサーバへのアクセス頻度が高くなり、パフォーマンスの低下を起す

## 4.2. 2層クライアント/サーバモデルから3層クライアント/サーバモデルへ

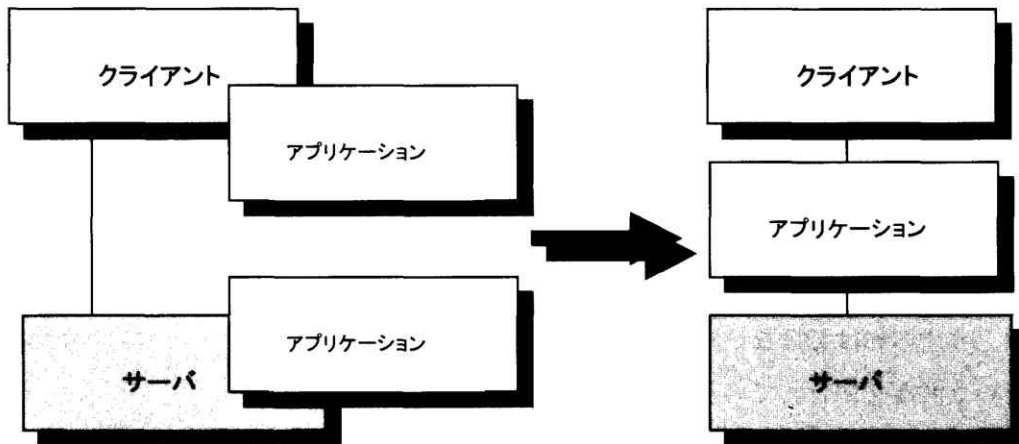


図5 3層クライアント/サーバモデル

2層モデルにおける上記の問題点を解決するために1995年頃から提唱され始めたのが、多層（マルチメディア）モデルというものであった。ユーザーインタフェースを持つクライアント層、データベース検索を行うサーバ層の間に、アプリケーション機能を持つ階層を設け、3層以上の階層を持たせる。この内、3層クライアント/サーバモデルが現在主流となっている。この中間層はミドルウェア層とかファンクション層と呼ばれるものである。

クライアント/サーバモデルを3層に分けることにより、アプリケーションの実行を担当するファンクション層と、データベース検索を担当するサーバ層の役割を切り離すことができる。2層モデルで発生しがちな、一つのサーバへのアクセスの集中を分散させ、アプリケーションごとに最適なプラットフォームで実行することが可能になる。したがって、より柔軟で大規模なシステムを構築できるようになるのである。

## 4.3. クライアント/サーバシステムとしての WebSystem

3層クライアント/サーバシステムの考え方は、WebSystemにも当てはめることができる。WebSystemはWebブラウザというクライアントと、Webサーバというサーバによって構成されたクライアント/サーバシステムと見ることができる。データはHTML言語によって書かれサーバ上に置かれており、WebブラウザがWebサーバに要求を送り、求められたHTMLファイルや画像ファイルを送り返すというのが基本的な仕組みである。

当初このシステムも作成者側からユーザーへの一方的な情報提供に過ぎないものであったが、CGIの出現により、ユーザからのデータをサーバに渡し、検索や加工などの処理を行うインタラクティブな仕組みが組み込まれるようになってきた。CGIでは

Webサーバから外部プログラムを起動させ、Webブラウザに表示されるフォームという入出力フォーマットを使用して、ユーザーからのデータを受け取ったり、データを検索したりといった操作を実現させている。

現在よく利用されている外部プログラムとしては、UNIXならシェルコマンドやPerlスクリプト、WindowsならCやVisualBasicで記述したプログラムやスクリプト、MacintoshならAppleScriptなどがある。またCGIとストアドプロシージャを使ってWebサーバからSQL ServerやOracleなど、データベースをアクセスすることも可能である。しかし、この場合直接APIレベルでWebサーバとデータベースプログラムがやりとりするわけではないため、サーバの規模が大きくなるとパフォーマンスが低下するという問題を内含している。

#### 4.4. 3層構造データベースモデルとしてのWebSystem

最近出てきたデータベースエンジンはこの問題の解決を示している。データベースエンジンと直接APIでやりとりできるWebサーバ、たとえば、SQL ServerとIIS (Internet Information Service) などがそれである。これらはWeb Serverをファンクション層として位置づけ、クライアント層とサーバ層を分離し、高度なパフォーマンスと拡張性を実現した。

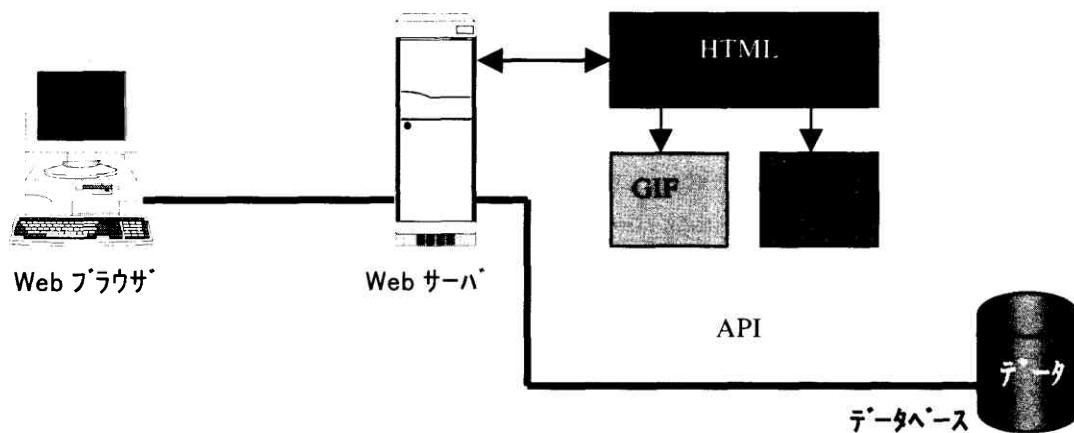


図6 3層 WebSystem

### 5. 3層構造データベース実験モデルの構築

上述のように、WebSystemを基本にした3層構造データベースモデルの概念をもとに、大学の一般的な学務処理の一部を実験対象としたモデルの構築を試みる。対象として、学生が行う「履修登録作業」と教員が行う「成績入力作業」をWebSystem上のシステムとして開発する。小中高等の教育機関の場合、教務側から生徒へ一方的で個定的な学務処理が実施されている。生徒の選択の要素が無いということで対象から外した。

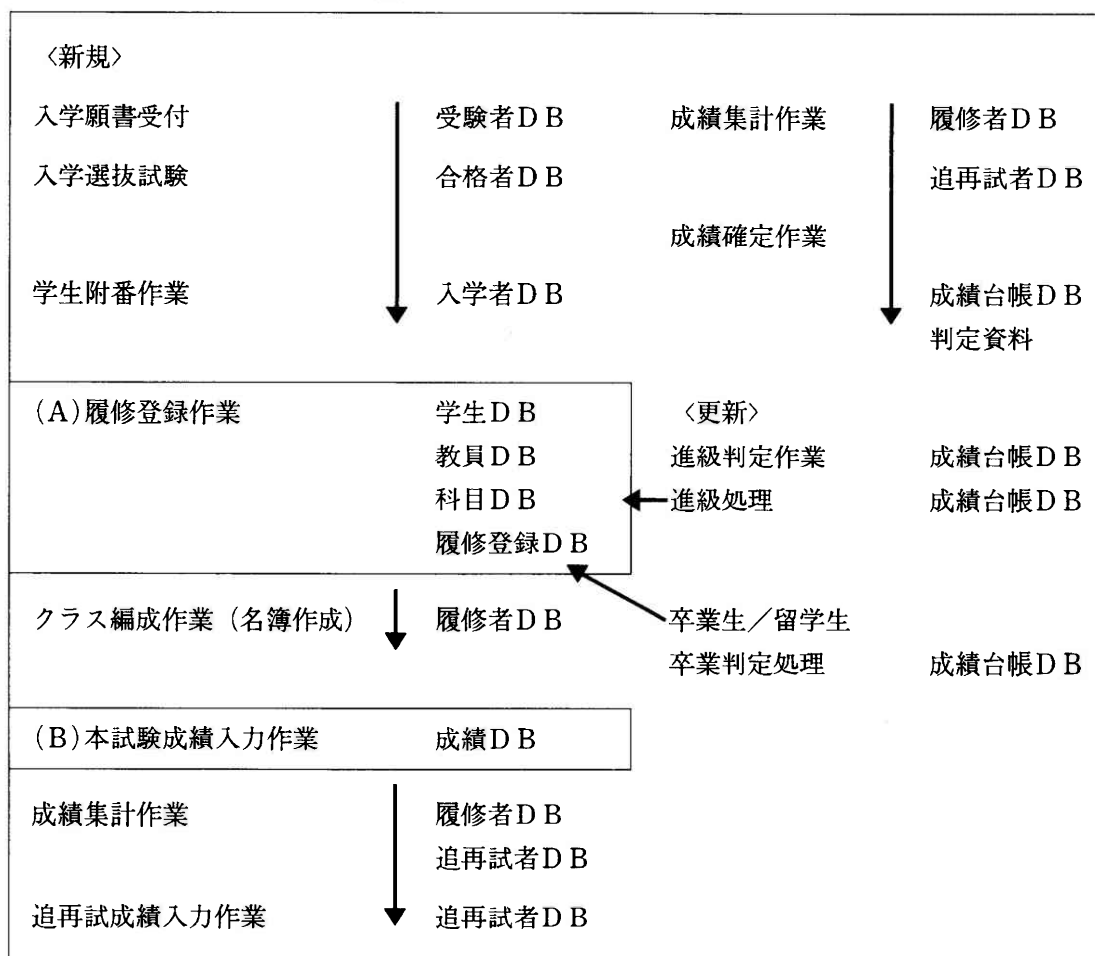


大学等の場合、履修登録に代表されるように、学生の選択の裁量がかなり存在し、時間割の組み立ては学生の単位取得意思に基づくものである。教務側の提示するシラバスや時間割表を参考にして自己の履修登録科目を決定する。その部分にシステムの導入価値があると考えたわけである。開発にあたっては、ユーザーの作業負担をできるだけ軽減することに考慮した。

### 5.1. ワークフローと作成されるデータベース

一般に大学では、新入生の入学手続きが完了した時点で学籍簿が作成される。その後、在学生と同様に科目履修登録手続きを行い、授業クラスが確定する。授業クラスに基づいて各試験を実施後、教員がその結果を成績入力し成績が確定する。この成績は進級判定や卒業判定の資料として、成績原簿に記帳される。このような業務フローが1年間を通しての学務的処理になろうかと思われる。一連の過程の中で下図のようなデータベースが形成され活用される。

図7 作業と作成データベース



※上記図中(A)と(B)が今回の実験モデル

## 5.2. ハードウェアとソフトウェア

本稿では、3層構造データベースモデルを以下の環境で構築する。

表3 ハードウェアとソフトウェア

ハードウェア	サーバー用PC・クライアント用PC	
ソフトウェア	サーバ	Windows NT・SQL Server・I I S
	クライアント	MS-Access・Internet Explorer MS-Access Upsizing Tools MS-Visual Database Tools T-SQL デバッガ

## 5.3. データベースの構築手順

実験モデルは ACCESS を中心にして作成する。

作成手順として、

- 1) ACCESS で、各データベースファイル（以下MDBファイル）を新規作成する。
- 2) それぞれのMDBファイルに、テーブルを作成する。
- 3) テーブルのフィールドに詳細設定（入力規則、既定値、インデックス）を行う。
- 4) テーブル間にリレーションを設定する。
- 5) 同時に参照整合性を設定する。
- 6) Upsizing Tools で SQL Server にアップサイジングする。
- 7) SQL Server の Enterprise Manager で、全オブジェクトの SQL スクリプトを生成する。
- 8) 自動型変換の修正
- 9) Microsoft Visual Database Tools で、ビューやストアドプロシージャやトリガを作成する。
- 10) T-SQL デバッガでストアドプロシージャをデバッグする。
- 11) 各オブジェクトに権限を与える。

## 5.4. 基本設計テーブル

以上の手順で作成したものが下記の設計テーブルである。

### ①学生マスター（表4）

フィールド名	フィールド型	
学生コード	数値型	プライマリーキー
学籍学号	文字型	インデックス

学 生 氏 名	文字型	
学 生 シ メ イ	文字型	インデックス
入 学 年 度	数値型	
学 年	数値型	
所 属 学 科	文字型	

②教員マスター (表5)

フィールド名	フィールド型	
教 員 コ ー ド	数値型	プライマリーキー
教 員 氏 名	文字型	
学 生 シ メ イ	文字型	インデックス

③科目マスター (表6)

フィールド名	フィールド型	
科 目 コ ー ド	数値型	プライマリーキー
科 目 名	文字型	
期 間	数値型	
単 位	数値型	
配 当 学 年	数値型	

④学科マスター (表7)

フィールド名	フィールド型	
学 科 コ ー ド	数値型	プライマリーキー
学 科 名	文字型	

⑤時間割テーブル (表8)

フィールド名	フィールド型	
時 間 割 コ ー ド	数値型	プライマリーキー
学 科 コ ー ド	数値型	インデックス
学 年	数値型	インデックス
曜 日	文字型	
時 間	数値型	
科 目 コ ー ド	数値	
教 員 コ ー ド	数値	

⑥履修登録テーブル (表9)

フィールド名	フィールド型	
学 生 コード	数値型	インデックス
学 科 コード	文字型	インデックス
時 間 割 コード	数値型	インデックス
履修登録コード	数値型	インデックス

⑦成績入力テーブル (表10)

フィールド名	フィールド型	
教 員 コード	数値型	インデックス
科 目 コード	数値型	インデックス
履修登録コード	数値型	インデックス
評 価 点	数値型	
評 価 コード	文字型	

⑧成績原簿テーブル (表11)

フィールド名	フィールド型	
学 生 コード	数値型	インデックス
学 生 氏 名	文字型	
学 生 シ メ イ	文字型	インデックス
履修登録コード	数値型	インデックス
科 目 コード	数値型	インデックス
科 目 名	文字型	
期 間	数値型	
単 位	数値型	
教 員 コード	数値型	インデックス
教 員 氏 名	文字型	
評 価 点	数値型	
評 価 コード	文字型	
評 価	文字型	
判 定	文字型	インデックス

上記表中、プライマリーキーは検索キーとして設定する。また、インデックスはクエリーでの抽出アイテムとして使用する。ここではプライマリーキーの属性を数値型に指定しているが、コード等の場合、文字型を指定する場合も多い。

### 5.5. 履修登録及び成績入力のリレーションシップ

上記基本テーブルより、下記のようなリレーションシップを設定する。

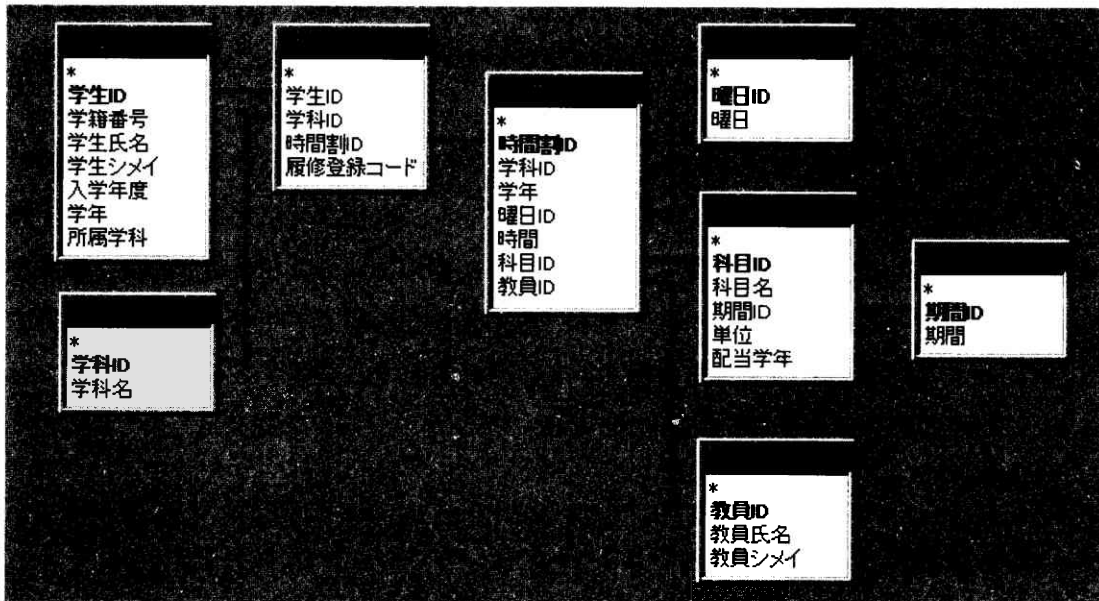


図8 履修登録リレーションシップ

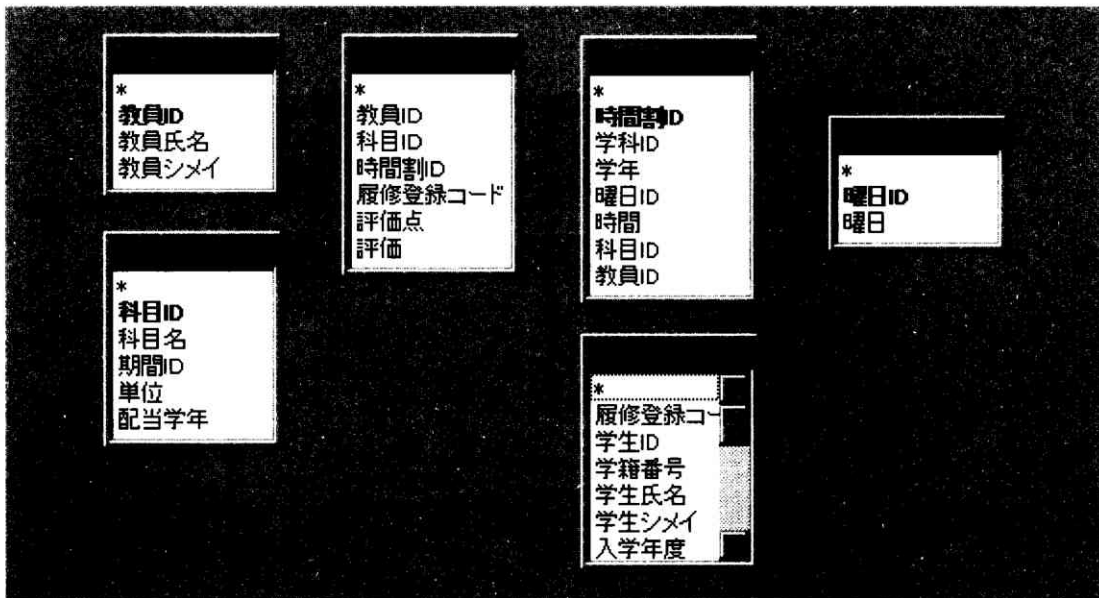


図9 成績入力リレーションシップ

### 5.6. 履修登録クエリーと成績入力クエリーのASP化

履修登録クエリー及び成績入力クエリーからそれぞれのフォームを作成し、これらをASP化する。

ASPとはHTMLページにスクリプトやActiveXサーバコンポーネントを組み合わせ、Web対応のアプリケーションを開発可能とする環境である。ASPによってデ

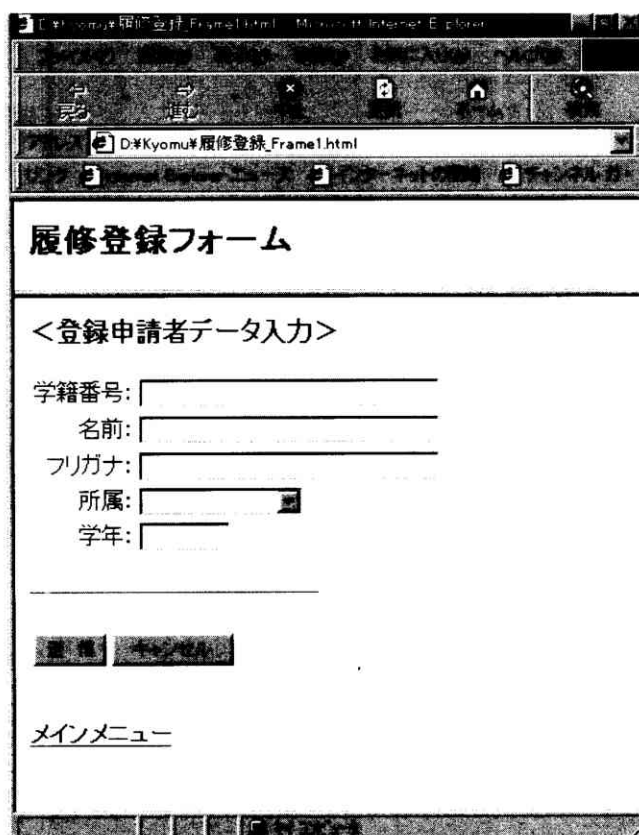
データベースの最新情報とリンクしたり、状況に応じて内容を更新するといった動的なコンテンツを容易に作成することができる。<sup>6)</sup>

ユーザーのアプリケーションに対する操作をできるだけ簡単にするために、GUI環境の整備を行わなければならない。Webブラウザを利用することでこの問題の解決が計れるわけである。すなわち、特定のアプリケーションを使用したクライアント/サーバシステムでは、「そのアプリケーションに対する教育」が常に必要になる。一時の学習でマスターできればいいが、それは容易なことではない。Webブラウザの優れたGUI環境を利用すれば、教育に費やされる時間がほとんど要らない。なぜならその主たる操作はマウスのクリック程度であるからである。逆に言うならばアプリケーションを開発する上で、そのようなGUI環境を提供できるように開発する必要があるということでもある。この点に関してASPを利用したWebアプリケーションの開発は、その容易性において優れているといえる。

## 6. 履修登録モデルの操作シミュレーション

### 6.1. 履修登録フォームへの基本データ入力

- ①学生がWebブラウザを操作して「履修登録フォーム」を開く。



The screenshot shows a web browser window with the address bar displaying 'D:\Kyomu\履修登録\_Frame1.html'. The page content is as follows:

**履修登録フォーム**

<登録申請者データ入力>

学籍番号:

名前:

フリガナ:

所属:

学年:

[メインメニュー](#)

図10 履修登録フォーム

- ②フォームに「学籍番号」・「名前」・「フリガナ」を入力する。
- ③「所属欄」は、コンボボックスを開いて、所定の学科名をクリックする。
- ④「学年」を入力する。
- ⑤すべてのデータを入力したら、〈送信ボタン〉をクリックする。この時点で先の入力されたデータが、データベースサーバへ転送される。画面上では「時間割フォーム」が表示される。
- ⑥入力した時点で、送信したくない場合は、〈キャンセルボタン〉をクリックする。キャンセルにより、すべてのデータは、データベースに送ることを中止する。
- ⑦〈メインメニュー〉タグをクリックすればホームページに戻る。

## 6.2. 時間割フォームにより履修登録

- ①入力された基本データとデータベースに登録されているデータとの照合により、学年と学科を抽出条件として抽出される時間割フォームが表示される。
- ②月曜日の1時限目から土曜日の5時間目までの枠内に登録されている「科目名」と「担当者名」が表示されたフレームが表示されている。
- ③できるだけキーボード操作を削減するため、各行の左端にあるチェックボックスに

履修登録フォーム

<時間割表示>

学年: 3  
曜日: 月  
時間: I

CHK	科目名	担当者
<input type="checkbox"/>	科目名3111	担当者3111
<input type="checkbox"/>	科目名3112	担当者3112
<input type="checkbox"/>	科目名3113	担当者3113
<input type="checkbox"/>	科目名3114	担当者3114
<input type="checkbox"/>	科目名3115	担当者3115

図11 時間割フォーム

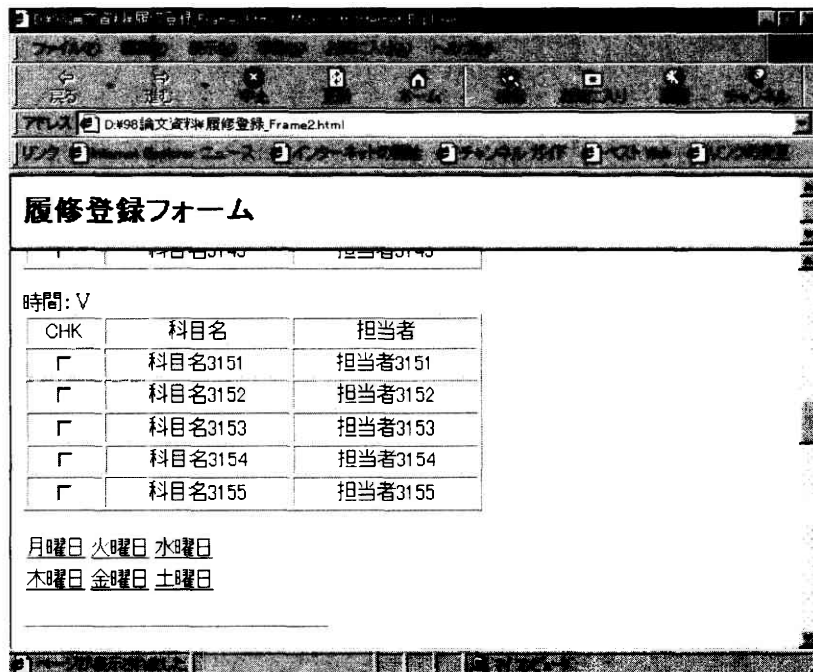


図12 曜日アンカー

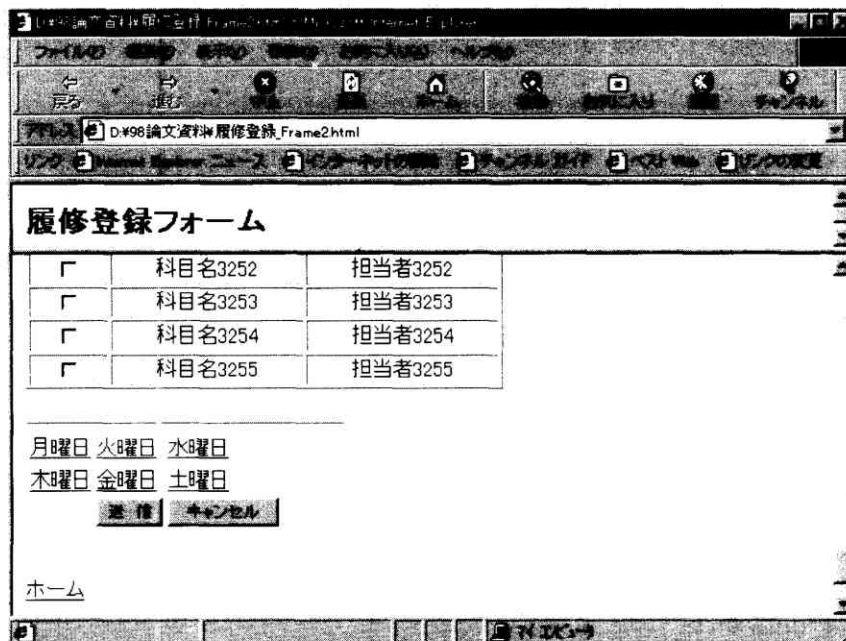


図13 送信・キャンセルボタン

チェックを入れるスタイルにする。このチェックにより、科目名と担当者を選択したことになる。

- ④①から③の作業を5時間目まで繰り返していく。
- ⑤曜日の最後になると〈曜日名〉タグが現れる。ここで飛びたい曜日をクリックすれば、その曜日の時間割が表示されるフレームへ移動する。



履修登録フォーム

<履修登録結果>

1998年度履修登録

学籍No	氏名	フリガナ	入学年度	学科
A99001	大手前花子	オオデマエハナコ	1999	A学科

月次	I		II		III		IV		V	
	科目名	担当者	科目名	担当者	科目名	担当者	科目名	担当者	科目名	担当者
	科目名3111	担当者3111	科目名3112	担当者3112			科目名3114	担当者3114		
					科目名3123	担当者3123	科目名3124	担当者3124	科目名3125	担当者3125
水	科目名3131	担当者3131	科目名3132	担当者3132						
木			科目名3142	担当者3142			科目名3144	担当者3144		
金							科目名3154	担当者3154	科目名3155	担当者3155
土										

上記の内容でよければ<登録>を、中止する場合は<キャンセル>をクリックしてください。

図14 履修登録確認フォーム

- ⑥履修のための時間割チェック作業がすべて完了すれば、最後のページで<送信ボタン>をクリックする。チェックされた部分がデータベースサーバへ送られトランザクションファイルとして登録される。
- ⑦<キャンセルボタン>をクリックするとすべてのチェック項目はキャンセルされる。
- ⑧履修登録確認のため、データベースより登録済みデータが返信されてくる。
- ⑨画面で確認し、修正がなければ<登録ボタン>をクリックする。データベースに正式に登録され履修登録作業は終了する。
- ⑩<キャンセルボタン>をクリックするとすべてのチェック項目はキャンセルされる。

### 6.3. 登録データのチェック

履修登録作業の終わったデータは、データベースサーバ内の履修登録マスターに格納されているので、履修条件によるマスターファイルのチェックを行う。本来的には、細かなチェック項目を設定すべきであるがその点は割愛した。あくまでもプロトタイプ的なものを試作することがねらいであるためである。しかし、ここまでの作業が進んでいけばチェック作業のためのテーブルは簡単である。あるいは基幹データベースに落とし込み、そちらでチェックを掛けるという方法でもよいであろう。

## 7. 成績入力モデルの操作シミュレーション

### 7.1. 成績入力フォームへの基本データ入力

- ①「成績入力フォーム」を開く。

成績入力フォーム

<成績入力者データ入力>

氏名:

コードNo.:

パスワード:

所属:

担当学年:

担当科目:

曜日:

時間:

[メインメニュー](#)

図15 成績入力フォーム

- ② 「氏名」・「教員コード」を入力する。
- ③ 「パスワード」欄にネットワーク・ログイン・パスワードを入力する。これは、他人が誤って成績ファイルを開かないためのセキュリティ処理である。
- ④ 「所属」欄はコンボボックスになっているので、所定の学科名をクリックすればよい。
- ⑤ 「担当学年」・「担当科目」を入力する。
- ⑥ 「曜日」をコンボボックスより選択する。
- ⑦ 「時間」を入力する。
- ⑧ すべての項目を入力し、誤りがなければ「送信ボタン」をクリックする。
- ⑨ 「キャンセルボタン」をクリックした場合は、中止となる。

## 7.2. 評価点数及び評価コード入力作業

- ①送信されたデータより抽出された「成績入力フォーム」が表示される。フォームに

成績入力フォーム

平成10年度

科目名	担当者
KMKA31111	KYNA111

時間割CODE	曜日	時限目
JCDA31111	月	1

学籍番号	氏名	フリガナ	点数	備考
A99001	AAAAAA	aaaaaa	<input type="text"/>	F
A99002	BBBBBB	bbbbbb	<input type="text"/>	F
A99003	CCCCCC	cccccc	<input type="text"/>	F
A99004	DDDDDD	dddddd	<input type="text"/>	F
A99005	EEEEEE	eeeeee	<input type="text"/>	F
A99006	FFFFFF	ffffff	<input type="text"/>	F
A99007	GGGGGG	gggggg	<input type="text"/>	F

図16 評価入力フォーム

成績入力フォーム

A99015	OOOOOO	oooooo	<input type="text"/>	F
A99016	PPPPPP	pppppp	<input type="text"/>	F
A99017	QQQQQQ	qqqqqq	<input type="text"/>	F
A99018	RRRRRR	rrrrrr	<input type="text"/>	F
A99019	SSSSSS	ssssss	<input type="text"/>	F
A99020	TTTTTT	tttttt	<input type="text"/>	F

送信・キャンセルボタン

図17 送信・キャンセルボタン

は、担当科目名・担当者・時間割コード・曜日・時間が上段に、登録されている学生の学籍番号・氏名・フリガナが表示されている。

- ② 「点数」欄に点数を及び評価コードを入力する。

- ③すべての学生の点数あるいは評価コードを入力したら、〈送信ボタン〉をクリックする。データベースの成績マスターにその結果が転記される。
- ④〈キャンセルボタン〉をクリックすると中止される。

## 8. 実験モデルの評価

ここではモデルの操作性をもとに評価していく。

表12 履修登録作業

目 的	対 象	操 作
アプリケーションの立ち上げ	WWWブラウザ	マウスのWクリック
フォームの表示	履修登録フォーム	マウスのクリック
基本データ入力	履修登録〈基本データ入力〉	キーボード入力
基本データ送信	送信ボタン	マウスのクリック
登録項目チェック	時間割フォーム	マウスのクリック
ページの移動	曜日アンカー	マウスのクリック
登録済み確認	送信ボタン	マウスのクリック
登録作業完了	登録ボタン	マウスのクリック

表13 成績入力作業

目 的	対 象	操 作
アプリケーションの立ち上げ	WWWブラウザ	マウスのWクリック
フォームの表示	成績入力フォーム	マウスのクリック
入力者データ入力	成績入力フォーム	キーボード入力
入力者データ送信	送信ボタン	マウスのクリック
成績入力	成績入力フォーム	キーボード入力
入力作業完了	送信ボタン	マウスのクリック

構築したモデルの操作性を確認してみると、ほとんどがマウスのクリック操作である。従来のシステムでは、機能性を高めるほどに初心者ユーザーに難しい操作を強いてきたと言える。しかし、Webを利用したシステムでは、パソコンを十分に使いこなせないユーザーにとって非常に扱いやすい環境となっている。実験モデルも若干のキーボード操作はあるが、わずかな数量の文字入力とテンキーを使用した数値入力操作のみでデータ入力は済んでしまう。

履修登録作業で見ると、WWWブラウザを立ち上げ、履修登録のページを表示させる。履修登録に先立ち「学籍番号」と「氏名」を入力させ、あらかじめ登録されているネットワークユーザーとのセキュリティチェックを行う。もし登録ユーザーでなければ、こ

の時点で拒絶され、ここから先へは進めない。時間割フォームでは、画面上で時間割を確認しながらチェックマークを付けていく。この作業はマウスのボタン操作のみで、それ以外の操作は全く必要ない。曜日移動はアンカー上でマウスをクリックすればよい。このように最小のキー操作で登録作業や入力作業が完了してしまうという点が、Web-Systemの優れている点である。

成績入力作業の場合も、入力者データの「コード No」と「パスワード」を入力することでセキュリティチェックを行うようになっているので、他人が誤って作業を行うという危険性は回避できる。「担当科目」はコンボボックスを作成することで選択型に変更すれば、入力作業はより容易になるであろう。このデータを送信し、教員DBとの突き合わせにより本人の確認と、履修DBからの担当科目履修者のデータを呼び出す。表示された履修者名簿の順に「点数」項目に評価を入力し、最後に「送信」することですべての作業が完了する。履修登録と同様、簡単なキーボード操作とマウス操作のみで作業は完了するわけである。

このように、WWWブラウザ上ではパソコンの初歩的な操作のみで目的が達成できるという利点がある。もともとGUI環境が整備されているため、ユーザーは難しい操作を覚える必要がないのである。実際システムの裏側では複雑な処理をしているのであるが、それを考える必要はない。あたかも目の前のマシンですべての処理を行っているように感じるのである。

しかし、ユーザーの操作性をより簡単に、確実にすなわち誤操作対策を追求すればするほどシステム構築が大変になってくる。要するに、ユーザーの立場と製作者の立場は両端に位置するのである。実験モデルでも、どこまでのレベルで構築していくかというところが大きなポイントとなった。マウス操作は最低限できなければならない。最小限の文字入力操作は要求される。それさえもできないユーザーに対して、どのような環境を提供すればよいかという点が、開発上の分岐点になるのである。実際、ビジネス環境でのシステム・インテグレータは、優れたGUI環境を如何に提供できるかがソフトの優劣の一つの価値判断になっているため苦慮するのである。教育現場におけるシステムもこの点は同様である。

## 9. 3層構造データベースの教育的利用への適用範囲と問題点

### 9.1. 教育的利用への適用範囲

WebSystem型3階層クライアント/サーバシステムは、利用者側から見た場合、そのGUI環境を基礎として運営されるという点で優れたシステムであると言える。従来型のデータベース・アクセスが、簡単な入力作業とマウス操作のみで目的が達成できるようになるのである。

教育分野への応用という観点で、以下のものを列挙する。

- 教材データベース

授業で利用される教材を内外から収集しデータベース化する。教育目的での閲覧を可能にすることでその利用価値は大きい。文献、地図、写真、絵画等、学生や教員が学習、研究する上での一助になる。

- 講義ノートデータベース

講義ノートをデータベース化し、いつでも閲覧できるように学生からの便宜を図る。講義に関連した教材等は教材データベースとリンクさせる。授業を欠席した学生が後日自習したり、授業の進捗状況を把握することができる。また、授業に出席した学生にとって復習や整理の資料ともなる。

- 図書情報検索システム

学内の雑誌目録データベースの検索等に3層構造モデルは最適である。その他貸し出し状況、開館・閉館の日時連絡表示。学外図書情報とのリンク等を盛り込んだシステムが構築できる。

- 学務情報システム

シラバス、授業時間割、学籍名籍等の静的なデータベースや履修登録状況、個人成績取得状況等のセキュリティチェックをかけた動的データベースを組み込み、Web上でいつでも確認できる環境を用意できる。また、月中や年中行事、休講連絡、会議の議案や開催日予定表の表示等、掲示板的要素を整備できる。施設配置図などのマップは、クリック処理により詳細な案内内容を表示させる。新入生とうには有効であろう。

- 就職情報検索システム

就職指導室が有する企業情報を業種・職種・企業名・地域等で学生が検索する。ヘッドライン的な一覧表形式の内容とフォーム化された詳細情報を使い分けることで密度の濃い情報を提示することができるであろう。データにその企業のホームページへのリンクを貼ることで、利用者は就職課で載せきれない UP TO DATE な情報をも取得することが可能となる。

- 教員情報システム

教員の個人記録（業績や研究活動記録、顔写真、連絡先住所等）のデータベース化。学生だけでなく教員との間でも、互いを認識することで、情報交換やコラボレーション的に共同研究を進める契機にもなるであろう。

#### • 学生情報システム

教職員側で主に利用すると思われるが、学生の個人記録閲覧（公開可能な限定的情報—顔写真、氏名、所属学科、クラブ活動等）。単位取得状況の表示、学科生一覧等のシステム化。学生を指導する上での活用が期待できる。

以上のようなものが教育分野への適用になじむと思われる。上記に掲げたものは、一挙に用意できるものではないが、既存のシステムとの調和を図りながら合目的に暫時移行していけばよいのではないか。学生や教員が、内外から情報を得るための一つのキーワードとしてこれらのシステムが運営されれば、システム化する意義は満足されると考える。

## 9.2. 問題点

3層型データベースシステムは、基本的にはクライアント／サーバシステムに他ならない。このシステムを維持管理するための作業は、ネットワーク管理者に高度な知識と労力を強いることになる。そのため、確固たる情報システム部門とスタッフが必要となる。自前でシステムを構築する場合はもとより、アウトソーシングで導入する場合でも同様であり、日常のルーチンワークで生ずるトラブルに対して迅速に対応できるような体制は必須のものである。

WebSystem を中心にシステムを構築した場合、従来のデータベース作成者と Web の知識を有するコンテンツ・クリエイターが必要になる。データベース部分は、このシステム上ではバックグラウンド的な立場に置かれるが、エンドユーザーとのデータの受け渡し操作はコンテンツ・クリエイターが担うことになる。データベース作成能力+コンテンツ作成能力がここでは要求されるのである。最近のオフィス系アプリケーションは、比較的簡単にファイルのHTML化を実現できる環境を提供している。静的なページを作成し次々とリンクを張るという構成のケースでは、このようなツールで十分であろう。しかし、データベースシステムとAPIでやり取りしながらクライアントの要求にリアルタイムに応答していくような動的 Web システムでは、結局のところ両者の知識、技量を併せ持った人材が必要となるのである。今後、Web コンテンツはますますマルチメディア的要素が強まってくると予想される。そうするとサーバには高いポテンシャルを持たせる必要が出てくる。また、伝送経路においても、大量のパケットを高速に流せるラ

インの整備が必要となる。インフラの上位シフト移行を進めながらネットワークの再構築を断続的に行う中で、コストと教育的効果というバランスがうまく保たれる運営を進めなければならない。

構内の限定されたスペースでは、クライアント数はある程度の数量で頭打ちとなるが、ネットワークの定着や学内指導の成果で、ダイヤルアップによる自宅からのアクセス数が増加していきだろう。クライアント数が増加していくと、その分スループットがダウンしていく。このような状況を迎えたときに、如何に快適な環境を維持させていくか今後の大きな問題点となる。

データベースシステムは、データがあって初めてその存在意義や利用価値が生まれる。では、その元となるデータは、いつ、誰が、どのように準備するのかという問題も大きい。システムそのものは器のようなものである。器があっても中身がなければ意味がない。インターネットが多くのボランティア精神のリンクで成り立っている如く、利用するすべての人が、情報の発信者とならなければならないのである。

## 10. まとめ

本稿は、学生や職員のコンピュータ操作による学務処理の効率化を図るため、従来のクライアント/サーバ型データベースシステムモデルから WebSystem による 3層構造データベースシステムへの移行を提案・設計し、履修登録システムと成績入力システムの実験モデルを構築した。今回構築した実験モデルは、未だ概念レベルでしかない。今後実用化へ向けて具体的な形へと発展させる方向で研究を継続していかなければならない。その意味で、API に直接アクセス可能な昨今のデータベース開発用ツールは、今回のテーマをより効率的に実現できる環境を提供してくれている。

今回提案したシステムの利点は、上述したとおりであるが、データベースに直接アクセスするという点で、従来存在したような中間的処理が排除され、利用者側の処理で目的は達成され、同時に当該データベースも出来上がっているというところにある。従来の紙媒体ではなく、電子的に処理された形でその後のデータ加工処理にそのまま進むことができる。システム化がより完成度の高いものであれば、その処理も自動化されて進められる。また、クライアントに Web ブラウザを使用するため、クライアントごとのアプリケーションの導入は不要になる。故に、アプリケーションの導入やバージョンアップのためのコストは必要なくなるのである。しかも、Web ブラウザの操作で作業が進められるため、アプリケーションのための教育も実施する必要はなくなる。このことは、人的・時間的コストの節約になるし、事務的生産性も向上することが基待できるものである。

扱うデータを不正アクセスから保護しなくてはならない場合のセキュリティ措置は、



今後開発の中で詰めなければならない問題である。また、データ量の増大や同時アクセスのクライアント数の増加によるスループットを如何に高めるかという問題は、様々な手段や方法を考えていかなければならない。

今後、上述したような問題の解決を図ると同時に、3階層データベースモデルの適用範囲の拡大を探りながら、既存システムの融和を図り、その移行方法を考えてみたい。

#### 注

- 1) <http://www.monbu.go.jp/singi/chosa> (1998)
- 2) 森本雅博：「エンドユーザーコンピューティングのための情報教育に関する一考察」大手前女子短期大学・大手前栄養文化学院・大手前ビジネス学院研究収録 VOL. 15 (1995)
- 3) <http://www.nw.com/> (1998)
- 4) 前掲1)
- 5) トップマネジメント著：「SQLデータベース入門」LOCUS p.26-27 (1998)
- 6) 望月裕恭著：「SQL Server 6.5 業務システムの設計と開発」SOFTBANK p.131 (1997)

#### その他の参考文献

- Phil Sully：「Modelling the World with Objects」Prentic Hall Intenational Ltd (1995)
- 八代一浩・山本芳彦：「インターネットオフィスの運用」アライドナレッジ (1996)
- 田坂広志：「イントラネット経営—スマート・コーポレーションの創造」生産性出版 (1996)
- 小島政行：「はじめるSQL Server with Access」アプライドナレッジ (1997)
- 大澤文孝：「Webアプリケーション構築ガイド Phase1」SOFTBANK (1997)