

はじめに

—情報メディアセンターの新しい試み—

名古屋情報メディアセンター長 坂東昌子

2005年12月22日、その日は朝から雪がちらつき、夜から大雪になるという予報でした。そんな中、情報メディアセンターの新しい企画であるプレゼンテーションコンテスト（以下、プレコン）が車道コンベンションホールで開催されました。また、12月21日には、豊橋校舎で「情報活用コンテスト」の表彰式が行われました。

今回の企画の紹介をしますと、プレコンはパワーポイントやフラッシュなどのプレゼンテーションツールを活用した、学生のプレゼンテーションを競うコンテストです。一方、情報活用コンテストはWeb ページや Web プログラミング、コンピュータグラフィックス等を使った作品のコンテストです。

こうした新しい試みは、名古屋・豊橋の両情報メディアセンター（以下、メディアセンター）で始まりました。旧情報処理センター（以下、センター）が、2004年度から「情報メディアセンター」と名称を改めて、ようやく新体制での企画が動きはじめました。

時代は今、動画の時代に突入しつつあり、パワーポイントだけでなくビデオ映像も取り入れた教材が少しずつですが、本学でも試みられるようになりました。かつて愛知大学で「情報処理教育」というものが始まった時代に、「ともかくコンピュータに慣れるには、学生達にタイピングの練習を！」を目標に、センターでさまざまな議論を重ねました。ある時、私の知人が使っていた「ベーシック」という簡単なプログラム言語で作ったタイピング練習ソフトがあることを知り、「それをいただけませんか」とお願いして、使わせていただくことからスタートし、そのタイピングソフトのプログラムを愛知大学の事情にあわせての改良を重ね、試行錯誤の結果センターの相談員たちと一緒に愛大版「タイピングソフト」を開発し、完成させたことを思い出します。あの頃は、どこの大学でも情報処理教育を手探りでやっていた時代で、皆で知恵を出し合い支えあって授業を進めていました。そもそも、情報は独占するものではなく、共有するためにあります。ひょっとしたら情報というものの原点には、黎明期のセンターにあったような精神がみなぎっていたのかもしれない。

当時は、高価なコンピュータを購入するのは大変なことでした。コンピュータがどうしても欲しいといって、親に頼んで「借用証」を書いて買ってもらった学生もいましたが、ほとんどの学生はコンピュータを使いたければセンターに行くしかなかった時代でした。センターには当時から学生の助けも必要ということで、「相談員」制度がありましたが、相談員達は、自分の仕事の時間帯でもセンターにやってきました。そんな相談員の中から、高度な情報処理技術を身につけて、今もそれを仕事にしているプロが何人かおられます。当時のセンターにはいつも人が集まり、新しい技術知識の交換やソフトの評価を議論する雰囲気がありました。こういう集団の中で、当時の情報処理教育の一翼を担ったタイピングソフトのような作品が生まれたのです。あの頃、学生のみなさんのタイピン

グの腕を上げようと、毎年タイピングコンテストを開催してにぎやかだったのを記憶しています。

月日がたち、パソコンが手軽に買える時代になり、パソコンを使うためだけにセンターに学生が集まる時代ではなくなりました。そして、以前ほど教員も学生もセンターに自然に集まるという時代ではなくなっただいぶん経ちます。

さて、プレコン・情報活用コンテストの話に戻りますが、これらのコンテストの素材を作るためには、動画などを入れようとすると、ちょっと高度なソフトや機材が必要になってきます。幸い、メディアセンターには現在みなさんがコンテストの素材を作るためのソフトや機材が揃っています。今回のプレコン・情報活用コンテストを契機に、メディアセンターが再び人の集まる場として意義のある時代になったのかもしれない。そうすると、センターに人が集っていた時代のような、活気あるメディアセンターが戻ってくるということにもつながります。人が集まれば、どうしたらうまいプレゼンや作品ができるか、といった交流が始まり、交流が始まるとさらに高い目標に向かってみんなの中に共同作業が始まります。人が集まることこそ、最大の創造活動のための条件なのです。ですから、できるだけ「ちょっとよっていこうかな」と思わせるような環境が必要です。もう1つ、1998年に名古屋校舎のセンターが名古屋図書館の上の階というずいぶんと奥まった場所に移動してしまいました。こうなると、学生が授業の帰りにちょっと立ち寄るには遠いのです。図書館の上の階にあるという構造は、残念ながら人が集まりやすい環境ではありません。それでも、何らかのモチベーションがあれば人は集まるでしょう。今、私が思うのは、人が集まり、交流が始まり、そして共同作業によって、メディアを駆使した教材や教育研究ができる環境づくりです。

プレコン・情報活用コンテストではたくさんの学生達が工夫を凝らして、発表してくれました。そして、プレコンに至っては雪の日にもかかわらず、熱気あふれる発表をしてくれました。採点を引き受けてくださった皆様のご協力に感謝するとともに、2006年度は、もっと持続的な取り組みにしていきたいと思っています。ゼミやサークルの皆さんが、たくさん応募して活躍してくれることを願っています。

センターの中にあるメディア教育開発室は、プレコン・情報活用コンテストをはじめとした活動の準備ができる中心的な役割を担っていきたいと思っています。そして、多くの皆さんの目にふれるだけでなく、今まで以上に知的好奇心や目標設定への意識を高揚できるメディアセンターを目指して、今後ますますの努力を続けていきたいと思っています。

目 次

はじめに 名古屋情報メディアセンター長 坂東 昌子

1. 論 文

授業評価システムの学内開発と運用への取り組み.....	龍 昌治.....	1
授業評価システム開発におけるオープンソース・ソフトウェアおよび フリー・ソフトウェアの利用.....	湯川 治敏.....	11
授業評価システムにおけるオブジェクト指向の設計について.....	蔣 湧.....	25
中国大陸部におけるマルチメディア支援教授.....	任 景波.....	49

2. COM コーナ

E-learning 教材の学習効果を確認するための演習問題の作成 —「インターネット情報論の基礎」章末確認問題—.....	土橋 喜.....	67
---	-----------	----

3. センターだより

1. 情報メディアセンターにおける委員会活動.....	111
2. 情報メディアセンター主催行事.....	113
3. 情報メディアセンター運営会議構成員.....	120
4. 愛知大学におけるコンピュータウィルスの動向及び対策について(4) ～トロイの木馬編～.....	122
5. 編集後記.....	124

原稿募集要項

1. 論文

授業評価システムの学内開発と運用への取り組み

龍 昌治*・蔣 湧**・湯川治敏**

*愛知大学短期大学部 **愛知大学経済学部

1. 学内開発の経緯

大学等における教育改善運動であるFD（Faculty Development）活動では、その一環として学生による授業評価アンケートを行っている。本学においても、2001年度からすべての学部において、取り組まれてきた。

しかしながら、従来の方式による授業評価アンケートは、光学読取式マークシート（OMR）を用いているとはいえ、対象となる授業数や学生数の多さ、また授業中に行うアンケート用紙の配布や回収の時間的なロス、さらには回収してからの読取・集計・公表の膨大な事務作業の煩雑さなどから、個々の教員にとっても、また学生や担当する事務職員にとっても負担感が大きい。これらに要する時間および金額的なトータルコストと、本来の授業改善活動への寄与というメリットの比較から、OMRによるアンケートシステムについては早期の見直しが求められてきていた。

これを受けて、2003年度末には、FD委員会ではWebを利用したシステムの開発を決定した。当時、事務局側ですすめられていた授業シラバスのWebシステム化と合わせて検討・開発することになり、学内理事会のもとに、事務局長をリーダーにしたプロジェクトが組織された。授業評価チームは、シラバスチームと連携しつつ、FD委員会からの開発要請を取りまとめ、Webによる授業評価アンケートシステムを設計・開発することになった。

本報告では、授業評価チームの設計・製作段階における取り組みと問題点、特に学内チームによる実運用システム開発・運用の課題を報告する。

2. プロジェクトチーム体制

プロジェクトは、シラバスチームと授業評価チームで構成された。シラバスチームは、教務事務の担当者らを中心に構成され、実際の開発は外部のシステム会社に委託した。授業評価チームは、学内の教員3名で構成し、さらにゼミ学生3名をくわえて、常時6名で設計と開発を担当した。運用テストなどの大量データ投入を要する工程では、臨時にアルバイト学生らの応援を求めた。

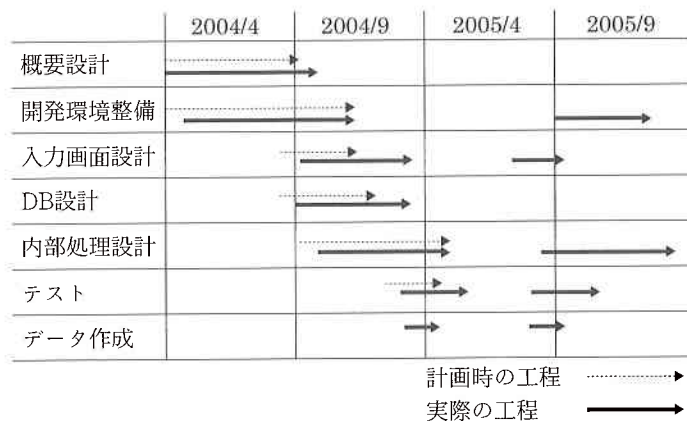


図1 工程スケジュール

2.1. スケジュール

2004年4月に発足したプロジェクトは、1年間の開発期間を経て、2005年4月に本稼動する予定であった。一方で授業評価アンケートの内容を検討するFD委員会では、OMRによるアンケート内容を単純にWeb上に置き換えるのではなく、毎時間の授業ごとに手軽にアンケートを行えるシステムを求めている。具体的には、4月に初期アンケート（2-3週間）、5月から6月に中期アンケート（8-10週間）、7月に後期アンケート（2週間）を行い、担当教員は随時アンケートの結果を参照する。このうえで、8月および9月の集中講義期間終了を待って、10月には春学期の全授業科目の集計を公表することが必要とされた。

これに向けて、開発ではシステム全体の設計と同時に、OMRでのアンケートにはなかった毎回のアンケートと随時の集計、また全授業科目の一括集計という、複数の開発案件をとりまとめることとなった。さらに、Webのみではアンケート回答への意識付けに欠けるために、携帯電話を用いた回答システムが必要とされた。通常システム開発では、詳細な設計仕様を机上で検討し、確定させた上で、システム化の範囲や実施体制をつめていく。しかしながら今回の開発スケジュールと体制からは、いくつかのプロトタイプを元にしたスパイラル型の開発手法と工程を組むことを余儀なくされた。

実際の開発場面においては、仕様の確定に予想以上の時間がかかると同時に、開発にはいつからの仕様変更（手戻り）作業も多く発生し、工程に遅れが生じている。内部処理設計では、春学期の集計処理時に大幅な追加修正が発生した。また計画時には想定していなかったデータ投入に、予想外の時間と工数を必要とし、管理用プログラムを追加開発するなど、大幅な工数増加となっている。

2.2. プロジェクト体制

授業評価システムの開発にあたっては、学内の教員と学生らによる共同開発体制を目指した。これは、教員ら自身の情報分野における開発経験をつむこと、学生らにシステム開発の実際を経験させることにねらいがあった。今回、開発を担当したメンバーの体制は、図のとおりである。

DB設計 画面デザイン設計 DBプログラミング	教員 1 名 学生 2 名
開発環境設計整備 内部処理プログラミング 携帯インタフェース	教員 1 名 学生 1 名
工程管理・システム環境・ テスト・データ整理	教員 1 名 アルバイト学生

図 2 開発体制と分担

学生メンバーの担当と配置は、プロジェクト発足時に公募した学生本人の希望を中心に、所属する卒業研究ゼミでの演習や卒業後の進路をもとに、教員らで検討した。なお、学生メンバーのうち、1名（DBプログラミング担当）は4年次生であり、2名は3年次生（2004年度）であったため、2005年4月の稼動時以降の運用や開発は、5名で担当した。

予想された開発規模に対して、仕様検討や設計開発、システム運用を担当するメンバーが少なく、技術的な相互補完や情報交換のためにも、開発メンバーの定例ミーティングは不可欠であった。反面、教員・学生とも日々の授業等もあるため、メンバーそろってのミーティングは、毎週金曜日の夕方以降になり、開発時間の多くは、土日や夏休み等の長期休暇中にならざるを得なかった。この点が、次に指摘する問題発生の原因となることが多かった。

3. 開発上の問題点

開発を進める上で、生じた問題点について、ここでは学内開発における特徴的な問題点をとりあげて、その原因や解決策を考察する。

3.1. チームの指揮命令・決定・報告承認

チームでの開発では、チームワークによる相互協力体制とともに、チーム全体の意思を決定し、プロジェクトを推進する求心力が欠かせない。しかしながら、教員同士、また教員と学生というチーム編成であったために、仕様上の決定事項は別にして、チームとしての決定事項があいまいなものになりがちであったことは否めない。学生らの勤務時間管理や各種支払い業務の多くは、担当事務課にて行ったものの、不規則になりがちな出勤簿への記入管理などは、チーム内での管理が必要であった。いわゆる報告・承認事項が明確に処理されず、担当事務課やメンバーにも多くの負担を強いることとなった。

これらの解決のため、チームメンバー全員参加による定例ミーティングを毎週実施し、情報交換に努めた。また担当事務課を含めたメーリングリストとグループウェアによる連絡や報告を徹底し、ミーティング議事録による報告を行った。なかでもグループウェアは、安価な自作パソコンとフリーウェアを用いて構築し、多くの開発ドキュメント等のファイルサーバーとしての役割も果たした。自作パソコンや Linux ベースのシステム構築を通じて、学生メンバーらの士気や技術向上にも有効であった。

3.2. スケジュールと目標管理

遅れがちなスケジュール管理は、当初、ミーティングのたびに口頭で確認し、工程表により管理していた。仕様設計工程では、大きな問題にはならなかったが、プログラミング工程では、それぞれが担当する作業に忙殺され、授業等との調整で遅れが目立つようになった。少ないチーム内での協力はもとより、チーム外からの応援が見込めないため、遅延には有効な対策がとることができなかった。

結果的に、当初スケジュールのうち、2005年3月完了の予定であった内部集計プログラムは、開発そのものをアンケート終了後の8月に延期せざるをえなくなった。くわえて、事務職員による代行入力や教務データの一括投入などの管理プログラムの追加開発などが必要となり、開発期間は6ヶ月の大幅遅延となった。

スケジュール管理と開発目標（稼動時期や仕様等）管理は、内部開発における最大の問題点であろう。専任の開発スタッフが確保できず、開発メンバー自身の授業等外部阻害要因などのほか、内部同士ゆえの安易な遅延や目標仕様の変更などが発生しやすい。スケジュールや最終的な詳細仕様決定は、開発チームとは別の組織体制で常に監視するとともに、仕様確認書や仕様変更書などのドキュメントにより確認する体制を確立することが不可欠であるといえよう。

本開発で、シラバスチームも含めたプロジェクト会議が有効に機能しなかったのは、会議メンバーに、システム開発の管理業務に精通あるいは重要事項（マイルストーン）として指摘できる要員がいなかったこと、アンケート方法などの仕様決定をすべきFD委員会で委員交代があり、詳細仕様の決定や確認が開発チームに委ねられたことによる。直接の開発メンバーだけではなく、チーム外に工程管理の要員をおく、あるいは工程管理のみを外部委託するなどの方策が必要であろう。

3.3. 品質管理

システムの品質は、個々のプログラム品質の延長にある。通常のウォーターフォール型開発では、個々のモジュールやサブシステムごとに単体テストを行い、品質を作りこむことができる。本システム開発で採用したプロトタイプ型開発では、仕様検討とともに設計開発が進行するため、最後の統合テストで初めて本格的なテストとなる。

統合テストで問題になった品質面は、主として画面表示の用語などのユーザインタフェース部分であり、プロトタイプでも十分発見対策できるものも多かった。プロトタイプ開発時点での品質面への配慮が低く、結果的に多くの手戻り作業となってしまったといえる。

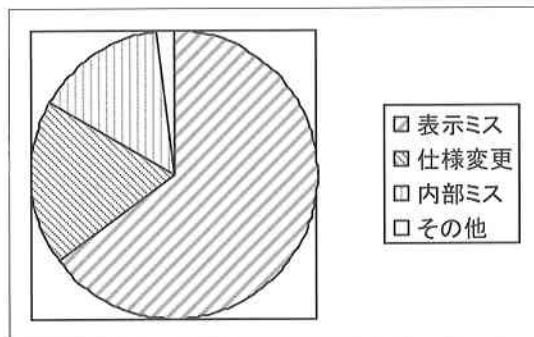


図3 発生バグ内訳（比）

統合テスト工程や本稼動直後の障害・品質管理は、バグ管理表で行った。その集計結果では、画面インタフェース関係など表示系で全48件中31件（65%）、仕様変更につながるもの9件（19%）、内部処理ミス7件（15%）、その他1件となっている。

システムの品質向上には、プログラマ個人の力量とともに、チーム全体での品質意識の徹底が必要である。本開発のような学内メンバーによるチーム編成では、限界とも言えるが、品質に関する学習会を行うなどのほか、仕様検討段階から総合的なバグ管理を行い、バグ発生曲線など既知の品質管理手法を用いて、品質意識の徹底が必要であろう。

3.4. ドキュメント管理

通常のシステム開発では、多くの文書（ドキュメント）が作成され、保管される。口頭での会議や確認事項を、最大もらさずに記録し、決定事項を明確にすることが極めて重要であるからである。本開発でもミーティング議事録やFD委員会などの確認事項は、その都度、文書化し保管されている。ユースケースなどの基本設計ドキュメントは、毎回のミーティングのたびに配布され、確認されてきた。

一方で先のスケジュール管理などと同じく、詳細仕様などの確定仕様の多くは、プロトタイプによる電子ファイルとして記録されていた。このため、担当者の思い込みによる安易な変更や確認漏れが発生した。基本設計ドキュメント類も、ファイルサーバー上の共用エリアに保管されているものと、担当者の手元のバージョンが異なるなど、文書管理の不徹底が散見された。

原因としては、開発場所が担当者の居室や学内の情報メディアセンタなど特定できず、時によって分散するなどして、これらのドキュメントが統一管理できなかつたことがあげられる。安易に共用電子ファイルとしたことも、改訂の責任やバージョン管理をあいまいにしまったことは否めない。学内開発では、チーム全員が一室に集まったの開発は困難であり、電子メールやグループウェアなどドキュメントの電子化は、きわめて有効ではあるが、ドキュメント管理のルールやその運用手順については、共通認識とともに徹底を図る必要がある。

4. 運用上の問題点

2005年4月にアンケート入力部が稼働し、つづいて8月にはアンケート集計部が稼働している。アンケート期間の設定や多くの非常勤教員らの代行設定入力などは、作成した管理システムによりアンケート事務担当者で可能となっている。しかしシステムの運用は、小規模な改訂作業やサーバーシステム操作をともなうことから、引き続き開発チームの教員メンバーが担っている。システムの開発とは別に、運用上の問題点と解決策について考察する。

4.1. データ投入

本システムでは、本学の全授業科目を対象にしている。このためその基礎データは、次表に示すとおり、膨大な件数となっている。これらのデータは、教務・学生データそのものであり、毎学期ごとの授業科目やクラス設定、学籍異動や履修届・変更などを反映していく必要があった。授業シラバスの基礎データと共通化し、同一データベースを参照することも考えられたが、各々のシステムの独立性やデータの安全を考慮して、毎学期ごとに教務データベースから必要項目のみを抽出して、個別にデータを投入することにした。

しかしながら、6学部1短大の教務データは、きわめて複雑で、変更や訂正も多い。学生らの履修届けが確定する時期が、5月と11月という学期中盤であったこともあり、学期初めに予定されていた初期アンケートは、本システムでの運用をあきらめざるを得なかった。またデータ投入作業自体が、授業期間中となるため、データ整理などの作業時間が確保しづらくなっている。

基本的なデータは、学内関係部署の協力により、教務データベースから整形の上、汎用的なCSV形式で抽出している。これをもとに、本システムのデータ形式にあわせた重複データ等の確認等を行っている。主な作業確認項目は次のとおりである。

- ・週複数時間開講クラスデータの整理統合
- ・集中講義クラスの日付設定
- ・文字コード変換統一

結果的にこれらのデータ投入には、予想外の工数を必要としている。特に秋学期については、春学期分のデータとの整合性を取る必要があることが判明した。データの誤りや重複をさけ、正規性を確

表1 主な投入データ件数 (2005年度春学期分)

データ	件数	備考
開講科目数	6,481 件	
学生履修科目数	44,768 件	(アンケート回答最大見込み数)
学生数	10,806 件	
教員数	978 件	
アンケートタイプ	42 種	

保する作業は、当面、データ構造に精通した担当者の手作業に頼らざるを得ない。約7日を要しているこの作業をシステム化していくことが、早急の課題となっている。

4.2. 認証システム

本アンケートシステムでは、アンケート回答者の本人確認のために、ユーザID（学籍番号）とパスワードによる認証を用いている。無記名アンケートのために、ユーザIDは認証キーとしてのみ使用し、アンケート結果の保存に際しては、学籍番号下4桁を消去している。認証システムは、学内の情報メディアセンタのLDAP（Lightweight Directory Access Protocol）システムを使用しており、これは授業シラバスなどの教務システム（UNIVERSAL PASSPORT）やキャリア支援システムと同一である。

本運用になってから、この認証が正常に行えない（授業評価アンケートに回答できない）ケースが多発した。一部の校舎に集中して発生する傾向も見られ、学生をはじめ利用者には大きな影響を及ぼす結果となってしまった。情報メディアセンタ等の学生窓口では、対応方法が分からず混乱を招いたほか、窓口相談に来ない潜在的な未回答者がいることが予想され、これがアンケート回答率の低下を招いた恐れは強い。

原因としては、次の2点の可能性を検討し、検証と対策をすすめている。

(1) 認証技術の理解不足

LDAPやkerberos認証技術は、UNIXやMS-Windowsの標準的な認証システムとして普及しているものの、その実装技術は必ずしも確立しているとはいえない。特にUNIXとWindowsが混在するシステムでは、相互認証の連携に不安が残る。認証サーバとクライアント（授業評価システム）とのシステム時間同期がとれていないと認証できないことも判明している。一定期間パスワードを変更していないと認証できないケースも多く報告されている。

これらの技術的仕様の理解不足は、開発経験のない開発チームでは、事象の発生するたびに原因きりわけと推定・対策に追われることになってしまった。ほぼ同時期に開発運用したシラバスシステムや図書館システムを担当した開発企業では、社内の技術蓄積や情報収集で対応している。先進的な技術を採用するには、その技術の検証を十分に行う必要がある。

kerberos：暗号による認証方式の一つ

(2) 学内認証システムの理解不足

先の技術理解不足にくわえて、学内システム自体の調査理解が不足していた点も大きいと思われる。学内の認証システムは、分散する校舎にあわせて、階層構造で構築されている。本システムでは、開発チームメンバーが在籍している校舎の認証サーバで認証しているため、他校舎所属のユーザ認証に時間差が発生していることが予想される。

階層的な認証システムは、大規模なシステムでは当然のシステムともいえるが、開発チームがこれを認識したのは、運用開始後8ヶ月を過ぎた時点であった。これは事前の調査不足と認証管理局との情報交換不足であり、プロジェクト会議での検証不足であろう。

いずれの原因も引き続き検証と対策を進めている段階であり、予断を許さないが、システム運用に関わる重要事項であり、専任のシステム運用要員の確保などとともに、今後の体制を確立していく必要がある。

5. 評価と今後の課題

5.1. システム開発の評価

システムの開発は、稼動すれば終了ではない。先に述べたとおり、本システムはまだ多くの問題点を抱えており、開発自体も継続中である。なかでも運用まわりの問題点は早急に原因を特定し解決を図らなければならない。

授業評価アンケートとしては、学生らによる回答数が少なく、システム利用にむけての広報不足や、システム自体の改善も必要である。そのような中でも、春学期で延べ7,999件、1,100名あまりの学生回答があった。回答学生が2005年度入学生（1年次生）に偏りが見られるものの、OMRアンケートではありえなかった授業時間外や夜間のアンケート回答も多く（全7,999件中、17時から8時の回答は3,009件 38%）、Webシステムの利用率は確実に向上していくと思われる。

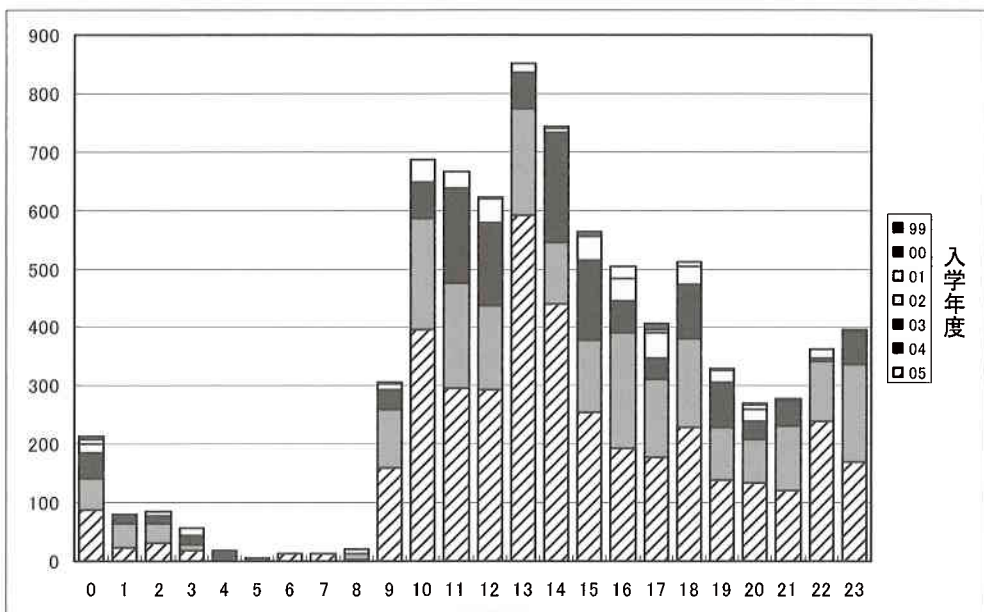


図4 時間帯別アンケート回答件数 (2005年度春学期)

一方で、教員と学生らによる協働のシステム開発では、多くの技術的な蓄積を得ることができ、文系大学ではややもすれば薄れがちなチームワークの大切さや楽しさを感じさせてくれた。情報分野を専攻する教員や学生にとっても、研究室や実習室の中ではなく、ユーザ数10,000人規模の実運用システムを構築・運用する機会は、あまりない。今回システム開発に採用したJavaやTomcatなどのシ

システム技術は、社会でも利用が始まったばかりの先端技術である。狭い研究室の中で、限られた条件の中での研究や学習ばかりではなく、より現実的なレベルで先端技術の学習や実装研究の機会を得たことは、大変に貴重でもある。

数名のチームで、1年間以上にわたって活動を継続することも、大学の中では得がたい経験であった。教員同士、また学生らとも対等な意見交換や情報交換を行うなかで、教員・学生それぞれに学ぶことは多かった。先端技術の調査収集などで、学生メンバーの果たした功績は大きく、また画面デザインなどでも独創的な設計力・提案力を発揮したことは、今後の学習や社会活動の中で、大きな力となるだろう。

5.2. 今後の課題

システム自体の課題にくわえて、学内開発と運用に関しては、いくつかの教訓を得た。本システムの継続的な開発や運用に関しても、引き続き検討課題としていきたい。

まず、学内開発とはいえ、実運用を目指すシステムであれば、システム開発企業と同等の管理体制が必要である。なかでも工程管理と仕様（目標）管理は、開発チームとは別にし、できるならば学外企業のスタッフに委託する、などが考えられる。本システム開発においても、学内に設置したサーバ機器（Webサーバ、DBサーバ）の管理運用は、学外企業に委託し、日常的なデータバックアップやセキュリティ対策はリモート管理を実現している。同様にシステム開発においても、管理業務を委託することは十分可能であり、必要であろう。

第2に、システム運用の体制と責任分担を明確にしておく。教務データなど個人情報に関わるデータも多く、システムの安全性や可用性を確保するのは、システム運用の専門的な知識技術が必要である。委員会主体の体制では、委員交代とともに引継ぎができないことがありうる。一方でシステムは常に監視し運用していかなければならず、開発を担当した教員や学生が、いつまでも運用に携わることも、事実上難しい。たとえば、開発を終えたシステムを、学外企業に一括して管理委託し、必要に応じて委託企業から開発担当者へ再委託をする方式を検討したい。

本システム開発と通じて、我々が学んだことはきわめて大きい。このような機会を与えていただいた学内関係各位に感謝するとともに、引き続き、学生らとも共同しつつ情報システムの研究や学習を行っていきたい。

授業評価システム開発におけるオープンソース・ソフトウェアおよび フリー・ソフトウェアの利用

湯川治敏*・蔣 湧*・龍 昌治**

*愛知大学経済学部 **愛知大学短期大学部

1. はじめに

2004年4月、愛知大学学内理事会に於いて授業評価システムプロジェクトの発足が正式に認められた。このプロジェクトは、これまで紙ベースで行ってきた授業評価アンケートを Web ベースで行うためのシステム開発プロジェクトであり、その大きな特徴として中心的な開発メンバーが本学教員3名と学生3名で構成されていた点が挙げられる。また、もう一つの特徴として開発環境および実行環境をできる限りオープンソース・ソフトウェアあるいはフリー・ソフトウェアを利用しているという点である。本稿では2点目の、特に開発環境におけるオープンソース・ソフトウェアおよびフリー・ソフトウェアの利用について、その選定や利用方法、設定等について解説するが、ある種の備忘録的な内容になってしまうであろう事をあらかじめお断りしておく。

2. オープンソース・ソフトウェア、フリー・ソフトウェアとは？

「オープンソース・ソフトウェア」という言葉はコンピュータ関連の分野ではすでに定着しつつあるが正確な定義については、Open Source Initiative (OSI) によって1998年に定義された[3]。それによると、「オープンソース」とは単にソースコードを得られるだけでなく、自由な再配布を認める事や個人、グループに対しての差別を行わない事等を含む10項目によって定義されている。しかし、これよりもっと以前、emacsに代表されるGNU（「グニユー」と発音する）プロジェクトの主宰者であるRichard M. Stallmanによって1985年に発足したFree Software Foundation (FSF) によると、「フリー・ソフトウェア」は、ユーザが実行する自由、プログラムを書き換える自由、無料あるいは有料でコピーを再配布する自由などの4項目によって定義している[5]。一見、これらは同じ事のように思われるがRichard M. Stallmanによれば「オープンソース運動にとって、ソフトウェアがオープンソースであるべきか否かという問題は、倫理的な問題ではなく、実用的な問題である」としている。また、「オープンソースは開発の方法論であって、フリーソフトウェアは社会運動である」という他人の言に同調している。一方、FSFによる「フリーソフトウェア」の定義を知らない人々にとって

導入した。実験サーバの導入目的は実行環境にシステムを移行する前のまさしく実験運用と実行環境が本格稼働した後にシステムの修正を行う際、実行環境を止めずにほぼ同じ環境で実験をした後にシステムの入替えをする必要があったためである。実験サーバの環境は開発環境のシステムソフトウェア環境と全く同じである。

開発環境および実験環境のシステム・ソフトウェアを以下に示す。

開発／実験サーバ環境：

OS：Fedora Core

Web Server：Apache 2.0.52

JSP コンテナ：Tomcat 5.0.28

データベース：PostgreSQL 7.4.6

さらに開発環境では以下の開発用ソフトウェア群もインストールした。

開発用ソフトウェア：

統合開発環境（SDK）：Eclipse

データベースツール：phpPgAdmin, PgAdmin

グラフィックライブラリ：JFreeChart

3.3 その他開発段階で用いたソフトウェアについて

実行環境、開発環境以外で用いたソフトウェアの中にはグループウェア、負荷テストツールなどがある。

グループウェアは以下に示す^(株)トップマネジメントサービスおよび^(株)システム・コンサルタンツが無償提供する La!Cooda WIZ を用いた。このグループウェア用に以下のマシンを用意した。

グループウェア用マシン環境：

OS：Windows 2003 Server

データベース：mysql 3.23.58

スクリプト言語：php 4.22

ソフトウェアの詳細については後述する。

4. システム開発に用いた各ソフトウェアについて

前述した授業評価システム開発環境および開発には直接関与しないものの補助的なツールとして使用したそれぞれのソフトウェアについての簡単な紹介と具体的な利用方法、設定、問題点等を解説する。

4.1 Red Hat Enterprise Linux 3 (ES)、Fedora Core (オペレーティングシステム)

Linux とは、1991 年に当時フィンランドのヘルシンキ大学在学中であった Linus Torvalds が開発を

開始した OS の中核部分であるカーネルを指す場合が一般的であるが、現在ではそのほかのソフトウェアも併せた OS 全体を指す場合もある。OS 全体を指す場合は GNU/Linux [9] と呼ぶべきであると FSF の Rechart M. Stallman らは主張している。この GNU/Linux にさらに様々なソフトウェアやインストールや設定を容易にするツールなども含めて配布されているものをディストリビューションパッケージと呼び、そのひとつが Red Hat 社が提供する Red Hat Enterprise Linux である。Linux はそもそもフリー・ソフトウェアとして開発されたものであるが、Red Hat Enterprise Linux の場合は保守契約を結ぶことでセキュリティパッチの配布等の情報提供を含むサポートを受けることができる。その必要のない場合は同じ Red Hat 社からサポートは無い代わりに無償で利用できる Fedora Core と呼ばれるディストリビューションを利用可能である。3.1 ですべに述べたように授業評価システムの Web サーバおよびデータベースサーバ用には Red Hat Enterprise Linux を用い、開発・実験環境では Fedora Core を用いた。これは当初から本格運用開始時にはシステムのメンテナンス等は外部業者への委託を検討していたこと。そのためには最もポピュラーなものうちのひとつであり、委託する業者にとってもメンテナンスの実績があるディストリビューションであることが望ましいことなどから総合的に判断した。また、開発・実験サーバには本格運用とできるだけ同じディストリビューションで開発あるいは実験を行うことが望ましいと考えられたため、さらにはコストの削減も考慮し、同じ Red Hat 社と関係のある Fedora プロジェクト [8] で開発している Fedora Core を選んだ。

環境設定としては日本語文字コードを EUC とした。これは、開発開始当時でもそろそろ UTF が使われ始めてきてはいるがまだまだ EUC で作られるシステムが多く、開発コンサルティングを依頼したサーバ納入兼システム保守業者でも UTF を使ったシステム開発はその当時は行っていないとのアドバイスによる。従って、システム内の文字コードはデータベースも含め、全て EUC とした。しかし、開発環境である Fedora Core のデフォルト漢字コードは UTF であり、設定変更をしないままインストールし、そのまま開発を行ったために特にグラフィックスでの日本語表示等でかなり不都合が生じてしまった。このような事を避けるためにも実行環境と開発環境では同じ環境となるよう、特に日本語文字コードについては開発初期段階に於いてどの文字コードを使用していくかについて慎重に選ぶ必要があることを痛感した。

4.2 Apache (Web サービスソフトウェア)

Apache Software Foundation (ASF、Apache ソフトウェア財団) [6] という非営利団体のプロジェクトで開発された Web サービスを行うソフトウェアで、現在世界で最も多く稼働している Web サーバソフトウェアといわれている。ほとんどの Linux ディストリビューションにはデフォルトで組み込まれており、Linux と同じくオープンソースの代表的なソフトウェアのひとつである。Red Hat Enterprise Linux および Fedora Core のディストリビューションにも組み込まれているため、OS インストール時にこのソフトウェアも併せてインストールし、サーバ管理ツールなどでソフトウェアを起動するのみで Web サービスを簡単に開始できる。ただし、授業評価システムでは Java 言語をベースにしたサーバアプリケーションを開発するため、デフォルトの設定のみでは十分ではなく、次の

Tomcat と連携させた Web サービスの運用が必要である。Apache の詳細は Tomcat と併せて次節で解説する。

4.3 Tomcat (JSP (Java Server Pages) コンテナ)

4.3.1 Tomcat とは？

Tomcat は前述の ASF のサブプロジェクトである Jakarta プロジェクトで開発された JSP (Java Server Pages) コンテナであり、Apache 同様オープンソースである。通常の Web サービスの場合はクライアントからのリクエストに対して Web サーバ上にある html ファイルを http プロトコルに従ってレスポンスとして返すだけである。このような Web ページを静的なページという。つまり、どのクライアントからいつ見ても同じページが表示されるという意味であり、この様な場合には Apache のみで十分である。しかし、最近の Web ページはクライアントからある情報を入力してもらい、その情報に応じてレスポンスを変化させる動的なページを扱う Web サイトが非常に多い。いわゆる Web アプリケーションが非常に普及しているが、これを実現する方法には大きく分けてクライアントサイドプログラミングとサーバサイドプログラミングがある。前者は JavaScript などのスクリプトを html のファイルに直接埋め込み、それをクライアントの Web ブラウザが解釈して動的なページを実現する方法である。この方法は非常に簡単に実現でき、また、Web サーバとしても Apache 単体で実現可能である。しかし、データベースと連携したアプリケーションや複雑な処理を行う大規模プログラミングには不向きである。一方、後者はクライアントからの情報を Web サーバで受け取り、サーバ内部で処理を行った後にクライアントに加工した html ファイルを返す方式であり、PHP や Perl などのスクリプト言語を用いた CGI (Common Gateway Interface) などがこれまで用いられてきた。しかし、CGI の利用を許可する事によりシステムのセキュリティーが脆弱になるという指摘や CGI はひとつのリクエストに対してひとつのプロセスを起動するので同時アクセスが多数あると想定される大規模サイトには不向きであるとされている。また、PHP や Perl などのスクリプト言語は本来大規模プログラミングには不向きである。これに対して Tomcat はプログラミング言語として Java を使用できる為、完全なオブジェクト指向プログラミングが可能である事が特徴である。つまり、大規模プログラミングにも耐えられ、また、クライアントからのリクエストに対してはプロセスよりも負荷の軽いスレッドが起動されるだけなのでシステムの負荷も小さく、大規模システムに適している。同じような製品に Microsoft の ASP があるが ASP の場合はプラットフォームが Windows に限られるだけでなく、Web サーバも IIS でなければならないという制約がある。これに対して Tomcat は Linux はもちろん、Windows でも稼働するプラットフォーム非依存であり、Web サーバも Apache 以外のサーバソフトウェアとも連携可能である。また、Tomcat 自体も Web サーバとしての機能を持ち合わせている。さらに大きな特徴は Tomcat は無償利用が可能な点である。今回開発した授業評価システムのシステムとしての一番の特徴はこの Tomcat を用いた点であろう。

4.3.2 Apache との連携と各ソフトウェアのバージョン

前節で述べたとおり、Tomcat は非常に優れたソフトウェアではあるが単体で Web サーバを兼任さ

せた場合には Apache と比較するとその処理能力は十分とは言えない。従って、大規模サイトでは Apache と Tomcat を連携させて動作させる事が一般的である。つまり、通常のリクエストの場合には Apache が対応し、JSP として動作する必要があるときは Tomcat がリクエストを受け付けて処理をする。この連携を行うソフトウェアのことを jk-Connector と呼んでおり Tomcat と併せて Jakarta プロジェクトが提供している。Apache, Tomcat, jk-Connector とも様々なバージョンがあるがこのプロジェクトで用いた各ソフトウェアのバージョンは以下の通りである。また、開発当初から Tomcat の最新バージョンは 5.5.x 系であったが、対応する Java が J2SE5.0 であり、開発スタッフが J2SE1.4.2 を使っていたことから Tomcat のバージョンをあわせた。

Apache : 2.0.46

Tomcat : 5.0.28

jk-Connector : 1.2.6

Java : j2sdk-1.4.2

4.3.2 Apache-Tomcat 連携とその設定

Apache-Tomcat 連携をするために Web サーバ上で以下の設定を行った。

- 1) /etc/httpd/modules ディレクトリに jk-connector のバイナリ・リリースである mod_jk.so をインストール
- 2) /usr/lib/httpd/modules/mod_jk.so から上記 mod_jk.so にシンボリックリンクを張る
- 3) /etc/httpd/conf/httpd.conf の 208 行目に
LoadModule jk_module modules/mod_jk.so を追加
- 4) /etc/httpd/conf/httpd.conf の最下部に下記 IfModule ブロックを追加
<IfModule mod_jk.c>
 JkWorkersFile "/etc/httpd/conf/workers.properties"
 JkLogFile logs/mod_jk.log
 (セキュリティ保護の為、中略)
</IfModule>
- 5) /etc/httpd/conf に下記内容の workers.properties ファイルを追加
worker.list=ajp13
worker.ajp13.lbfactor=1
worker.ajp13.port=8009
worker.ajp13.host=wbces
worker.ajp13.type=ajp13

上記設定の後、Tomcat および Apache を再起動すれば Apache-Tomcat 連携が実現できる。

4.4 PostgreSQL (データベース)

オープンソース・データベースの代表格であり、PostgreSQL[4](ポストグレエスキューエルまたはポストGRESキューエルと読む)という名称は1997年のバージョン6.0からであるが元をたどれば1980年代のUCB(カリフォルニア大学バークレー校)におけるIngresというリレーショナル・データベース研究プロジェクトが発端となって現在に至っている。現在の最新バージョンは8であり、UNIX系OSだけでなくWindowsでも稼働するシステムとなっているが、開発開始当初のバージョンは7であったため、本稼働システムおよび開発環境に於いてはバージョン7.4.6を利用した。PostgreSQLは無償利用可能であるにもかかわらず商用データベースに匹敵する高機能、高性能で知られており、何百万人分ものデータを扱うG(ギガ)バイトクラスのデータベースも構築されているという報告がある。授業評価システムにおける学生の履修データの件数としては、全学生数×各学生の履修科目数×春・秋学期の計約8万件に達し、アンケートの回答数も最大で同数が見込まれる。様々な実績報告などでかなり複雑かつ大規模なシステムも取り扱いが可能であることを知ってはいたが、実際に開発チーム内でこの規模のデータを扱った経験が無かったことから、サンプルデータを投入し、実際にデータベースからの読み込み、書き込みが問題無くできるかどうかは非常に不安であった。しかし、実際には全く問題なく稼働している。ただ、PostgreSQLに関して次の2点の問題があった。

1. OSにバンドルされているRed Hat版PostgreSQLでは簡単な問い合わせに対してもレスポンスが極端に遅い
2. データベースのダンプ、リストアが正常に機能しない場合があった

1については未だに原因不明である。PostgreSQLは前述のように世界中で使われているオープンソース・データベースであり、数々のLinuxディストリビューションでもバンドルされている場合が多い。Red Hat Enterprise Linux 3(ES)の場合も当初はPostgreSQL7.3.8をベースにしたRed Hat版PostgreSQLを利用するつもりであり、システム保守を委託した外部業者からもメンテナンスやサポートも受けられるRed Hat版を勧められた。しかし、実際にサンプルデータ投入後、VIEWからの簡単な問い合わせに対するレスポンスが返るまでに2分強掛かるという考えられない状況であった。データベースのチューニングを行わなければ最適なパフォーマンスが得られないとの説明を受けたが、いくらデフォルト設定のままだとしても非常に簡単な問い合わせに2分も掛かるのはチューニング以外の要因が考えられた為、今回はRed Hat版ではなく一般的に使われているPostgreSQL7.4.6を新たにインストールすることとした。こちらの場合はパフォーマンスに全く問題はなく、現在も非Red Hat版が稼働中である。

2の問題の詳細は以下の通りである。システム開発中および本稼働を始めた場合もデータベースは定期的にバックアップを行う。PostgreSQLでは具体的にはpg_dumpというコマンドを用いてデータおよびスキーマをまとめてひとつのファイルとして保存可能である。特に、PostgreSQLではデータ

ベース稼働中でもバックアップのできるホット・バックアップ機能を持っているので非常に便利である。この `pg_dump` は単にバックアップのみに利用するだけではなく、データベースの名称変更や開発マシンから実稼働マシンへデータベースを移動する場合にも頻繁に用いられる。例えば開発段階でスキーマ (TABLE や VIEW、FUNCTION) の変更があった場合にはそれらを反映させるために変更したデータベースをダンプし、移動先でリストアして全く同じ物を再現する。この操作がかなり行われたが、ある時ダンプしたファイルをリストアしても本来あるべき VIEW や FUNCTION が再現されないという現象が発生した。そこで、ダンプしたファイルはテキストファイルであるので内容を調べてみると原因は VIEW や FUNCTION の変更によるものであることが判った。つまり、次節に述べるデータベース・ツールなどによって VIEW や FUNCTION などは非常に簡単に変更でき、その変更もその時点で使われているデータベースでは即反映される。データベースをダンプする際にはそれぞれのスキーマが定義された順にダンプ、リストアされる為、最初の段階では問題ないものの、スキーマが変更された場合はその変更されたスキーマの定義が最後尾、つまり一番最後に定義されたかのようにダンプファイルに記述されてしまう。このことにより、本来の定義順が狂ってしまう。例えばある VIEW b を定義する為に他の VIEW a を参照していたとすると、本来ならばダンプファイルの中では VIEW a、VIEW b の順に定義されていなければならないが、VIEW b の定義後に VIEW a を変更するとダンプファイルの中では VIEW b、VIEW a の順になってしまい、VIEW b を定義する際に必要になる VIEW a が参照できず、結局、VIEW b は未定義のままになってしまう。これは `pg_dump` の仕様なのかバグなのかかわからず、また、コマンドの利用方法が間違っているのかあるいは別の方法があるかも判らなかつた為、解決策としては依存関係が正しく参照されるようダンプファイルを直接編集した後、リストアを行うことで解決した。なお、一度定義順が修正されたダンプファイルをリストアすればその後は通常のダンプ、リストアが行えることを確認した。

今回は問題解決までそれほど時間が掛からず開発に大きな影響を及ぼすことは無かったが、データのダンプ、リストアというデータベースの管理にとっては最も重要な機能のひとつが意図したように動作しない状況に陥った際、オープンソース・ソフトウェアの場合は基本的には自らが問題解決に取り組まざるを得ない。これが商用データベースであればメーカーあるいは販売店等に直接問い合わせることができるのだが、オープンソースの場合はメーリングリストや BBS、提供サイトの FAQ などを参考にするしかない。多分、ほとんどの問題がこのような方法で解決可能であるだろうが、時間的な制約がある場合などはすぐに問題解決ができるかどうかは非常に大きなファクターとなるだろう。今回の問題により改めてオープンソース利用の利点だけでなく、利用の際の注意点、問題解決手段の確保の必要性を学んだ。

4.5 PgAdmin, phpPgAdmin (データベース操作ツール)

PostgreSQL はオープンソースであるにも関わらず、非常に多機能、高性能である事はすでに紹介したが、PostgreSQL 単体だけでは CUI (キャラクター・ユーザ・インターフェイス) のみしか提供していない。つまり、TABLE や VIEW の定義、データ入力などはコマンドラインで入力するか、

SQL 文を書いたファイルを読み込ませることによって行われる。しかし、大規模なシステム構築には GUI (グラフィカル・ユーザ・インターフェイス) によるデータベース操作ツールが望まれる。幸い、PostgreSQL にはいくつかそのようなソフトウェアが存在し、本プロジェクトにおいても PgAdmin および phpPgAdmin の双方を利用した。どちらも似たような操作性であるが一長一短があるので両者について紹介する。

PgAdmin は PostgreSQL の開発チームとは直接関係ないようであるが、phpPgAdmin と共に PostgreSQL 本家の Web サイトに Graphical Clients として紹介されている。現在のバージョンは PgAdmin III であり、Linux を含む UNIX 系 OS だけでなく、Windows、MacOSX でも動作するマルチプラットフォーム対応のアプリケーションである。授業評価プロジェクトチームでも Linux および MacOSX 上で動作させデータベース操作を行った。データの入力や編集作業も GUI ででき、また、スキーマの定義を確認し、変更する事が非常に簡単にできる事が特徴である。ただ、大きな欠点は CSV (Comma Separated Values) ファイルなど外部ファイル読み込み機能が無いことである。これに対して phpPgAdmin は CSV ファイルの読み込み、書き出し機能がついており、Excel ファイル等からのデータ移行が非常に楽であった。ただし、phpPgAdmin は PgAdminIII の様な単独のアプリケーションではなく、php による Web アプリケーションとしてインプリメントされているため、実行するにはあるマシンで Apache などの Web サーバを動かす、スクリプト言語 php を利用できる環境が必要である。これは単独のアプリケーションである PgAdmin と比較すると大きな欠点であるかのようにも思えるがネットワーク上のどこかのマシンで phpPgAdmin を実行可能であり、そのマシンへのログインが可能であればリモートで利用することが可能である。例えば、授業評価のデータベースサーバは学内マシンからのアクセスしか受け付けない設定にしてあるので、学内のどこかのサーバで phpPgAdmin を動かす事により学外からデータベースサーバにアクセスすることも可能となる。同様に PgAdmin ではネットワーク経由でのサーバ接続も可能であるが授業評価データベースサーバは学外からの直接のアクセスを許可していないので学外からの接続はできない。

このようにこの 2 つのツールは相互補完的な機能および、長所、短所があるため両ソフトウェアを使い分ける事で効率よく作業を行うことができた。

4.6 Eclipse (統合開発環境)

Eclipse は IDE (Integrated Development Environment、統合開発環境) と呼ばれるアプリケーション開発ツールのプラットフォームであり、2001 年 11 月に IBM 社がオープンソース・コミュニティ、のちの Eclipse Foundation [7] に寄贈したことからオープンソースとしての開発が始まった。当初からこのソフトウェアには 4000 万ドル相当の価値があるとされてきたが、プログラマー、特に Java 開発者の間で急速に広がり、Java 開発のデファクト・スタンダードと言っても過言ではない存在となった。基本的な機能だけ挙げても Java コンパイラ、デバッガ、Ant ビルドツール、CVS ツールなどである。また、様々な機能をプラグインの形で拡張可能であり、Eclipse のプラグインのみを紹介する Web サイトがある程、無数のプラグインが世の中に存在する。もちろんその中には有償のもの

も含まれるが今回のシステム開発では無償利用可能な Tomcat プラグインを用いた。このプラグインを用いることにより、開発マシン上では Apache - Tomcat の連携をする必要が無く、必要な設定さえすれば Eclipse の画面から Tomcat を単独で起動、再起動、停止を操作できる。Tomcat で動作するアプリケーションをコーディングしてはブラウザで確認、Eclipse に戻って修正後、ブラウザで再確認、という作業が非常に効率良く行えた。また、デバグも非常に使いやすく、デバグ効率も非常に良かったと思われる。もし、Eclipse を利用せずに同じシステムを開発しなければならないとしたら開発に掛かる期間は多分 1.5 倍から 2 倍程度に延びていたのではないかと思われるくらい開発効率向上に貢献したソフトウェアである。ただ、あまりに高機能すぎてデフォルトで提供されているにもかかわらず使い切れなかった機能がある。そのひとつが CVS (Concurrent Versions System) である。これはプログラムやドキュメントのバージョン管理を行う機能であり、いつ誰がどのように変更したかを逐一記録し、「どの時点のプログラムやドキュメントに戻したい」などの要求に応える事ができる。本格的なチーム開発には欠かせない機能である。今回のシステム開発ではエースプログラマがほとんどひとりでコーディングを担当していたため、大きな混乱は無かったが、それでもコードの修正を同時期に複数人が担当することもあったため、本来であれば CVS の機能を用いてコードのバージョン管理をすべきであった。これが今回の Eclipse 利用に関する最大の反省点である。

4.7 JFreeChart (Java グラフィック・クラス・ライブラリ)

JFreeChart[10] はフリーの Java グラフィックス・クラス・ライブラリであり、2D および 3D の非常にきれいなグラフを簡単に生成することができる。様々なグラフを PNG 形式や JPEG 形式で保存することが可能であり、また、JSP や Servlet からの利用も可能である。授業評価システムで用いるグラフは積み上げ棒グラフと、棒グラフ + 折れ線グラフの 2 種類のみであり非常に単純であったが、それらを awt などの Java のグラフィックス・ライブラリを用いて直接書いていたとしたらグラフィックスの部分に非常に多くの時間を割かなければならなかったであろうし、時間を掛けたとしても必ずしも良い物ができるとは限らない。今回、かなり早い段階でこの JFreeChart を見つけることができたことは開発効率の向上に大変役になったと思われる。Java がオブジェクト指向であり、コンポーネントの組み合わせにより現存するソフトウェア資源を有効に活用できることを身をもって体験できたソフトウェアである。

4.8 LalCooda WIZ (グループウェア)

システム開発に直接関係するソフトウェアではないが、プロジェクト全体として非常に有効利用したソフトウェアのひとつがこの LalCooda WIZ[12] である。このソフトウェアは(株)トップマネジメントサービスおよび(株)システム・コンサルタンツが無償提供しているグループウェアであり、以下の機能を提供している。

- ・ファイル共有
- ・スケジュール

- 伝言メモ
- 掲示板
- 回覧板
- 施設予約
- アドレス帳

このうち開発プロジェクトで特に頻繁に用いたのがファイル共有である。プロジェクト構成メンバーの都合上、全員が同じ場所に集まれる機会は概ね1週間に1度だけであった。従って、会わない間の成果物の共有、議事録の配信、事務的な文書のやりとりなど、プロジェクト全体の進行にとって非常に重要な役割を果たした。

4.9 Mailman (メーリングリスト管理システム)

同じような機能を持つソフトウェアにメーリングリストがある。プロジェクトが発足してから2006年1月で約1年半になるが、この間、プロジェクトメンバー内での電子メールのやり取りは1200件を超える。1通のメールを複数人に送るという操作は些細な事かもしれないが、これだけの数の電子メールによって情報共有と情報伝達が必要だった事を考えるとメーリングリストが提供するグループ内への一括送信という機能は非常に大きな省力化につながったのではないかと考えられる。このプロジェクトで用いたメーリングリスト管理システムはGNUプロジェクトで開発されたMailman[2]でありメーリングリストサーバは大学が管理しているサーバを利用して頂いた。

4.10 JMeter (負荷テストツール)

前節のグループウェアと同様、システム開発に直接的には関係しなかったものの、実際にシステムを稼働させる前に必ず行うべきテストのひとつが負荷テストである。授業評価システムは本学学部生および短大生全員を対象としているため、ユーザ数は約1万人である。これらのユーザが一斉にシステムを使う可能性はほとんど考えられないが、50人規模のコンピュータ実習室が3キャンパス併せて9存在するので、その1/3の約150人程度の同時アクセスは可能性として否定できない。実際にそれだけの人数を使って人海戦術による負荷テストも考えられるが、時間と労力のコストが掛かりすぎる。最終的には80名同時アクセスを試行し、問題なく接続できたが、その準備段階として負荷テストツールによって最大150同時アクセスを実験し、認証、ログイン後の表示まで問題なく動作することを確認した。負荷テストツールとしてはApacheのサブプロジェクトであるJakartaプロジェクト(Tomcatと同じである)で開発されたJMeter[1]を用いた。JMeterはJavaアプリケーションとしてインプリメントされているため、プラットフォームを選ばない事、またApache、Tomcatとともにオープンソースであることが特徴である。負荷テストの設定、実行はGUI画面で行うことができ、結果もリアルタイムでグラフ化されるなど非常に使い易く作られている。実際にどれくらいの同時アクセスに耐えることができるのか?あるいは特定の同時アクセス数の場合のレスポンスタイムはどれくらいになるか?など最終的な人海戦術を行う前にある程度システムの信頼性を確認できた事は有益

であった。

4.11 iモード HTML (携帯電話シミュレータ)

授業評価システムではアンケートに回答できる機会をできる限り増やそうとする意図から様々な工夫を行った。そのひとつが携帯電話からの回答を可能にしたことである。この開発途中では実際に各携帯キャリアの電話機を用いてアクセステストを行ったが、それ以前の画面表示テストなどでは携帯電話シミュレータを用いて画面設計や修正などを行った。

5. まとめ

授業評価システムプロジェクトが始動して約1年半が経過した。実際には現在も修正を要する機能、画面等が存在し、プロジェクトからなかなか抜けきれない状態が続いているが、今後もこれまで紹介してきたオープンソース・ソフトウェア、フリー・ソフトウェアを用いての修正作業が続くことであろう。時間の経過とともに、それらのソフトウェアが洗練され、ますます、使いやすくなっていくことを期待するとともに、このプロジェクトで得た経験を今後の研究、教育活動に生かしていきたい。

引用・参考文献、関連 Web サイト

- [1] Apache JMeter, <http://jakarta.apache.org/jmeter/>.
- [2] Mailman, the GNU Mailing List Manager, <http://www.list.org/>.
- [3] Open Source Initiative, <http://www.opensource.org/>.
- [4] PostgreSQL Global Development Group, <http://www.postgresql.org/>.
- [5] Richard M. Stallman、長尾高弘訳、フリー・ソフトウェアと自由な社会 : Richard M. Stallman エッセイ集、株式会社アスキー、2003。
- [6] The Apache Software Foundation, <http://www.apache.org/>.
- [7] The Eclipse Foundation, <http://www.eclipse.org/>
- [8] The Fedora Project, <http://fedora.redhat.com/>.
- [9] The GNU Operating System, <http://www.gnu.org/>.
- [10] The JFree software projects, <http://www.jfree.org/>.
- [11] The Jakarta Project, <http://jakarta.apache.org/>.
- [12] ウェブグループウェア La!cooda WIZ, <http://wiz.syscon.co.jp/>

授業評価システムにおけるオブジェクト指向の設計について

蔣 湧*・湯川治敏*・龍 昌治**

*愛知大学経済学部 **愛知大学短期大学部

はじめに

大学におけるFD活動の一環として、学生により授業評価はアンケートの形で行われていた。今まで紙ベースで実施した授業アンケートをより効率的、正確的、経済的に行うために、Webベースの授業評価システムを開発した。このシステムをWeb Based Class Evaluation Systemと名づけ、本文にはWBCESと簡略する。

WBCESシステムはデータベースサーバー、ウェブサーバーとクライアント（携帯電話を含め）により構成した典型的な3層ウェブアプリケーションである（図1）。サーバーはLinux環境で作動し、

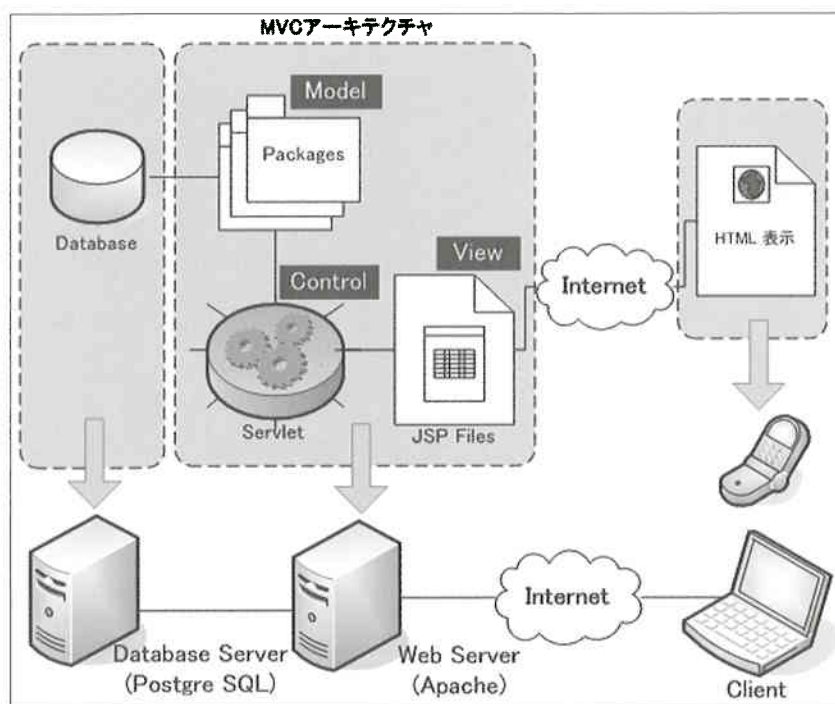


図1 WBCESのシステム構成

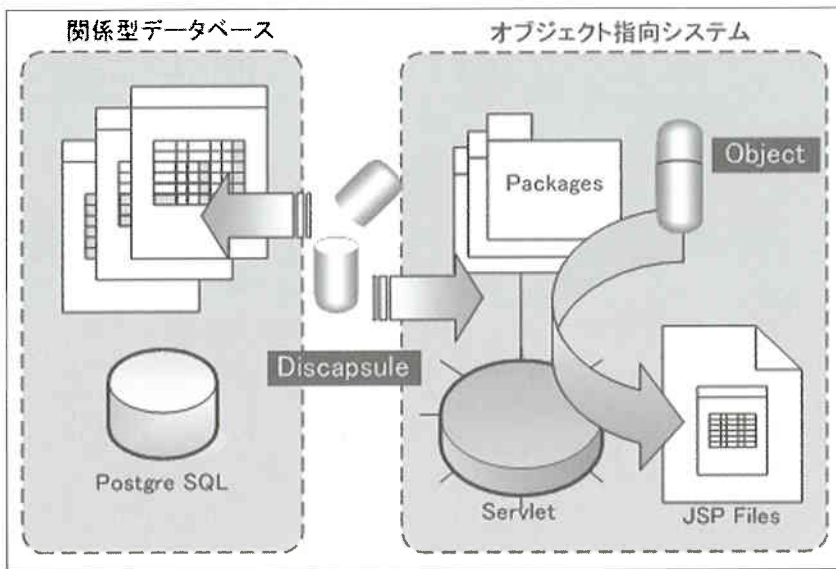


図2 オブジェクトとデータベースの間のデータマッピング作業

ウェブサーバーとして Apache を、データベースサーバーとして PostgreSQL を採用した。システムアーキテクチャに MVC (Model、View、Control) モデルが導入され、JSP (Java Server Pages) により HTML 生成処理のロジックと JavaBeans によりアンケート処理のロジックが分離されたことで、より優れたシステムの保守性とコードの再利用性を目指した。

開発環境については、Eclipse3 を中心に、Tomcat、EclipseUML、PhpAdmin、PgAdmin など Open Source が採択され、高品質、高機能なオブジェクト指向プログラミング環境とデータベース構築・操作環境が無償で実現された。

WBCES は関係型データベースを利用したオブジェクト指向のウェブアプリケーション・システムである。WBCES にもっとも頻繁に行われるのは、オブジェクトとデータベースの間に行われるデータマッピング作業である。たとえば、ユーザー認証時、データベースから抽出したユーザー情報をユーザーオブジェクトに実装する作業、或いは、回答オブジェクトに含まれている回答データを取り出し、関連テーブルに書き込む作業など、様々なオブジェクトがデータベースとアクセスしながら、データの読み取り、書き込みを行う (図2)。これらの作業はウェブアプリケーションにとって、もっとも基本的な作業である。ときには大量なデータベースアクセス作業が集中的に行われるため、システムの負荷と信頼性に影響を与えかねない。

本文では、多くのオブジェクトに共有できるデータベースプログラミングのデザインパターンを提案する。WBCES システムでの実験結果により、データベースプログラムの再利用性が極めて向上し、より優れたシステム信頼性と保守性が得られた。また、このデータベースプログラミングのデザインパターンは汎用性を持つ、広範囲の Java ウェブアプリケーションに利用できる。

本文は次のように構成される。第1節には授業アンケートシステムの概要を述べる。データ構造を含めたデータベースモデルは第2節に説明する。第3節には標準的なデータベースプログラミングを

紹介し、第4節には共有できるデータベースプログラミングデザインパターンを提案する。最後の第5節には、提案したデータベースプログラムパターンの応用事例を紹介する。

1. 授業アンケートシステムの概要

授業評価システムには、学生、教員と管理員、3種類のユーザーがあり、そのうち管理員は数名の教員が兼任することを想定している。全てのユーザーは共通のログイン画面を利用し、ユーザー認証を行う。学生は携帯電話によるアクセスも可能である。認証に成功した場合、それぞれ学生と教員用の画面へ移り、ユーザー権限によりアンケートの実施やアンケートに関わる初期設定などができる(図3)。管理権限を持つ教員の画面には、管理員画面へのログインリンクが自動的に表示される。管理員としての認証が成功した場合、管理員の画面に移る。図3には、学生、教員と管理員、ユーザーごとのシステムと主な機能を表示している。

学生画面には、学生個人の履修科目が学期ごとにリストされている。各々の科目名と共に、該当科目のアンケート実施状態も明示されている。例えば、アンケートの回答期間において、まだ回答していない科目に対し、アンケート状態は「未回答」と明記すると同時に、アンケート回答画面へのリンクも自動的に張られる。科目名をクリックするだけで、アンケートを回答することができる。回答完了後、リンクは自動的に消え、アンケート状態は「回答済み」と変わる。

教員画面には、教員個人の担当科目が学期ごとにリストされている。アンケートの進行状態により画面上適切な内容を表示しながら、対応した機能を提供している。たとえば、アンケート実施日の変

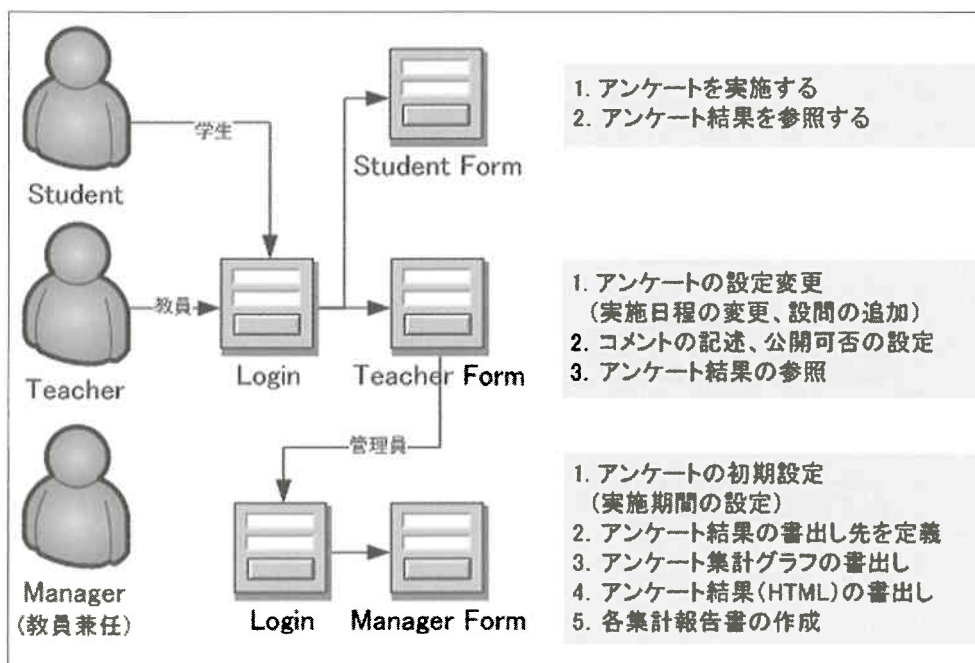


図3 ユーザータイプとシステムの主な機能

更作業と設問の追加作業は、回答期間内には行えない。また、回答期間後、科目ごとのアンケート結果と科目群の平均結果が集計され、図表形式で確認することができる。その結果に基づいて、教員によりコメントを記述し、そのアンケート結果を公表するかどうかについて、教員自ら選択することが可能である。

学生と教員の画面には、今までのアンケート結果を参照するボタンが配置され、本人の履修科目だけではなく、全ての開講科目のアンケート結果を参照することができる。

管理員サイトでは、アンケートの初期設定、集計グラフの書出し、アンケート集計と結果公表に関わる作業が行われる。初期設定には、アンケート実施期間の設定と回答データの保存先の設定が含まれている。設定した項目はデータベースに保存され、アンケート実施の基準データになる。

アンケートの結果報告書はウェブページで表示される。科目ごとの回答結果と科目群の平均値結果が集計され、図表の形で公表する。大量な集計作業を要するため、1ページを表示するまで多くの時間がかかる。ページ表示のレスポンスタイムを最小限に減らすために、集計グラフを予めJPGの形式で一括書き出すツールを管理員サイトに用意した。また、管理員サイトには、各種のアンケート報告書を作成するために、多くの集計関数を用意した。従来の紙ベースのアンケートと比べ、より正確、より多種類の集計データを効率的に得られるようになった。

2. データベースモデル

授業評価のワークフローを踏まえて、関わるデータとデータの性質、さらにデータ間の関連を整理しながら、データベースのデータ構造を探っていく。授業評価に参加するのは学生と教員であり、教員が開講した全ての講義は評価の対象となる。下図のように、授業評価に関わる全てのデータは、「学生」、「教員」、「科目」、この3つの実体データの元から派生される。

まず、科目と担当教員さえあれば、講義を開講することができる。その場合、開講科目名、担当教員、開講学期、開講日と開講形態（集中講義など）は開講データとして生じる。次に開講した講義に対し、学生の履修が現れる。履修データとして、誰がどの講義を履修するかについて記述される。ま

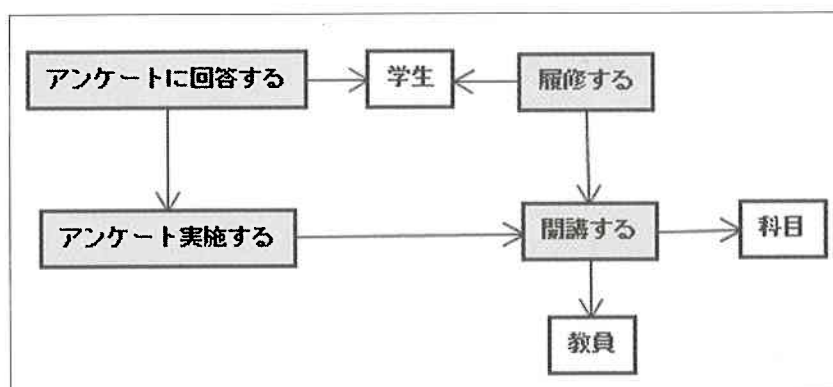


図4 授業評価に関わるデータとデータの性質

た、開講した全ての講義は授業評価の対象となり、学生を対象に授業アンケートが実施される。最後に、実施しているアンケートに学生が回答するので、回答データが集まる。このワークフローに現れる「開講する」、「履修する」、「アンケートを実施する」と「アンケートに回答する」は、それぞれ開講、履修、アンケート実施とアンケート回答に関わる4つのイベントが発生するたびに、生じる事象データである。

ワークフローに基づく、詳細な論理分析やクラス構造全貌の紹介は、紙幅上の理由で省略する。クラス構造に永続性のあるエンティティを抽出し、構築したデータベースモデルは下図のように表わせる。

学生、教員のデータはそれぞれ students、teachers テーブルに格納する。学生の識別子は学籍番号 (studentid)、教員の識別子は教員番号 (teacherid) に設定し、学生と教員を一意的に識別することができる。学生は学籍番号 (studentid) で、教員は loginname でログインする。

科目のデータは subjects テーブルに保存している。授業アンケートは科目群ごとにタイプを定めているので、科目のアンケート区分は外部キーの inquirytypeid を通して、inquirytype テーブルと関連をつける。アンケートタイプ名は科目群名を利用している。アンケートの設問内容と設問数はアンケートタイプごとに異なっている。そのため、配列型の que_length フィールドの中には、設問数が格納されている。

開講科目のデータは classes テーブルに格納している。classid と termno (学期番号) を識別子とし

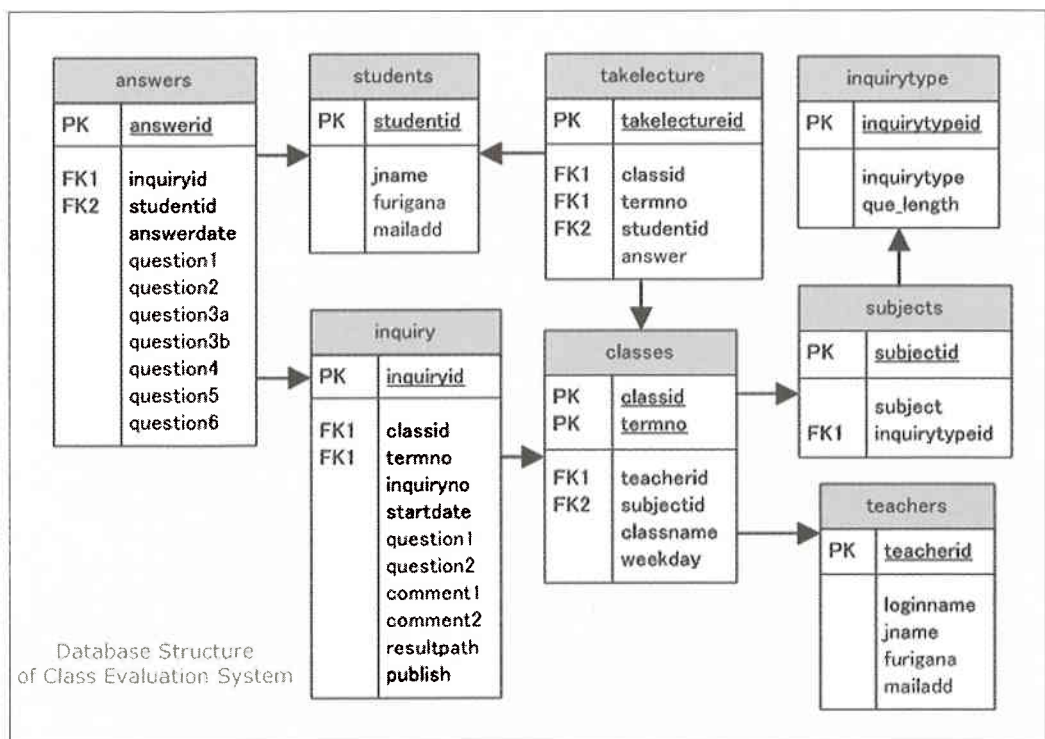


図5 WBCES のデータベースモデル

て指定するため、同一の科目の通年開講問題は解決できる。開講テーブルには、担当教員の情報、科目の情報、開講日の情報が欠かせないため、外部キー teacherid と subjectid を通して、それぞれ教員と科目テーブルに関連をつけている。

履修データは takelecture テーブルに記述し、どの学生がどの講義を履修するかを定めるために、studentid と classid、termno でそれぞれ学生テーブルと開講テーブルに関連づける。アンケートへ回答記録は answer フィールドに記述する。

アンケートの実施データは inquiry テーブルに格納する。開講テーブルと関連づけるために classid と termno 外部キーが設定している。中期、後期アンケートの識別は inquiryno で判断し、アンケート結果の公表可否は publish で決める。公表する場合、公表データの保管パスは resultpath に記憶する。アンケートの開始日、追加設問と教員コメントはそれぞれ startdate、question1 ～ question2 と comment1 ～ comment2 フィールドで収める。

最後に、アンケートの回答データは answers テーブルに保存される。だれが、いつ、どのアンケートに回答したかについて、それぞれ studentid、answerdate と inquiryid フィールドに記述する。各問題への回答は配列型のフィールド question1 ～ question6 に格納する。

図5で示したデータベースモデル以外に、WBCES には 15 のデータビューを設け、いろいろなケースのフォームプレゼンテーションに備える。そのうちのひとつとして、開講科目・アンケート情報を総合した vinquryclass データビューを紹介する。下図に示されたように、このデータビューは inquiry テーブルをベースにし、更に classes、subject、inquirytype と teachers、4 つのテーブルの情報を加えて作ったデータビューである。

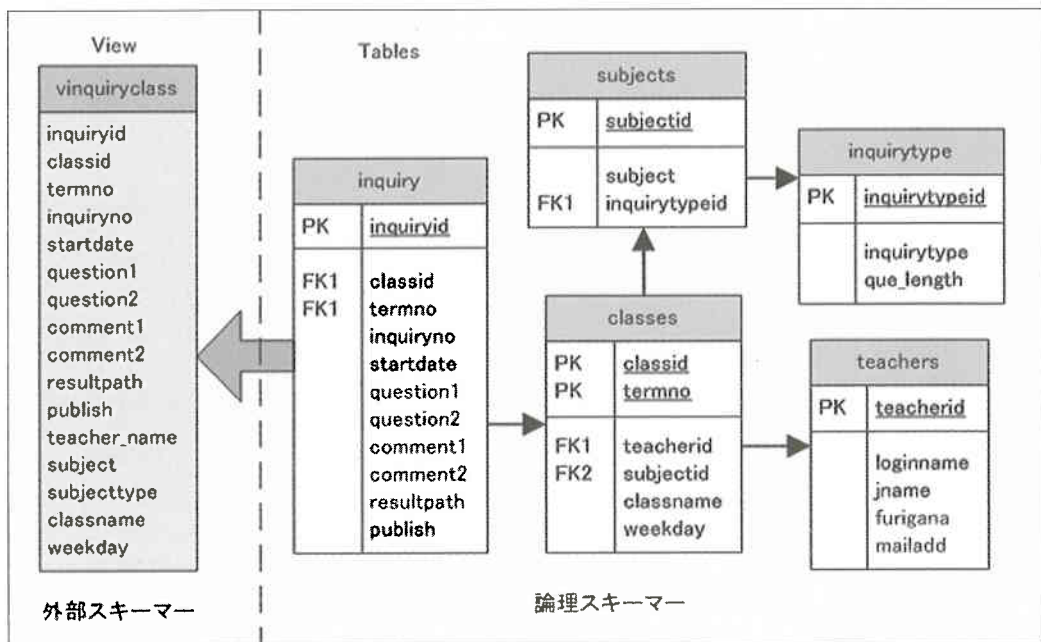


図6 データビュー vinquryclass の構成イメージ

データの集計はデータベースシステムのもう一つ重要な機能である。WBCES には、大量な集計作業に備えるために、47 のデータベース関数を PostgreSQL Store Procedure で定義し、データ集計の効率と能力を向上させた。

3. 標準的な Java データベースプログラミングコード

近年、WECES のようなデータベースと連携したウェブアプリケーションが非常に増えているため、Java データベースプログラミングも標準的になりつつある。SQL (Structured Query Language) によるテーブルの基本操作と JDBC (Java Database Connectivity) によるデータベースへのアクセスは、Java データベースプログラミングの基本と言える。SQL 文を利用したテーブルの操作は Select、Insert、Update と Delete4 種類の基本クエリが含まれている。それらのクエリは JDBC を通して、データベースへ送り、その結果は再び JDBC を経由し、受けとるという流れになる。

図 7 では「教員テーブルから専任教員全員のデータを取り出す」という事例を取り上げ、標準的な Java データベースプログラミングコードを検証してみる。

1 行には java.sql パッケージで規定されているクラスを Import することで、JDBC ドライブに実装される機能を使える環境に整える。

```
1. import java.sql.*;
2. private final String
   url="jdbc:postgresql://localhost:5432/wbcесdb?useUnicode=true&characterEncoding=EUC-JP";
3. private final String user="aichi_user";
4. private final String password="aichi_pwd";

5. Class.forName("org.postgresql.Driver");
6. Connection cn = DriverManager.getConnection(this.url,this.user,this.password);
7. Statement st = cn.createStatement();
8. String sql =" Select * from teachers where teacherid like `1100%`";
9. ResultSet rs = st.executeQuery(sql);
10. ArrayList aTeacherList = new ArrayList();
11. String[] aTeacher;
12. aTeacher = new String[4];
13. while(rs.next()){
14.     aTeacher[0]=rs.getString("teacherid");
15.     aTeacher[1]=rs.getString("loginname");
16.     aTeacher[2]=rs.getString("jname");
17.     aTeacher[3]=rs.getString("furigana");
18.     aTeacher[4]=rs.getString("mailadd");
19.     aTeacherList.add(aTeacher);
20. }
21. rs.close();
22. st.close();
23. cn.close();
```

図 7 標準的な Java データベースプログラミングコード (1): Select ケース

2行～6行では、JDBCを利用しデータベースへの接続を行う。そのうち、5行目に PostgreSQL用の JDBC ドライバをロードし、続いて第6行目では、localhost というデータベースサーバーにある wbc esdb というデータベースに接続を行う。そのとき、データベースユーザー、パスワードと文字コードを設定する必要がある。この接続オブジェクトは cn と定義している。

第7～9行では JDBC を通して SQL クエリをデータベースへ送信し、データベースからの検索結果を受け取る。まず、第7行目では、SQL を送信するために使われる Connection クラスの Statement オブジェクトを st と定義し、次に SQL クエリを文字列 sql に記述する。最後に第9行目で、st.executeQuery(sql) で SQL クエリが送信され、その結果を ResultSet オブジェクトの rs で受け取る。

10行～20行では、ResultSet オブジェクト rs から受け取った教員リストを配列 aTeacherList に格納する。

21行～23行には、ResultSet、Statement と Connection の順に、それぞれのオブジェクトを解放し、データベースとの接続を切断する。

次に、以下のデータを更新するための Java データベースプログラミングコードを検証する。

図8に示したコードと図7で紹介した検索コードを比較してみると、データベース接続部分のコードはまったく同じことがわかる。異なるのは以下の3箇所だけである。

1. 第8行で記述した SQL クエリの内容が違う。
2. 第9行に、SQL を実行するために st.executeUpdate(sql) オブジェクトが使われる。図7の場合、st.executeQuery(sql) を使用し、Select タイプの SQL クエリを実行する。
3. 更新 (Update) クエリには、通常返される結果はないため、ResultSet オブジェクトは使用しない。

追加 (Insert) と削除 (Delete) タイプの Java データベースプログラミングコードも更新 (Update) タイプのコードとほとんど同様であり、修正に必要な箇所は、図8の第8行の SQL 文の内容だけであ

```
1. import java.sql.*;
2. private final String
   url="jdbc:postgresql://localhost:5432/wbc esdb?useUnicode=true&characterEncoding=EUC-JP";
3. private final String user="aichi_user";
4. private final String password="aichi_pwd";

5. Class.forName("org.postgresql.Driver");
6. Connection cn = DriverManager.getConnection(this.url,this.user,this.password);
7. Statement st = cn.createStatement();
8. String sql="UPDATE teachers SET loginname='taro', jname='愛知太郎', furinaga='Taro Aichi',
   mailadd='taro@aichi-u.ac.jp' WHERE teacherid ='1100111' ";
9. st.executeUpdate(sql);
10. st.close();
11. cn.close();
```

図8 標準的な Java データベースプログラミングコード (2): Update ケース

る。

上述のコード分析で、標準Javaデータベースプログラミングに何らかの構文パターンが存在していることが確認された。JDBCパッケージの使用は、固定したプログラムパターンが現われる主な原因である。しかし、SQLクエリは各々の事例により異なっているので、SQL文の実行に使われるオブジェクトは異なる。次の節には、このプログラムパターンを活かし、オブジェクト指向のJavaデータベースプログラミングのデザインパターンを提案する。

4. オブジェクト指向のJavaデータベースプログラミングのデザインパターン

4.1 WBCESにおけるオブジェクト指向デザインの概要

授業評価システムWBCESのアンケートロジック部分の設計は、オブジェクト指向の手法を取り入れた。システム全体は、下図に表れるように8つのパッケージで構成される。そのうち、Teacher、Student、Manager、Inquiry、Subject_ClassとReportは、何らかの実体と対応するクラスで構成したパッケージであり、一方DAO (Database Access Objects) パッケージとUtilityパッケージは補助機能を提供するクラスにより構成され、データベースアクセス機能やJSPファイルの間に情報伝達などの機能を提供している。

表1には各にパッケージに含まれる主なクラスを示す。

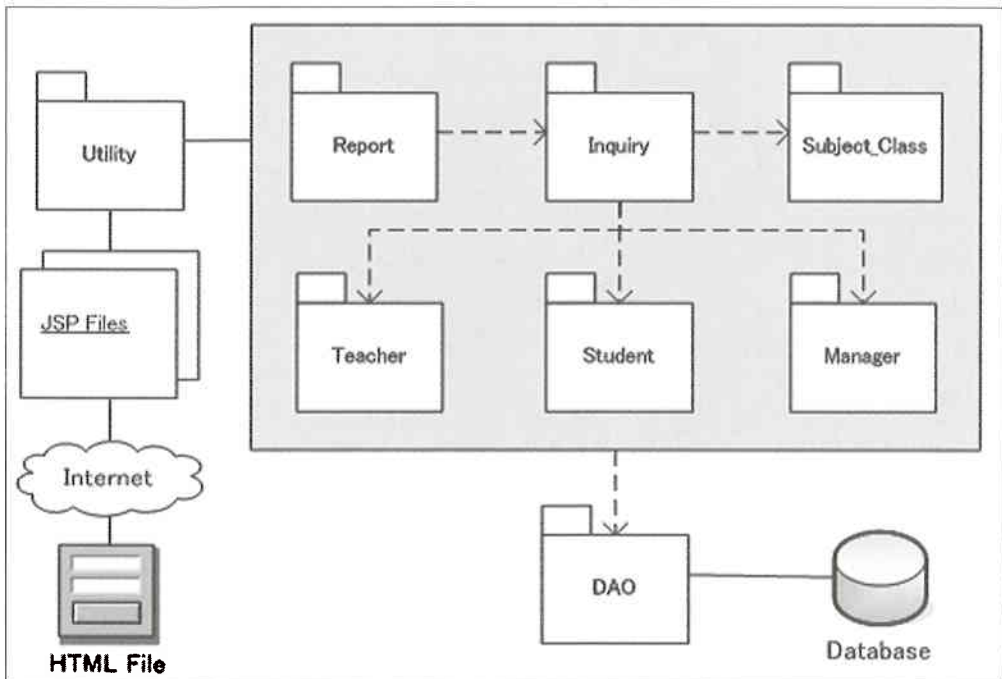


図9 WBCES システムのパッケージ構成

表1 WBCES システムの主なクラス構成

パッケージ名、クラス名	説明
Teacher.cTeacher	単独教員のクラス (Teachers テーブルに対応)
Teacher.beanTeacherList	教員リストのクラス
Student.cStudent	単独学生のクラス (Students テーブルに対応)
Student.beanStudentList	学生リストのクラス
Manager.cManager	単独管理員のクラス
Subject_Class.cSubject	単独科目のクラス (Subjects テーブルに対応)
Subject_Class.beanSubjectList	科目リストのクラス
Subject_Class.cTakeLecture	履修科目クラス (TakeLecture テーブルに対応)
Subject_Class.cClasses	開講科目のクラス (Classes テーブルに対応)
Subject_Class.beanClassesList	開講科目リストクラス
Subject_Class.ClassStudent	履修科目・アンケートの回答状態を扱うクラス (vTakeLecture データビューに対応)
Subject_Class.ClassTeacher	担当科目・アンケート実施情報を扱うクラス (vClass データビューに対応)
Inquiry.cInquiry	単独アンケートのクラス (Inquiry テーブルに対応)
Inquiry.beanInquiryList	アンケートリストクラス
Inquiry.cInquiryType	単独アンケートタイプのクラス (InquiryType テーブルに対応)
Inquiry.cAnswers	単独回答のクラス (Answers テーブルに対応)
Inquiry.beanAnswerList	回答アンケートリストクラス
Inquiry.cInquiryClass	開講講義・アンケート実施情報を扱うクラス (vInquiry データビューに対応)
Inquiry.cStatisticalReport	アンケート回答を集計するクラス (vStatistical_Reports データビューに対応)
Report.cInquiryPublish	公表するアンケートを集計するクラス (Store Procedure に対応)
Report.cInquiryAnswer	学生におけるアンケート回答の状況を集計するクラス (Store Procedure に対応)
Report.cInquiryTeacher	教員におけるアンケート実施の状況を集計するクラス (Store Procedure に対応)
DAO.cDBConnection	データベース接続機能を提供するクラス
DAO.cDAO	SQL クエリと Store Procedure を実施するクラス
DAO.cGetObjectFromDB	データベースから Object リストを引出すクラス
Utility.svltWebCES	サーブレットクラス
Utility.JaasKrbBean	ユーザー認証クラス (Open Source)
Utility.cFile	フォルダ、ファイルを扱うクラス
Utility.ctagDataTable	カスタマー・タグを作成するクラス
Utility.cHTMLWriter	HTML ファイルの書き出すクラス
Utility.cDrawAnswerChart	集計グラフの書く出すクラス
Utility.JFreeChartBean	グラフの基本要素クラス (Open Source)

4.2 WBCES におけるオブジェクト指向の設計思想

WBCES におけるクラスの設計は、次のような考えに沿って開発してきた。

I. 基本クラス (Basic Class) の定義

データベースモデルを踏まえて、永続性のある実体データと実体データを対応したオブジェクトを中心に、基本クラス群を定義する。それらのクラスは、単独のオブジェクトの集合であり、その属性は対応するデータベース・テーブルのフィールドと一致する。テーブルとのやり取りに備え、Insert、Delete、Update と Select など SQL 文を実行するメソッドを実装する。上述の表1の中に、Teacher.cTeacher、Student.cStudent、Subject_Class.cClasses と Inquiry.cInquiry などは基本クラスに属する。

II. 基本クラスからリストクラス (List Class) へ拡張

前述の基本クラス群は単独のオブジェクトに注目したクラス群である。たとえば、一名の教員、一人の学生、一科目と一回アンケートなどがあげられる。一方、WBCES には同類のオブジェクトの集合も頻繁に使われる。たとえば、ある教員の担当科目一覧、ある学生の履修科目一覧、予定しているアンケートのリストとアンケート回答リストなどは具体化したオブジェクトの集合になる。このようなオブジェクトの集合も効率よく作り出すために、各種のリストクラスを定義した。リストクラスは、前述の基本クラスを使用し、ArrayList で基本クラスの配列を返すことになる。その意味で、リストクラスは基本クラスの拡張と言える。たとえば、基本クラスの Teacher.cTeacher に対し、その拡張 Teacher.beanTeacherList はリストクラスになる。

III. 基本クラスからプレゼンテーションクラス (Presentation Class) へ拡張

次にデータベースのデータビューに注目し、データビューごとのプレゼンテーションクラスを定義する。複数のテーブルから必要な情報を抽出し、形成したのはデータビューである。ほとんどのユーザーインターフェースで表示したのはデータビューの情報である。MVC 型の Java ウェブアプリケーションの場合、データビューの情報こそ、サーブレットを経由し、JSP まで伝達する必要がある。オブジェクト指向の観点から見れば、プレゼンテーションクラスも基本クラスを特化したクラスである。たとえば、プレゼンテーションクラス Subject_Class.ClassStudent は、基本クラス Subject_Class.cTakeLecture を継承し、更に Inquiry の情報を加え、拡張したクラスである。このクラスは学生画面に学期ごとの履修科目一覧と対応したアンケート実施状態を表示するために使われる。

IV. 集計クラスの定義

上述の I から III のクラスは、基本クラスの Family であり、以下の 2 つの共通点を持つ。

- ① クラスの属性は、何らかの実体データ、或いは事象データと対応する。
- ② クラスのメソッドには、SQL 文を実行するため機能を実装している。

WBCES の集計クラスは、さまざまなアンケート結果を集計するために定義したクラスであるので、基本クラスを継承する必要はない。なお、データの集計を効率的に行われるために、直接データベース内部の Store Procedure を駆動した集計メソッドを実装した。表1の Report.cIn-

quiryPublish、Report.cInquiryAnswer と Report.cInquiryTeacher は集計クラスである。

V. 補助クラスの追加

既に述べた DAO パッケージと Utikity パッケージに含まれたクラスは、補助クラスである。特に、Utility.svltWebCES クラスは、サブレッドクラスとして、MVC モデルの中にコントロールの役割を果たしている。また DAO パッケージに含まれる 2 つのクラスは抽象クラスとして定義されている。それぞれ備えている SQL 実行とオブジェクトのリスト化機能を、全ての基本クラスに継承させるため、コードの再利用性は極めて高いレベルに達した。この部分の詳細について、次の節に解説する。

4.3 WBCES におけるオブジェクト指向のデータベースプログラムデザインパターン

全ての基本クラスは SQL 文を実行するメソッドが実装されている。データベースのテーブルとのやり取りの際に、1 オブジェクトがテーブルの 1 レコード (1 行) とデータマッピングしながら、情報の取り出しや書き込みが行われる。性格の違うオブジェクトに対して、当然データマッピングコードも異なってくる。ただし、共通の JDBC を利用しているため、第 3 節で検討したプログラムのパターンが、どこまで最大限に共有できるかは、設計の最大目標であった。

全ての基本クラスに対し、データベースアクセス、SQL 文の実行とオブジェクトのリスト化、3 つの機能を提供するクラス群を次の DAO パッケージとしてまとめた。

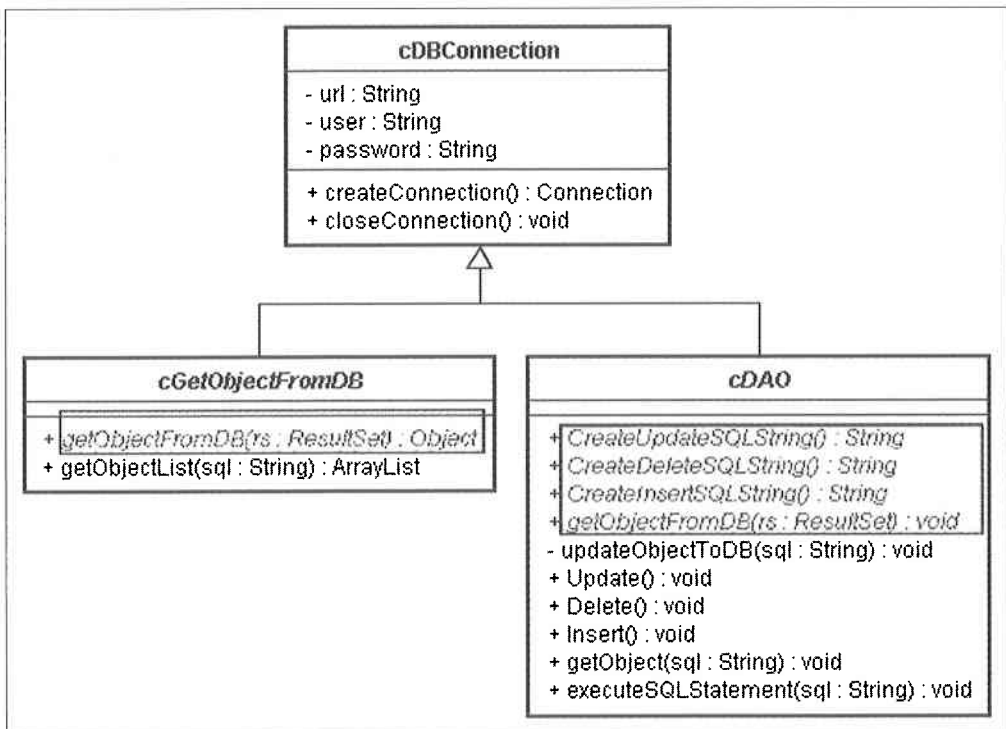


図 10 DAO パッケージ構成

DAO パッケージは3つのクラスにより構成される。cDBConnection はデータベースへのアクセス、cDAO は SQL 文の実行、cGetObjectFromDB はオブジェクトリストの作成、3つの役割を分担している。データベースへのアクセスは、全ての作業に関わっているため、cDBConnection クラスを cDAO クラスと cGetObjectFromDB クラスに継承させる。

cDBConnection クラスには、Connection 型の createConnection() 関数と closeConnection メソッドを実装している。データベースに接続するためのパラメーターを一箇所に設定することで、テスト・管理上のメリットは大きい。

クラス cDAO は、cDBConnection を継承した抽象クラスである (図 12)。外部に対し Update (行 22 ~ 行 24)、Delete (行 25 ~ 行 27)、Insert (行 28 ~ 行 30)、getObject (行 31 ~ 行 48) と executeSQLstatement (行 49 ~ 行 59)、5つのメソッドを提供する。前の4つのメソッドは、Update、Delete、Insert と Select、4つのタイプの SQL クエリに対する実行機能を提供し、5つ目の executeSQLstatement はデータベース Store Procedure を駆動するためのメソッドである。あらゆるオブジェクトに対応できるためには、SQL クエリの記述部分 (行 4 ~ 行 6) とデータマッピング部分

```
1. package aichi.wbces.dao;
2. import java.sql.*;

3. public class cDBConnection {
4.     private final String
        url="jdbc:postgresql://localhost:5432/wbcesdb?useUnicode=true&characterEncoding=EUC-JP";
5.     private final String user="aichi_user";
6.     private final String password="aichi_pwd";

7.     public cDBConnection(){};

8.     public Connection createConnection(){
9.         try {
10.             Class.forName("org.postgresql.Driver");
11.             Connection cn = DriverManager.getConnection(this.url,this.user,this.password);
12.             return cn;
13.         } catch (ClassNotFoundException e) {
14.             System.out.println("Can not found JDBC Driver. ¥n");
15.         } catch (SQLException e) {
16.             System.out.println("Connect Error. ¥n");
17.         }
18.         return null;
19.     }

20.     public void closeConnection(Connection cn) {
21.         try {
22.             cn.close();
23.         } catch (Exception ex) {}
24.     }
25. }
```

図 11 cDBConnection クラスのコード

(行7) は抽象的に宣言し、コードの実装はそれぞれの基本クラスに譲り渡す。

```
1. package aichi.wbces.dao;
2. import java.sql.*;

3. public abstract class cDAO extends cDBConnection{
4.     public abstract String CreateUpdateSQLString();
5.     public abstract String CreateDeleteSQLString();
6.     public abstract String CreateInsertSQLString();
7.     public abstract void getObjectFromDB (ResultSet rs);

8. public cDAO(){}

9. private void updateObjectToDB(String sql){
10.     Connection cn = null;
11.     try {
12. cn=this.createConnection();
13.         Statement st = cn.createStatement();
14.         st.executeUpdate(sql);
15.     }
16.     catch (Exception e) {
17. e.printStackTrace();
18.     } finally {
19. this.closeConnection(cn);
20.     }
21. }

22. public void Update(){
23.     this.updateObjectToDB(this.CreateUpdateSQLString());
24. };

25. public void Delete(){
26.     this.updateObjectToDB(this.CreateDeleteSQLString());
27. };

28. public void Insert(){
29.     this.updateObjectToDB(this.CreateInsertSQLString());
30. };

31. public void getObject(String sql){
32.     Connection cn = null;
33.     try {
34.         cn = this.createConnection();
35.         Statement st = cn.createStatement();
36.         ResultSet rs = st.executeQuery(sql);
37.         while(rs.next()){
38.             this.getObjectFromDB(rs);
39.         }
40.         rs.close();
41.         st.close();
42.     }
43.     catch (Exception e) {
44.         e.printStackTrace();
```

```

45. } finally {
46.     this.closeConnection(cn);
47. }
48. }

49. public void executeSQLstatement(String sql){
50.     Connection cn=this.createConnection();
51.     try {
52.         Statement st = cn.createStatement();
53.         st.execute(sql);
54.     }
55.     catch (Exception e) {
56.         e.printStackTrace();
57.     } finally {
58.         this.closeConnection(cn);
59.     }
60. }
}

```

図 12 cDAO クラスのコード

```

1. package aichi.wbces.dao;
2. import java.sql.*;
3. import java.util.*;

4. public abstract class cGetObjectFromDB extends cDBConnection {

5.     public cGetObjectFromDB(){};
6.     public abstract Object getObjectFromDB (ResultSet rs);

7.     public ArrayList getObjectList(String sql){
8.         ArrayList ResultList= new ArrayList();
9.         Connection cn = null;
10.        try {
11.            cn = this.createConnection();
12.            Statement st = cn.createStatement();
13.            ResultSet rs = st.executeQuery(sql);
14.            while (rs.next()) {
15.                ResultList.add(this.getObjectFromDB(rs));
16.            }
17.            rs.close();
18.            st.close();
19.            return ResultList;
20.        }
21.        catch (Exception e) {
22.            e.printStackTrace();
23.            return null;
24.        } finally {
25.            this.closeConnection(cn);
26.        }
27.    }
}

```

図 13 cGetObjectFromDB クラスのコード

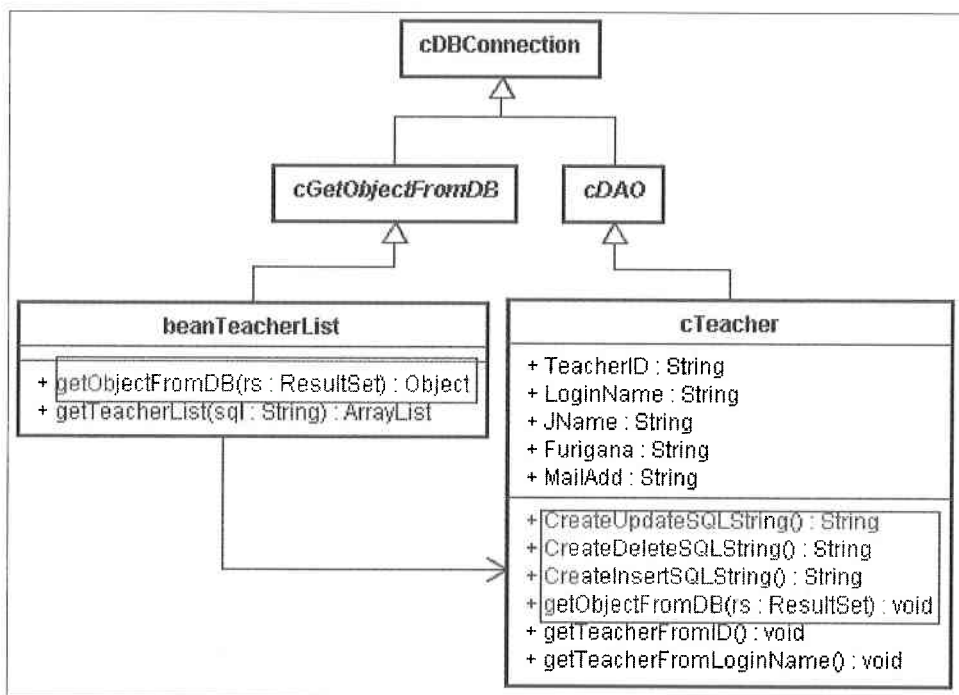


図 14 DAO を継承した教員クラスと教員リストクラス

クラス cGetObjectFromDB は、cDAO と同様に cDBConnection を継承した抽象クラスである。複数のオブジェクトを抽出し、配列型のオブジェクトリストを返すことは getObjectList (行 7～行 27) メソッドの任務である。そのうち、各々のオブジェクトは第 15 行の this.getObjectFromDB で受け取っているが、このメソッドも抽象メソッドとして第 6 行で定義され、その中のデータマッピングの実装は各基本クラスに譲る。

以上の DAO パッケージを定義した後、次には基本クラス (プレゼンテーションクラスも同様) とリストクラスがどのように DAO を継承するかについて、教員クラスの例で解説する。

基本クラス cTeacher は cDAO 抽象クラスを継承しているので、cDAO で定義された抽象メソッドを実装する必要がある。また cDAO に持っている Update、Delete、Insert などのメソッドは使える。一方、リストクラス beanTeacherList クラスは cGetObjectFromDB を継承しながら、cTeacher のオブジェクトを次々と ArrayList に取り入れ、cTeacher のリストを配列型で返される。

図 14 に示されたように、JDBC に関わるほとんどの作業が DAO パッケージに集約され、教員クラスに特化した作業は cTeacher と beanTeacherList に残されている。そのため、DAO パッケージと Teacher パッケージは完全に分離することができ、お互い独立していることが確認できる。その DAO パッケージの独立性によって、優れたシステムの保守性とコードの再利用性を得られた。

```

1. package aichi.wbces.teacher;
2. import aichi.wbces.dao.*;
3. import java.sql.*;
4. import java.util.Date;
5. import java.util.*;
6. import java.text.*;

7. public class cTeacher extends cDAO {

8.     private String TeacherID_="";
9.     private String LoginName_;
10.    private String JName_="";
11.    private String Furigana_="";
12.    private String MailAdd_="";

13.    public cTeacher(){};

14.    public String getTeacherID(){return this.TeacherID_;}
15.    public void setTeacherID(String TeacherID){this.TeacherID_=TeacherID;}
16.    public void setJName(String JName){this.JName_=JName;}
17.    public String getJName(){return this.JName_;}
18.    public String getLoginName() {return this.LoginName_;}
19.    public void setLoginName(String name) {this.LoginName_ = name;}
20.    public String getFurigana(){return this.Furigana_;}
21.    public void setFurigana(String Furigana){this.Furigana_=Furigana;}
22.    public String getMailAdd(){return this.MailAdd_;}
23.    public void setMailAdd(String MailAdd){this.MailAdd_=MailAdd;}

24.    public String CreateUpdateSQLString(){
25.        String sql="UPDATE teachers SET loginname=" + this.LoginName_ + " jname=" +
                this.JName_ + " furinaga=" + this.Furigana_ + " mailadd=" + this.MailAdd_ + "
                WHERE teacherid = " + this.TeacherID_ + " ";
26.        return sql;
27.    };

28.    public String CreateDeleteSQLString(){
29.        String sql="DELETE FROM teachers WHERE ( teacherid = " + this.TeacherID_ + " )";
30.        return sql;
31.    };

32.    public String CreateInsertSQLString(){
33.        ArrayList DBCol = new ArrayList() ;
34.        ArrayList Values = new ArrayList();
35.        DBCol.add("teacherid");
36.        DBCol.add("loginname");
37.        DBCol.add("jname");
38.        DBCol.add("furigana");
39.        DBCol.add("departmentdivisionid");
40.        DBCol.add("mailadd");
41.        Values.add(""+ this.TeacherID_ + "");
42.        Values.add(""+ this.LoginName_ + "");
43.        Values.add(""+ this.JName_ + "");
44.        Values.add(""+ this.Furigana_ + "");

```

```

45. Values.add("" + this.divisionID_ + "");
46. Values.add("" + this.MailAdd_ + "");
47. String sql="INSERT INTO teachers ( teacherid, loginname, jname, furigana, mailadd )
VALUES ( "" + this.TeacherID_ + "," + this.LoginName + "," + this.Jname + "," +
this.mailadd + " )";
48. return sql;
49. };

50. public void getObjectFromDB(ResultSet rs){
51. try{
52. this.setTeacherID(rs.getString("teacherid"));
53. this.setLoginName(rs.getString("loginname"));
54. this.setJName(rs.getString("jname"));
55. this.setFurigana(rs.getString("furigana"));
56. this.setMailAdd(rs.getString("mailadd"));
57. }
58. catch (SQLException e) {
59. e.printStackTrace();
60. }
61. }

62. public void getTeacherFromID(){
63. String sql="select * from vteacher where teacherid = "" + this.TeacherID_ + """;
64. this.getObject(sql);
65. }

66. public void getTeacherFromLoginName(){
67. String sql="select * from vteacher where loginname = "" + this.LoginName_ + """;
68. this.getObject(sql);
69. }
70. }

```

図 15 cTeacher クラスのコード

「cTeacher クラスコードの解説」

- 行 1～行 6： パッケージ aichi.wbces.teacher の中に、cTeacher クラスを宣言し、また、この cTeacher クラスの中に、aichi.wbces.dao パッケージと複数の java.sql、java.util と java.text パッケージの利用を宣言する。
- 行 7： cDAO クラスを継承しながら、cTeacher クラスを宣言する。
- 行 8～行 12： cTeacher クラスフィールドをカプセルの内部で宣言する。
- 行 13： インスタンスの初期処理を行うためのコンストラクタ (Constructor)
- 行 14～行 23： クラスの外部からフィールドを操作するためのメソッドを宣言する。
- 行 24～行 27： cDAO に定義した CreateUpdateSQLString 抽象メソッドを実装する。
- 行 28～行 31： cDAO に定義した CreateDeleteSQLString() 抽象メソッドを実装する。
- 行 32～行 49： cDAO に定義した CreateInsertSQLString 抽象メソッドを実装する。
- 行 50～行 61： cDAO に定義した getObjectFromDB 抽象メソッドを実装する。
- 行 62～行 65： 教員 ID から教員のデータを抽出し、教員インスタンスを作り出すメソッドを定義する。

行 66 ～行 69： LoginName からデータを抽出し、教員インスタンスを作り出すメソッドを定義する。

抽象メソッド、CreateUpdateSQLString、CreateDeleteSQLString() と CreateInsertSQLString の実装は、カプセル化したオブジェクトの内部フィールド情報をカプセルの外部へ取り出し、テーブルのフィールドに合わせながら、それぞれ Update、Delete、Insert タイプの SQL 文を構成する。その過程はディス・カプセル (Discapsule) と呼ぶ。

一方、抽象メソッド getObjectFromDB の実装は、逆の方向でデータを流れている。教員テーブルからデータが抽出され、そのデータは ResultSet 変数 rs を通して引き渡される。次に ResultSet 変数 rs に格納している教員テーブルの各フィールドデータを、それぞれクラスフィールドに合わせ、教員オブジェクトの内部フィールドに入れる。その過程はカプセル化と呼ぶ。

カプセルとディス・カプセルの過程により、オブジェクトとデータベースの間にデータマッピングが行われ、詳細な作業は cDAO の getObject(sql) に集約されたため、getTeacherFromID() メソッドと getTeacherFromLoginName() メソッドは非常に簡潔な形で記述できた。

次に教員リストクラスへの拡張コードを解説する。

```
1. package aichi.wbces.teacher;
2. import java.sql.*;
3. import java.util.*;
4. mport aichi.wbces.dao.*;

5. public class beanTeacherList extends cGetObjectFromDB {

6. public beanTeacherList(){};

7. public Object getObjectFromDB(ResultSet rs){
8.     cTeacher aTeacher = new cTeacher();
9.     aTeacher.getObjectFromDB(rs);
10.    return aTeacher;
11. }

12. public ArrayList getTeacherList(String sql){
13.     ArrayList aTeacherList = this.getObjectList(sql);
14.     return aTeacherList;
15. }
16. }
```

図 16 beanTeacherList クラスのコード

「beanTeaherList クラスコードの解説」

行 1 ～行 4： パッケージ aichi.wbces.teacher の中に、beanTeacherList クラスを宣言し、また、このクラスの中に、aichi.wbces.dao パッケージと java.sql と java.util パッケージの利用を宣言する。

- 行 5 : cGetObjectFromDB クラスを継承しながら、beanTeacherList クラスを宣言する。
- 行 6 : インスタンスの初期処理を行うためのコンストラクタ (Constructor) である。
- 行 7～行 11 : cGetObjectFromDB に定義した getObjectFromDB 抽象メソッドを実装する
- 行 12～行 15 : 複数の教員インスタンスを配列型で返すメソッドを定義する

抽象メソッド getObjectFromDB の実装に、cTeacher のインスタンスが使われることで、継承している cDAO の getObjectFromDB(rs) がそのまま再利用できる。また、教員リストを抽出するためのメソッド getTeacherList(sql) も、cGetObjectFromDB クラスから継承した getObjectList(sql) メソッドを利用し、簡単に教員リストを返すことができた。

今回のオブジェクト指向の設計案が、いまの段階になって、コードの再利用性にもたらずメリットが実感できるようになった。

5. 応用事例

DAO パッケージを利用することで、サーブレットと JSP でのプログラミングは極めて簡潔になった。この節では、2つの事例を通してデータベースプログラムデザインパターンを利用したサーブレットと JSP プログラミングを紹介する。

5.1 事例 1 : 学生の履修科目一覧を抽出し、Web 上表示する

作業の流れは次のようになる。まず、学籍番号 (UserID) を指定し、下図のサーブレットメソッド jobShowLectureList に送信する。サーブレットでは、データビュー vtakelecture (学生履修情報・アンケート実施状態を集めるデータビュー) から、該当する履修科目リストを抽出し、その結果を JSP ファイルへ送る。そして JSP ファイルは、サーブレットから送ってきたリストを受け取って、HTML コードに埋め込む。次に作業の流れにそってコードを紹介する。

```
1. private void jobShowLectureList(HttpServletRequest request, HttpServletResponse response,
    String UserID) throws ServletException, IOException {
2. HttpSession session = request.getSession(true);
3. beanClassStudentList cList = new beanClassStudentList();
4. String sql="select * from vtakelecture where studentid='" + UserID + "' order by fullclassname";
5. ArrayList ClassList = cList.getClassStudentList(sql);
6. session.setAttribute("sesLectureList",ClassList);
7. response.sendRedirect("ces/students/side.jsp");
8. }
```

図 17 学生履修科目リストを抽出するサーブレットのコード

コードの説明 :

- 行 1 : メソッド jobShowLectureList を定義し、学籍番号を UserID として受け取る。
- 行 2 : 履修科目データを次の JSP ファイルへ転送するために、ウェブアプリケーションのセッ

ションを設定する。

行 3： 履修科目リストを扱うクラス `beanClassStudentList` のインスタンスを `cList` として定義する。

行 4： SQL 文を記述する。

行 5： SQL 文を実行し、抽出した履修科目のリストを配列 `ClassList` に格納する。

行 6： 配列 `ClassList` をセッション `sesLectureList` に乗せて、JSP へ発信する。

行 7： 次に、ブラウザに学生画面のメニューページ `"ces/students/side.jsp"` 表示するように指示する。

次に、JSP 側のコードを見てみよう。

```
1. <%@ page contentType="text/html; charset=EUC-JP" session="true"
    import="java.util.*,aichi.wbces.subject_class.*"%>
2. <html>
3. <head>
4. <title> 学生 </title>
5. </head>
6. <% ArrayList ClassList = (ArrayList)session.getAttribute("sesLectureList");%>

7. <table>
8. <tr><td> 履修科目名 </td><td> 回答状況 </td></tr>
9. <% int ListSize=ClassList.size();
10. for (int i=0; i<ListSize; i++) {
11.     cClassStudent s = (cClassStudent)ClassList.get(i); %>
12.     <tr>
13.         <td><%= s.getFullClassName() %></td><td><%= s.getAnswer() %></td>
14.     </tr>
15. <% } %>
16. </table>
17. </body>
18. </html>
```

図 18 学生履修科目リストを表示する JSP のコード

コードの説明：

行 1： JSP ページが呼び出されるたびに実行するディレクティブである。その中に HTML より出力、EUC_JP の文字コードの指定と `java.util` と `aichi.wbces.subject_class` パッケージの利用を宣言する。

行 2～行 18： HTML コード。その中、以下の JSP のプログラムは埋め込まれている。

行 6： セッション `sesLectureList` によりサーブレットから送られてきている履修科目リストデータを受け取り、配列 `ClassList` に格納する。

行 9： 配列 `ClassList` のサイズで、履修科目の数を取得する。

行 10～行 14： 一科目一行で、HP 上に表を書き出す。そのうち、

行 11： 配列から各々の履修科目を `cClassStudent` クラスのインスタンス `s` に格納する

行 13： スクリプト式 `<%= %>` で、履修科目名 `s.getFullClassName()` とアンケート実

施状態 `s.getAnswer()` をテーブルのセルに埋め込む

データベースの外部で、各々のクラス内部で記述した SQL 文を、DAO パッケージを通して、データベースを操作する仕組みをオブジェクト指向のデザインパターンで実現した。

ところが、より複雑なデータベースの操作、或いは大量なデータベース操作をまとめて行う場合、データベース内部の Store Procedure を利用する必要がある。Store Procedure とは、データベース操作ためのプログラムであり、データベース内部に置くことで、効率的に実行することができる。

DAO パッケージを Store Procedure の使用まで拡張することは、WBCES システムにとって非常に重要である。次には、管理員のアンケート初期設定作業を例として紹介し、DAO の Store Procedure への使用を説明する。

5.2 事例 2：後期アンケートの初期設定

作業には以下 3 つの目的がある。

- (1) 春・秋学期ごとに、後期アンケートの実施日を決める
- (2) 全ての開講科目を対象に、1 回だけの後期アンケートを設定する
- (3) 以上のデータを一括的に inquiry テーブルに書き込み

一回でやく 2000 個以上のデータを一気に書き込まれるので、そのために以下の Store Procedure プログラム `insert_final_inquiry` が使われている。

```
1. declare
2. term_no alias for $1;
3. start_date alias for $2;
4. total integer;
5. begin
6. execute `insert into inquiry(classid, termno, inquiryno, startdate, question1,
   question2, comment1, comment2, resultpath, publish)
   select classid, term_no as termno, 12 as inquiryno, start_date as inquirydate,
   " as question1, 'as question2, 'as comment1, 'as comment2,
   " as resultpath, -1 as publish
   from classes where termno= term_no and classid not in
   (select classid from inquiry where termno= term_no);`1
7. get diagnostics total = row_count;
8. return total;
9. end;
```

図 19 アンケート初期設定ための Store Procedure コード：insert_final_inquiry

コードの説明：

- 行 1～行 4： 引数 `term_no` (学期番号)、`start_date` (開始日) を宣言する。
また、データベースの処理回数を結果として返すための変数 `total` を宣言する。
- 行 6： inquiry テーブルへデータを追加するための SQL 実行文。

¹ この SQL 文を読みやすくするために、いくつかの文字処理ための記号を省略している。

行 7: データベースの処理回数を取得する。

行 8: データベースの処理回数を返す。

図 19 に示された Store Procedure コードを実行するためには、cDAO クラスの中に下図に示す executeSQLstatement メソッドを用意した。

```
1. public void executeSQLstatement(String sql) {
2.     Connection cn=this.createConnection();

3.     try {
4.         Statement st = cn.createStatement();
5.         st.execute(sql);
6.     }
7.     catch (Exception e) {
8.         e.printStackTrace();
9.     } finally {
10.        this.closeConnection(cn);
11.    }
12. }
13. }
```

図 20 cDAO クラスの executeSQLstatement メソッド

そこで第 5 行目の st.execute(sql) は、JDBC パッケージの Statement オブジェクトのメソッドである。実行する Store Procedure の指定は、sql 文に記述し、st.execute(sql) の形で実行させる。

この作業は管理員の権限であり、クラス cManager が cDAO を継承することで、管理員はこの Store Procedure を駆動するメソッドが利用できる。

以下に、サーブレットに後期アンケートの初期設定のための jobManagerSetFinalInquiry メソッドを通して、executeSQLstatement の使い方を説明する。

```
1. private void jobManagerSetFinalInquiry(HttpServletRequest request,
2.     HttpServletResponse response) throws ServletException, IOException {

3.     cManager manager = new cManager();

4.     int TermNo = Integer.parseInt(request.getParameter("term_no"));
5.     int year1 = Integer.parseInt(request.getParameter("year"));
6.     int month1= Integer.parseInt(request.getParameter("month"));
7.     int day1 = Integer.parseInt(request.getParameter("day"));

8.     String strDate = year1 + "-" + month1 + "-" + day1;
9.     String sql="select insert_final_inquiry_date(" + ts.getTermNo() + "," + strDate + ")";

10.    manager.executeSQLstatement(sql);
11.    getServletConfig().getServletContext().getRequestDispatcher("/ces/managers/project.jsp").
12.        forward(request,response);
13. }
```

図 21 cDAO クラスの executeSQLstatement メソッド

コードの説明：

行 1: メソッド jobManagerSetFinalInquiry を定義する。

- 行 2： 管理員クラスのインスタンス `manage` を定義する
- 行 3～行 6： 外部 HP から学期番号、アンケート開始日の年、月と日を取得する
- 行 7： アンケート開始日の文字列を作成。
- 行 8： Store Procedure 関数 `insert_final_inquiry` を指定するための SQL 文を記述する。
- 行 9： `executeSQLstatement(sql)` で Store Procedure を駆動する。
- 行 10： ブラウザが管理員画面に戻る。

まとめ

大学の授業評価システム WBCES の開発にあたって、オブジェクト指向における再利用のためのデータベースプログラミングデザインパターンを提案した。このデザインパターンは、Java データベースプログラムで利用される JDBC パッケージに関連するコードと、各々のオブジェクトの特有な SQL クエリを分離させることで、高いコードの再利用性が得られた。

WBCES の関連事例を紹介することにより、提案したデータベースプログラミングのデザインパターンは、高度な汎用性を持って、広範囲の Java ウェブアプリケーションに応用できることが判明した。

参考文献

- [1] アイティースト著、はじめての JSP & サーブレットプログラミング、秀和システム、2004.
- [2] 今野睦 等著、サーブレット / JSP、ソフトバンク社、2003.
- [3] 今野睦 等著、データベース設計のための UML、翔泳社、2003.
- [4] 結城浩著、オブジェクト指向における再利用のためのデザインパターン、ソフトバンク社、2002.
- [5] Richard Stones、Neil Matthew 著、エキスパートから学ぶ PostgreSQL 活用テクニック、インプレス社、2002.

中国大陸部におけるマルチメディア支援教授

任 景波

愛知大学文学部

主 旨

中国大陸部において、マルチメディアとマルチメディア支援教授という言葉についてはさまざまな解釈がある。二十世紀八十年代におけるマルチメディア支援教授は、多品種のメディアを総合的に活用し、学生の学習効果の向上を目的とした視聴覚教育のことであった。九十年代に入ってから、マルチメディアはデジタル化された映像、音声、グラフィックス、それに文字データなどを組み合わせた複合システムを指す場合が多くなり、マルチメディア支援教授に新たに要素を取り入れ、コンピュータに基づいたマルチメディア技術を活用した教授方法を意味するようになり、遠隔教育を含む場合もある。中国政府は教育におけるマルチメディアの活用戦略を取り上げるのは、中国における教育の産業化、とりわけ大学教育部門の産業化という政策目標に合致したものである。中国は経済的には発展途上国でありながら、先進国へのキャッチアップの一環として、教育やハイテク産業に大いに力を入れてきた。マルチメディアをはじめとする情報通信技術の活用は、高等教育の充実に新たな可能性を開くものとして大きな効果を期待できるものであり、それに教育機関において円滑に実施されるための条件整備を積極的に図っていくことが求められる。本論文では、中国大陸の教育、特に大学教育におけるマルチメディア教授推進戦略について考察し、その政策に期待された効果と、実際にその政策が行われた結果として、生じたさまざまな問題について議論を行いたい。

キーワード：マルチメディア支援 遠隔教育 教育工学 インターアクティブ

目 次

1. 中国大陸部におけるマルチメディア支援教授の定義と意味
2. インターネットを利用した遠隔教育及び語学授業
3. 中国大陸部における教育のマルチメディアを活用した教授戦略の流れ
4. 大学教員の現代教育技術に関する訓練プログラム
5. 大学のマルチメディア活用戦略がもたらした問題
6. 真のマルチメディア支援教授へ
7. 結 語

主要参考文献：

1. 中国大陸部におけるマルチメディア支援教授の定義と意味

中国の大陸部においては、政府の指導のもとで、マルチメディア支援教授の推進、ネットワークシステム資源の整合が、従来の教育パターンを変えようとしており、大きな成果を上げてきていると大いに評価されている。その一方、マルチメディア支援教授戦略にさまざまな問題も潜んでいる。これらの成果を評価し、問題を批判するには、まずマルチメディア支援教育の概念を明らかにする必要がある。というのは、マルチメディア支援教育戦略によって生じた問題の一部が、マルチメディア支援教育という概念についてあいまいな理解に基づいているため、これらの問題を解消するためには、概念に対して明白な解釈を行わなければならないからである。

伝統的に優位に立つ教授道具は、黒板とチョークに、紙の教科書を合わせたものであり、いまでも一般的に使われている。広い意味においては、黒板とチョークに、紙の教科書といったような単一の媒体を利用した教授方式はシングルメディア支援教授といってもいい。これに対し、マルチメディア支援教授は、マルチメディアを活用した教授のものであり、それに含まれる内容はマルチメディアの概念にかかわる。

マルチメディアは「Multimedia」の訳語で、中国語では「多媒体」といい、「Multiple（多様な、多くの部分から成る）」と「Media（媒体、メディア）」という二つの語源からなる。媒体についても、少なくとも二つの意味があり、一つは、情報伝達の媒介手段を指すもので、紙面、画面や映像などを意識した場合が多いが、マスコミを意味する場合も少なくはない。もう一つは、記憶媒体のことであり、近代的な電気技術、情報技術と関連することである。今日において、マルチメディアはデジタル化された画像、グラフィックス、映像、音声そして文字データなどを組み合わせた複合システムであるが、一言でマルチメディアといっても、特にマルチメディア支援教授に言及した場合、あいまいな部分が多く、実は非常に広い分野に及んでいる。

伝統的には、学校が教育の中心的存在になる必要条件是、学校には知識を蓄積した書籍があることや、知識について詳しい解釈を行う教育者がいることにある。これについては、唐の韓愈（字退之、紀元769-834年）の『師説』に、教育者の仕事の内容を「伝道、授業、解惑」と主張したのである。しかし、近代になってから、学校の垣根がだんだん崩れてきており、学校教育自身にもさまざまな変化が生じている。つまり、教育の普及である。この教育の大衆化という動きは、社会経済の発展を基礎とした知識の社会化が進んできたことによるものであり、その背景には、知識の伝達手段の進歩や教授方法の多様化などがある。

実際には、この動きは古代から今日にかけて、絶えず進化してきたものであり、これからも、進化していき、情報伝達技術の進歩が社会に与えるインパクトは計り知れないことになると考えられる。一般的に、紀元前一五世紀頃からとされるが、持ち歩ける情報の媒体としての亀甲や獣骨に甲骨文が刻まれた。その後、文字の記載は皮、木などの媒体に移っていく。人類社会の歴史と比べると、近世になるであろうが、紙の発明は、当時の世界にとっては、革命的なものであり、教育が初めて一般大衆にとって身近なものとなったといえよう。

このような骨、皮、紙などが媒体であり、機能的には、磁気ディスク・磁気テープなどのような、今日の記憶媒体と変わらないものである。

一方、知識や教育と直接な関係を持たないが、古代においては、最も効率的な情報伝達手段として知られる烽火台が三千年前の中国の周朝にすでにあった。外敵が襲来したといった知らせを瞬時にとっていいほど烽火台ごとに遠方へ伝達していく。それは、まさに今日の情報ハイウェイのような効率的な情報機器であった。

二十世紀に入ってから、世界中多様な通信メディアを高度に活用した教育研究の取組がさまざまな形で行われ始めたが、あくまでも「支援」、「補助」などの言葉で表現されたように、これらの通信メディアは、教師の役割に取って代わることはできなかった。学生の主体的学習を現実に結びつけたのは、コンピュータ技術などの現代情報技術の発達であり、コンピュータ、インターネットに基づいたマルチメディア技術は、教育の現場をこれまでと比べるものにならないほど拡大したのである。教育分野全体からみると、教師も学生に対して伝統の権威的な立場から降りてきて、学生を教授の場における平等なパートナーとして付き合い、新たな授業法を探らなければならない状況に強いられたことは認めざるを得ない。中国では、政府が情報教育の将来性を考えて、いち早く全国的なマルチメディア支援教授の推進戦略を打ち出したのである。

コンピュータマルチメディア教授は、九十年代の初めから既に始まり、今は、中国大陸部にあるほとんどの大学、大部分の中学校、高校及び一部の小学校にマルチメディア支援教育を行っている。

広い意味においては、マルチメディアは、多種のメディアをある目的を達成するために複合メディア応用システムである。過去において、マルチメディア支援教授は伝統の視聴覚教授と同じ意味であった。伝統の視聴覚教授の特徴は、受講者に画像、音声、映像などの直接的な刺激を与えることが目的とされていた。教授形式は教師が主であり、統一された視聴素材を広げることによって、前の時代と比べて教授の生産性と効率性が高まった。二十世紀八十年代末までに、中国においては、マルチメディア支援教授というと、広い意味での多種のメディアを活用した視聴授業方法であり、主として、幻灯や録音機、ビデオなどの機械を用いて音声、画像などを表示することによって、教育目標に合致した情報受講者に伝達して授業を行うことであり、言語教育や実験の録画やドキュメンタリー映画を上映するものであった。今日になっても、伝統的な視聴メディアを活用した教授方法はまだ大きな役割を果たしているのが、現状である。九十年代前半においては、マルチメディア支援教授というよりも、「電化教学（教授）」という言い方が主流であった。その時、いわゆるマルチメディア教授（Multimedia Instruction、MMI）システムは、ただ視聴メディアを組み合わせた複合システムである、と思い込んだ教員が多かったようである。今日においては、狭い意味でのマルチメディアは、コンピュータ技術を基礎として、文字、音声、画像、映像など情報を伝送できるシステムを指すが、現状では、マルチメディア支援教授といっても、あくまでも正式の授業を支援する立場にあって、学生による主体的学習というメリットをうまく引き出すことはできずに、教師が一方向的に教え込むことが多い。

伝統のコンピュータ支援教授（Computer Assisted Instruction、CAI）は、情報量や情報を伝達ス

リード及びインターアクティブ性の活用などの面では不足点が多かったが、時間が経つにつれ、マルチメディア技術が大きな進歩を遂げており、伝統のCAIとマルチメディアを融合した。今日のマルチメディア支援教授というのは、コンピュータのマルチメディア機能を活用した教授を指すものは多く、予め製作されたマルチメディア支援教授コースウェアを使って教授を行うものを意味する。CAIという言い方とほぼ同じように使われており、時には、マルチメディアコンピュータ支援教授(Multimedia Computer Assisted Instruction、MCAI)ともいう。インターネットの普及によって、もともとCD-ROM形式であったコースウェアがウェブを経由して利用されることが増えたことによって、マルチメディアを活用した教授に言及することは、インターネットを介した学内教育及びそれ以外の遠隔教育をも含む場合が、多くなっている。

マルチメディア支援教授では、受講者がより主体的に学習内容に取り組むようになる。教授コースウェアは技術的には保存、伝達及び処理することができるため、授業効果の向上が期待される。コンピュータ支援教授のキーポイントは、コンピュータと学生の間、そして学生と学生、教師と学生の間自由にコミュニケーションが取れることにある。コンピュータには、情報を学生に伝える機能だけではなく、学生から情報を取り入れ分析判断を行う機能が揃っている。これがないと、学生による主体的学習を実現することはできない。教授側の意思が教授コストを通じて表現されるため、ソフトウェアの使用と開発はコンピュータ関連設備と同じく重要か、またはそれ以上重要であろう。

2. インターネットを利用した遠隔教育及び語学授業

マルチメディア活用による教授は簡単にいうと、四つの部分からなる。まずは、コンピュータを通じた教員と学生との交流、次はコンピュータによる学生の管理、三つ目はコンピュータのマルチメディアの活用による授業内容の展示、最後にインターネットによる遠隔教育の可能性である。そこで、さらには、教育資源の共有、情報の検索などを取り上げることができる。

伝統的な授業方法では、教員が自分の言語をより分かりやすく理解してもらうために、授業の補助措置としては、板書、掛図、模型及び実験などを借りて、教授の目的を達成する。マルチメディア活用教授は教授の目標に立って、教育対象を考慮に入れながら、マルチメディアを活用しながら、記号、言語、文字、音、図形、画像、映像などコンピュータで操作し、授業のデザインを行うことである。

マルチメディア支援教授では、コンピュータを操作することによって、それら伝統的手段に取って代わることである。こうすると、人とコンピュータとの間に、ある種のつながりが生じるようになる。基本操作の上で、教授の目標をはっきりし、授業の内容を如何に論理的マルチメディアの活用によって学生に円滑に伝達されているかを考えなければいけない。

マルチメディア環境は学生による主体的学習を中心としたものである。広義的には、パソコンを利用し、またインターネット接続したうえで行う学習全般をeラーニングと称することもある。実際には、マルチメディア支援教育はメディアによる遠隔教育と緊密な関係を持つものであった。

インターネットを利用する方式では、もし、教師と学生が同時にネット上のクラスにいて、同時に授業を行う形式を取れば、インターネットのインターアクティブ性から考えると、理論上、一つの教室での対面授業を行うと同じような教授効果が期待される。インターネットを通し、情報の伝達、一問一答式でのやり取り、学生による宿題の提出と教員による宿題のチェックなどが非常に簡単になる。その上、受講時間が不定期な学生も受講可能になり、少しずつでも、念入りに前回中断したところから学習内容に取り組むことができる。受講生が遠隔授業のコースウェアをそのままう一度受講することができるであろうし、マルチメディアコンテンツを自分の必要に応じて開くことができ、学習を繰り返し行うことが期待される。

実際の授業環境と高度に類似したインターネット上の教育環境を作るには、インターネットによる遠隔教育システムは、少なくとも三つの部分からなる。すなわち、バーチャル教室 (Virtual Classroom)、電子ホワイトボード (Electronic Whiteboard) 及び疑問解答システム (Answer System) である。

バーチャル教室は学校に実際にある教室を模倣し、インターネット上にいる者を交流させるために作られたバーチャル空間である。教える側と教わる側とをうまく協力し、如何に支障なくコミュニケーションを取れるかが研究のテーマの一つである。

ホワイトボード (Electronic Whiteboard) は、普通の教室にある黒板に当たる公共空間及び講師または受講生が発言する前に発言を編集する空間から構成されている。書き込みはそのまま電子データとして記録される点などがメリットである。

次は疑問解答システム (Answer System) である。これは、動画や音声問答システムのほかに授業内容を入手するための検索システムである。リアルタイムのやり取りに参加できなかったユーザーも、過去の発言を読み進むことによって、議論のプロセスを知ることができるという点もメリットである。

グループウェアを使用して共同作業を行う CSCW (Computer Supported Cooperative Work) は遠隔教育において重要な地位を占めている。コンピュータ同士を接続したネットワーク環境をベースに、複数の人間が相互にコミュニケーション、情報共有を行って、それぞれの役割を果たす。遠隔教育においては、合同でのコースウェア共用と共同開発政策、ネットソースの開発及びバーチャル教室の管理など、さまざまな方向から検討する必要がある、とされる。

インターネットを利用した遠隔教育は、時間の経つにつれ、多くの分野に亘っており、如何に授業をうまくデザインするかは重要な課題である。

遠隔教育に重要な地位にある外国語教育については、大いに議論され研究が行われている。外国語教育の目的は聞き取り、書き、話し、読み及び翻訳という五つの能力は基本の基本であり、言語の学習には大量な総合的な実践が必要である。当然、言語授業で講義だけする教師が少ないであろうが、学生とのやり取りがほとんど行わない教師がいるようである。文法、句型などについての説明は専門書の書いた通りに行うケースが多い。衛星放送が遠隔教育の主流であった時、特に言語が教育の主要な内容とされる場合、学生と教師との間に素早くやり取りをすることは物理的に不可能であった

め、対面授業と完全に同様な効果を取めるのが不可能とする受け止め方もあり、遠隔教育といいながら、年に数回学生と教師を対面させる試みもあった。

今、中国大陸の大学において、インターネットを利用した外国語授業には、次の二つの部分が含まれていると考えていい。一つには、教師の講義であり、単語の説明、文法やキーポイントについての解釈、本文についての講読を含む。衛星放送では、基本的に一方的な講義であった、今のシステムでは、だいぶ改善されているようにみえる。二つには、練習部分である。本文の講読と同時に進む練習と、本文の解釈が終わった後の練習があるが、ほかには一回の練習としては比較的量の大きい小テストもある。受講生に本文の内容を深く理解してもらうためには、教授の目標、本文にかかわる文化的、時代的背景を紹介する資料を用意する必要がある場合もある。伝統的な遠隔教育はこれぐらいの準備が整えられるならば、すでに十分とされていたが、インターネットを利用した遠隔教育では、教師と受講生とコミュニケーションをうまく取る機能が備わっているので、やり取りのシステムデザイン及び授業プランの設計についてさらなる研究が大切である。この部分は、インターネットのインターアクティブ性を生かすところであり、教師の講義と緊密につながっている。

実際には、コースウェアのほかに、教員として、学生に外国語のサイトにアクセスする習慣を身につけさせることも重要と考えられる。教員ができるだけ外国語で日記をつける習慣を身につけ、インターネットを活用し外国語の文章を大量に読み、外国人と直接にオンライン交流を行うように要求すればいい。そして、時おり学生にあるテーマについて外国語で発表してもらい、発表したものが小論文になり、それを回収し添削してから学生に返却し読ませ、学生にとっては貴重なオリジナルなテキストにもなる。ネイティブスピーカー並みの言語力を身につけるためには、長い期間の海外留学をしなければならないと、よくいわれるが、インターネットの世界がこれを転換させた。実際には、よく考えると、海外留学に行っても、現地の生活に入り込まないと、本国と同じか、またはそれ以下の学習効果も予想される。中国人の大学生は、キャンパスネットを通じて、世界中の人々と外国語で積極的にコミュニケーションを取り、外国語能力の向上に向けて一步一步と日々着実に進んでいる。この点においては、大学側から強力な支援のもとで、日本人の学生はもっと頑張るべきである。

3. 中国大陸部におけるマルチメディア支援教授戦略の流れ

中国は近現代においては、激しい社会的変動の中、民と官との両方からメディアを活用した教授の推進を試みてきた。今日においては、教員を中心にマルチメディア支援教授の有効な方法を模索するとともに、政府から、強いマルチメディア支援教授を推進する方針を打ち出している。しかし、この政策の実施によって、効果を得る一方、問題も多発し、真剣に検討する必要がある。

1919年に、中国商務印書館の製作した教育映画が上映されていた。これは、中国における近代的メディアを利用した支援教授の始まりだとされている。今日になって、大学教育機関がより効果的な教育活動を行うためには、マルチメディアを活用することが一つの有力な手立てと思われる。高度情報技術が急速に浸透しつつある今日の社会状況の中で、特にニーズの高いものと考えられる。

教育におけるマルチメディアの活用戦略を取り上げていることは、中国における教育の産業化、とりわけ大学教育部門の産業化という国の政策目標に合致したものである。マルチメディアをはじめとする情報通信技術の活用は、高等教育の充実に新たな可能性を開くものとして、大きな効果が期待できるものであり、それに、高等教育機関において円滑に実施されるための条件整備を積極的に図っていくことが求められている。

情報通信技術の発展は、従来の教育の教育形態の概念に大きな影響を与えている。今日において、教育を取り巻く環境は、教育の大衆化、情報化及びグローバル化の進行によって大きく変化を遂げてきている。この中でも、情報通信技術の進展はめざましく、世界各国において高度情報通信社会の実現に向けたさまざまな取組が活発になっている。中国は、経済的に発展途上国でありながら、先進国へのキャッチアップの一環として、教育やハイテク産業に大いに力を注いできた。教育分野においては、「現代教育技術」という言葉で表現したように、教育をテクノロジーとして捉えて、研究や政策立案を行う傾向がある。

1978年に実施され始めた「改革開放」政策によって、中国の教育事業は大きく発展し、特に教育体制の面において、特筆すべき成果が得られた。中国の教育規模が拡大し、教育と研究レベルも絶えず上昇してきた。その背景には、ますます活発になりつつあるメディアの活用戦略がある。八十年代初めから、北京、上海などの大都市では、メディアを利用した視聴覚教授は広く行われていた。八十年代半ばからコンピュータ知識の普及教育は中高校、場合によっては小学校でも行われ、コンピュータプログラム作りを内容とした授業も一部の高校や大学で実施され始めたのである。中国の大陸部において、大学の理工系は、当然早くから情報教育という授業があったが、文系では、全面的情報教育の始まりは九十年代に入ってからのものであると思われる。その前は、主として、英語教育における伝統のメディアによる視聴覚教育が圧倒的であった。こういった状況の裏には、情報技術や教育手法などの面における認識上の問題があれば、教育経費の不足という複雑な社会的、経済的要因もあった。

中国政府は、八十年代からいち早くコンピュータ支援教育（Computer Assisted Instruction、CAI）を国家戦略として実施し始めた。中国のコンピュータ支援教育研究は八十年代初めに始めたものである。華東師範大学によって開発された「パソコン BASIC 言語教授システム」がその中の一つである。1984年2月、中国政治の実力者であった鄧小平氏の「コンピュータの普及は子供から」との一言は、中国教育システムに大いに影響を与えて、コンピュータ補助教授（CAI）を全面的に導入するきっかけといっても過言ではない。この流れは、中国特有な社会体制、つまり、いわゆる「国情」によるものだ、といわざるを得ない。

1986年に、中国国家計画委員会はコンピュータ支援教授を「第七回五年計画」の重点項目と位置づけ、コンピュータ支援教授に関する研究を資金面においてより多い支援を行うとともに、「中華学習機」と名づけられた学生向けのパソコンを量産し始め、低価格に向けて、大きな一歩を踏み出した。そのため、小中学校、高校におけるパソコン所有台数が大幅に増加し、コンピュータリテラシー教育は大きく邁進したのである。1987年に成立した全国コンピュータ支援教授学会センター（中国語表記：全国計算機補助教育中心）と全国中華学習機教育ソフト審査委員会と合同でコンピュータ支

援教授ソフトの開発や応用の促進に取り込んだ。

九十年代に入ってから、コンピュータ技術は、これまでの教育機器の諸機能をうまく統括し、インターネット技術の利用、そして新たなマルチメディア支援教育、さらにマルチメディア支援遠隔教育を可能にした。中国政府はより大きな支援方針を打ち出し、従来の情報化教育の推進という考えのうえに立ちながら、教育の産業化の一環としても、全面的にマルチメディアを活用した教育の推進プログラムを実施した。前述したように、一般教育にかかわる用語として、中国では、マルチメディア支援教授（中国語表記：多媒体教学）とコンピュータ支援教育（中国語表記：计算机辅助教学）とがよく同じように取り扱われるが、コンピュータのマルチメディア機能を重要視するという点で、「マルチメディア支援教授」という言い方のほうが比較的多いようである。

九十年代後半から、高い経済成長率を背景に、中国は九年義務教育や基礎教育システムの整備を急ぐとともに、高等教育の大衆化と産業化政策を実施した。教育への膨大な需要に対し、教育資源が相対的に不足しているのが中国において恒常的な現象であるため、中国経済の高度成長、情報技術の進歩などは、この矛盾を解決する可能性を提供したのである。中国政府は、ネットワーク教育を発展させ、情報技術と教育技術を使いこなしてさまざまな教育資源を整合し、全面的にマルチメディア支援教授を推進する決意を固めた。中国政府は強い主導権を持ち、世界中情報革命の波に乗って、戦略的に情報化に取り組み、中国国内の全国規模の情報共有化に向けて、地域や各系統内の文献資源の共有化措置を絶え間なく実施し、基礎教育から高等教育まで、教育分野におけるマルチメディア支援教授を国家意思で推し進め、戦略的に情報化に取り組んできた。

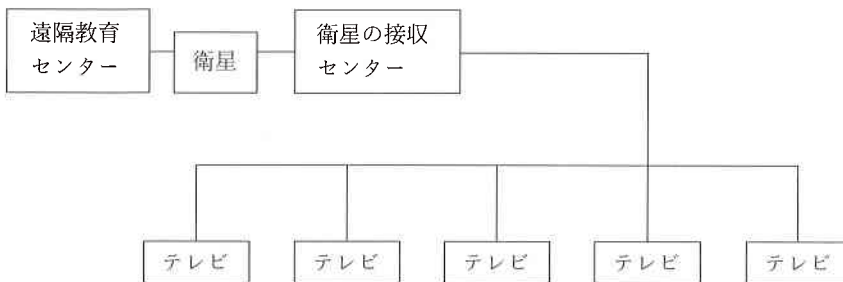
CAIの研究開発は、1993年から中国国家教育委員会（日本の文部省にあたる）による国家プロジェクトとして推進された。発布された「中国教育改革と発展綱要（中国における教育改革と発展綱要）」にラジオ・テレビ遠隔教育をさらに進めるとともに、マルチメディア教授など、いわゆる現代教授技術をいち早く普及させようという方針が強く打ち出されたのである。

1994年4月に中国大陸部はインターネットに正式につながり、10月に中国国家計画委員会が中国教育と科学研究ネットワーク（CERNET）の模範プロジェクトの成立を承認し、中国教育委員会（今の教育部、日本の文部省に当たる）がリーダーシップを発揮し、清華大学を含めて十校の大学に200人余りの研究グループの開発によって、1985年に国家の査収を経て正式に設立されたのである。今日になって、CERNETは全国センターを一つ、地域センターを八つ有する中国大陸部全土に行きわたるネットワークに成長しており、全国数百校の大学と研究機構とを結び付けている。

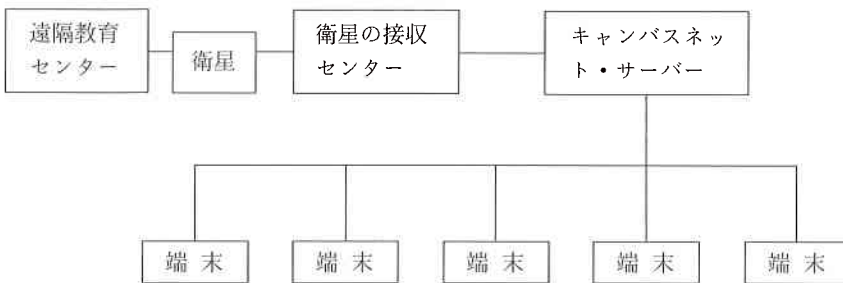
1998年4月までは既に300校以上の大学にマルチメディアネットワーク授業システムが整備された。1998年9月に、中国教育部は清華大学、湖南大学、浙江大學、北京郵電大学など四校の大学にコンピューターネットワークによる遠隔教育の試行を承認した。1999年、四校でインターネット遠隔教育を受ける学生数は9000人に上り、教育部はさらに北京大學、財政部会計通信教育学校、中央ラジオ・テレビ放送大学にインターネットによる遠隔教育を実験する許可を下した。1999年6月に中国政府が批准した「關於深化教育改革、全面推進素質教育的決定（教育改革を深化させ、全面的に素質教育を推進するための決定）」においては、教育手段の近代化、水準の向上及び教育情報化が取り上

げられている。中国教育部が2001年に批准した「关于加强高等学校本科教学工作提高教学质量的若干意见（大学学部教育の品質を向上させるためのガイドライン）」によると、国家の重点大学（一流大学）では、マルチメディアを活用した教授が総授業数の30%以上占めなければいけない、また、一般大学でも15%以上の水準に保つ必要があるとされている。

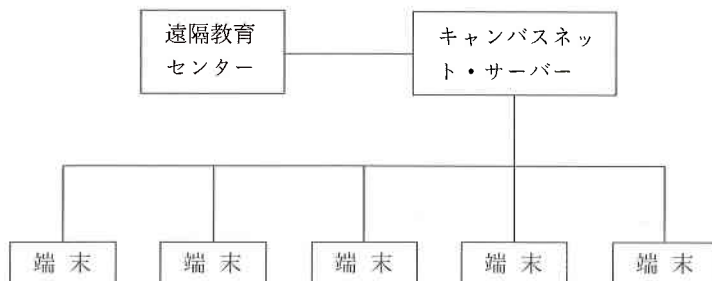
遠隔教育の面において、具体的には、1978年に、中国政府は教育事業を発展させるために、テレビ、ラジオ放送などの教育の手段として活用するように遠隔教育の方針を打ち出した。1978年に、中国中央ラジオ・テレビ放送大学が成立し、チベット自治区を除いて、中国大陸部の省、直轄市、自治区レベルでは放送テレビ大学を有していた。省の下に地区、さらに県レベルまで放送大学の授業が行われていた。1986年に、衛星テレビ教育チャンネルが開通するとともに教育テレビ局が設立され、大規模の遠隔教育ネットワークが確立されたのである。1990年代に入って、旧ソビエトの崩壊とア



図一：伝統の遠隔教育



図二：衛星とLANと複合した遠隔教育システム



図三：インターネットによる遠隔教育

アメリカが湾岸戦争における圧倒的勝利を収めたため、アメリカにおける軍事技術の民用化が一層進められた。インターネット技術はその代表的なものであり、それからの発展は九十年代のアメリカ史上なかった好景気に伴ってハイテクバブルといわれるまで発展していた。世界のこの流れは中国に大きな影響を与え、中国大陸部におけるインターネット基盤建設を加速させたのである。

インターネットを通じた遠隔教育はますます大きな地位を占めている。今まで、中国大陸部における遠隔教育は三つの段階をたどってきたと思われる。まずは通信教育であり、今日と比べると、かなり小規模であった。第二段階はラジオ、テレビなどを利用して行う教育であり、第三段階はコンピューターネットワークやマルチメディアを運用し、デジタル化環境の中で交互に行う教育である。今、主として、中国の遠隔教育は衛星放送単独方式（図一）、衛星とインターネット融合方式（図二）及びインターネット単独方式（図三）、という三つの方法で行われている。昔から、遠隔教育を受ける側にある受講者が地域によって一箇所に集まって、集散的に授業を受ける伝統がある。衛星放送単独方式の時代では、みなでテレビを視聴するのが一般的であったが、今日においても、受講者はマルチメディア教室やコンピュータ室で遠隔教育を受ける光景は珍しいことではないのである。というのは、インターネットはまだ全国的に普及したといえない状況の中では、キャンパスローカル・エリア・ネットワークを開通することによって、その不足を補うことにしているからである。

伝統の衛星放送単独方式の遠隔教育は一方向的な教育であり、学生から教師への質問や宿題の提出にコストがかかり、受講生と教師との交流が大いに抑制される。この方式では、教師が受講生の反応を素早く受けるのが不可能である。今日では、伝統のアナログ信号からデジタル放送に切り替えることを視野に入れながら、急スピードで衛星放送に変化が生じているところだが、インターアクティブ性の欠如ということを解決できなければ、伝統的な衛星放送による遠隔教育がいくら技術の進歩をしても、長期的な立場に立てば、衛星放送単独方式はだんだん撤退する傾向にあると考えられる。しかし、インターネットによって、伝統の衛星放送の弱点を解決することが期待できる。

4. 大学教員の現代教育技術に関する訓練プログラム

中国政府による強力なマルチメディア支援教授の推進方針は、教師が授業に臨む姿勢に大いに影響している。特に、大学教員はそうである。

中国大陸部において、大学教員は、大きく区分して、助手、講師、助教授、教授に分けられており、政府及び大学当局には教員に対して厳しい評価制度があり、かつて限定的であった任期制を全面的に実施するとともに、上位のランクに進むためには評定機関の細かい評価を受けなくてはならない。しかし一定の論文数と著書が要求されるだけではない。外国語能力試験及びいわゆる現代教育技術試験に一定の点数が求められるほか、授業に関する非常に詳しいルールが決められており、二ヶ国語で（一般的に英語と中国語）授業を行うことができるか、そして授業中マルチメディアを活用することができるが実際には活用したことがあるかどうかは、今日では既に重要な指標の一つになっている。評価方法を大きく三種類に分けることができ、すなわち、「学評教」、「教評教」、「教評学」とい

うことである。つまり、学生による授業評価のほか、教員同士、実は教務による授業評価及び教員の自己評価である。大学では、教員が授業プランを提出し、大学当局の検査をうけるのが義務であるが、マルチメディアコースウェアという形での提出が提唱されている。検査する側が十分な時間を取れないこともあって、授業全体への評価は、形式上のものだけに惹かれる可能性が大きく、特に、マルチメディア支援教授が教育水準の評価指標とされる場合に、この傾向は一層強まると考えられ、授業中マルチメディアを活用したか否やとの違いによって、授業に対する評価値が大きく上下するケースもありうる。結局、Akerlof (1970) のレモン市場のように、評価システムの根幹が崩れる可能性さえある。当然、この問題は、マルチメディア支援教育に限らず、教育全体、または経済全体に亘ることであるため、中国における縦社会の本質に対し、さらなる改革が求められる。

中国大陸部における教育研究の発信地である北京市は、いち早く具体的なマルチメディア支援教授に関する方針を打ち出している。北京市政府に直属、または管理下にある大学の教育技術試験は、教育の方針にしたがい、低いレベルから高いレベルまで、一級（初級）、二級（中級）、三級（高級）という三つのレベルに分かれている。助手をも励ましてマルチメディア支援教授の実施を行わせるが、市政府レベルの規定がない。2005年1月1日から、これまで実施してきた一部の教員に対する試験免除制度の実施を改め、大学教員が講師またはそれ以上の職階へ昇進しようとする際に、試験が求められる。中級職階、すなわち講師への昇進を申し込むには、一級試験が要求される。その内容は Word、Excel、PowerPoint に関する理論的試験である。講師から助教授へ昇進するためには、現代教育技術二級という資格が必要となる。試験がコンピュータで行われ、二部からなる。まずは理論的問題で、その後は三つのソフトを運用する実技テストである。ソフトは、Dreamweaver のほかに、Photoshop、Flash、Authorware の中から二つを選んで受験する。筆記試験及び実際のコンピュータ操作、あわせて二科目を受験しなければならない。実際に実施された試験は、入学試験や社会に公開された資格試験ほど難しくないといっているであろうが、大多数の文系教員にとっては、試験に合格するためには、時間と金銭的な支出は大である。三級試験は強制的ではなく、試験を受ける対象は、教授経験及び現代教育技術に関する知識が豊かで、しかもインターネットコースデザインに熱心な教員、とされている。

試験は原則的にすべての大学教員を対象として、マルチメディアの基本的操作の理解と活用を目的とした講習プログラムを設定し、マルチメディアを活用して、画像、映像、音声情報の処理に関するカリキュラムを取り入れた授業を行わせるものである。中国大陸部の国立、公立大学においては、一般教員の退職年齢は 60 歳とされているが、1960 年 1 月 1 日以前に生まれた、つまりだいたい 55 歳以上の教員に対して、ルールを緩めて、Word などで簡単なプレゼンテーションをしてもいいとしている。通常、試験に合格させるために、それなりの訓練プログラムに参加するだけでもかかる授業時間は長く、研修へ補助するどころか、それなりの出費も大学によってはほとんど自費とされる場合が少なくない。

5. 大学のマルチメディア活用戦略がもたらした問題

数多くの大学には、域内ネットワーク、遠隔教育センター、CAI コースウェア政策センターなど、マルチメディアにかかわる部門が設けられている。

マルチメディアとそれを取り巻く環境の変化は、伝統的な教授手法に大きなインパクトを与え、それなりの変化をもたらした。中国では、教育当局の強力的なマルチメディア活用戦略によって、マルチメディアが有力な教授手段として、ますます大きな役割を果たすであろうと期待される。大学、とりわけ北京、上海などの大都市では、全授業でマルチメディアを活用する大学があるようである。大学教育の産業化によって、大学の規模拡大が著しく進んでいる。マルチメディア設備が揃う教室が一般的となり、特別な場合を除くと、黒板とチョークを一年中まったく触らないケースは少なくない。

しかし、一方、この流れにさまざまな問題も潜んでおり、マルチメディアを利用して授業を行っただけで成果を上げているかどうかに対しては、疑問の余地がある。マルチメディアというとテクノロジー的な優位性の側面が注目されがちであるが、その優位性ゆえに生じた問題に注意を払う必要がある。そのほか、技術的な優位性があるからといって、実際の教授と学習との間に内面からの緊密な関係がないと、まったく意味がないし、「マルチメディアの乱用」ということによって正常な教授システムに障害をもたらす恐れさえある。

全体からみると、数多くの教員、特に 50 歳台、またはそれ以上の文系教員はコンピュータに関する知識が乏しく、多くの教師がせいぜい Word が使えるレベルにあり、PowerPoint の使用は一部の人に限られる。マルチメディア支援教授ではよく使われる Authorware のようなソフトについての理解が浅いのが現状である。

マルチメディア・インターネット技術を教育に取り入れる真の目的は、教育の品質を向上させることにあろうが、政府主導のマルチメディア活用政策は、教育の産業化が背景にあり、教育の収益率の向上という思惑もある。

まずは、教育の効率化である。教育を受ける側にある学生にとっては、限られる時間に学ぶべき知識を理解し、それぞれの社会活動に適應する技能を身につけることである。一方、教える立場に立つ教員が限られた時間内に計画した通りにまたはそれよりも順調に教育目標をより確実に達成することを図るものである。

マルチメディア支援教授は伝統教授手段になかった直感性、面白さがあることで、教える側と教わる側が両方ともに積極的授業に取り組み、教授過程中にある難点を比較的簡単に解決することに寄与する。しかし、だからといって、一方的に大いにマルチメディア支援教授手法を授業に取り入れるべきかという点、必ずしもそうではないであろう。マルチメディア支援教授にはさまざまな問題があるように思われる。問題の一部は、このマルチメディア支援教授という比較的新しい手法を導入する段階に発生しがちなものであろうが、ほかには、マルチメディアを使う必要がないか、または使うと学習に逆効果をもたらす場合もありうる。

デジタル化によって、情報の大量収集が可能になり、知識の伝達がよりスムーズに行われること

が可能になる。マルチメディア活用による教授は、学生の学習意欲を引き起こすことができ、授業における効果の向上が期待できるものである。マルチメディアを活用することによって、授業の難関、難問をより簡単に解くために役立てることができる。特に言語の授業（CALL など）、一部の理工系科目ではシミュレーションによる表現は非常に効果的である。

一方、過剰なマルチメディア運用を避ける必要がある。伝統的な教授方式は長年に亘って実践を積んだものである。マルチメディア活用による教授は、大多数の授業ではマルチメディアはあくまでも補助という立場にあるほかない。しかし、数多くの教員は、マルチメディア機能を有するコンピュータを単にテキストを製作する機器と見なし、黒板などの使用はだんだんなくなっている。

授業道具として、すべてそれを使ってしまうと、返って学生の注意が無関係のところ引かれる可能性は大きく、授業の効果を低めると考えられる。時には、一部の教員は学生の反応や理解の度合いを顧みず、完全に自己流で授業を進めるケースもみられる。マルチメディアの比較優位性をできるだけ発揮させるために、必要以上の情報量を学生に教え込む場合もある。教員自身自身のマルチメディアの特徴についての理解が足りないのが原因の一つであろうが、教員は形だけの授業を追求し、実質上のサボタージュであるといわざるを得ない。

中国大陸部において、マルチメディア支援教授のもう一つの問題は、授業用ソフトウェアへの選択である。市場に出回っているソフトウェアは、実質上電子読本のものにすぎないものが少なくない。問題集などのデータベースもあるが、それは、宿題としてはいいかもしれないが、マルチメディアと直接な関係を持つとはいえず、紙で造られた問題集とあまりにも変わらない。マルチメディアを活用した授業におけるメリットの一つには、比較的統一されたソフトウェアを使用することによって、教員が授業にかかる物的及び精神的コストの軽減である。しかし、これこそ教員が単純なコンピュータ操作に陥って、授業の真の目標さえ見捨てる恐れが潜んでいる。授業というのは、あくまでも学生という人間を対象とするため、学生に個人差が大きいことに注意をよく払い、十分に対応できる方法を考える必要がある。さもなければ、機械だけでも十分であり、教員は不要になるであろう。

マルチメディア支援教授の効果を評価するのも簡単なものではない。それを確実にを行うためには、全面的な評価システムを作り、多数の指標について、総合的な比較分析を行い、実証研究をする必要になる。おおざっぱに言えば、教育目標への達成度合いこそ、授業効果に対する判断の終極の基準だと思われる。基本的な教育方針に従い、技術的に正しい情報をスムーズに学生に伝えることに成功すれば、マルチメディア支援教授が効果的だと、ようやく判断することができるのであろう。

今のマルチメディア支援教授のコースウェアには、受講生の好みに対応できるものが少ない。授業用コンテンツの順序が固定され、一旦授業を始めると、受講生の実際に学習内容について理解した状況に応じて調整することが難しい。このことを考えると、一部のマルチメディア支援教授といわれる、または大学当局によって認められるが、実際には、まだ八十年代のレベルに止まったままであるものといわざるを得ない。

映像や音声がスムーズに流れることが、今日におけるマルチメディア支援教授の特徴の一つであるが、これよりも遥かに重要なのは、マルチメディア支援教授自身にあるインターアクティブ性に

よって、学生自身による自習的学習が可能になることである。しかし、中国大陸部における大学の一般的状況をいえば、マルチメディアを利用することは、教師による教授が主とされており、学生の学習を主眼に置いていないため、学生が主体的な学習を十分に意識しているとはいえない状況である。これは、素質教育を進めるため、という中国政府のマルチメディア活用戦略の初志とかみ合わないことである。

6. 真のマルチメディア支援教授へ

今日においては、知的情報が経済の発展に非常に有益且つ重要であることがすでにだれにも認識されている。教育におけるマルチメディアの活用については、今後とも、中国教育のさらなる充実を図るとの視点に立ち、各大学における活用の状況等を踏まえつつ、その活用の在り方や制度上の諸問題について、適切な見直しなどを行っていく必要がある。

中国のマルチメディア支援教授は政府の後押しによって、急速に普及されたとはいえ、品質面ではさらに発展する余地が残されている。マルチメディア支援教授をあるべき姿として行うには、知能CAIシステム、即ち、ICAIシステム（Intelligent Computer Assisted Instruction System）またはITS（Intelligent Tutoring System）という概念及び操作を押し広げる必要がある。ICAIは人工知能をコンピュータ教授に取り入れることで、マルチメディアを活用した教育方法におけるインターアクティブ性という最大の特徴を生かしたものである。

ICAIシステムは伝統CAI系統的な欠陥を克服するうえで、学生の学力に関する基礎、レベルと能力を把握し、それに基づいて、それぞれの学生が持つ特徴に合致した指導を行うことができ、学習内容を自由に変化させるとともに、学生はコンピュータ言語を使わずに、人間の自然言語でコンピュータとコミュニケーションを取り、学生の個性に合わせた答えを出すことができるシステムが必要になる。その上、システムは、学生の学習プロセスに発生した誤りを診断する機能を整え、誤りの原因を分析して、解決案を提供するこることに止まらず、これを基礎として、次第にシステムに判断の経験を蓄積して、システム自身に学習能力を有することが望ましい。

これらの機能を備えるシステムを構築するには、ICAIには学生モジュール、教師のモジュール、データベースと人工知能インターフェイスなどを備える必要になる。学生のモジュールには、学生の「入学時」の学力水準を数値化したうえで、それを保存し、学生とコンピュータとのインターアクティブの経歴を記録する。そうすると、学生の学習状態に応じ迅速な調整を行うことが可能になる。教師のモジュールは学生のモジュールの状況に対応してインテリジェント化の教授方策を作り出し、学生の学習効果について判定を行い、改善する方法と意見を出す。知能のインターフェイスを通じて、人間の自然言語を理解することができ、マンマシン対話の実現に結びつく。

ICAIの構築は学際的な学科であり、コンピュータ科学のほかに、認知心理学、一般教授法などに直接にかかわっている。ネットワーク技術の改善、マルチメディア技術とシミュレーション技術との複合技術であるバーチャル・リアリティー（Virtual Reality）技術などはICAIの構築に大いに役立つ