

履修登録システムの構築と運用

檜垣 泰彦^{†,††} 阿由葉 努^{††} 土屋 俊^{†††}

Construction and Operation of Registration System in University

Yasuhiko HIGAKI^{†,††}, Tsutomu AYUHA^{††}, and Syun TUTIYA^{†††}

あらまし 大学環境においても情報化が進んでいるが、履修登録システムにはルールが非常に複雑で、正確な履修登録判定を行うことが難しいという問題がある。精度の高い判定を行うためには多様な情報を備えたデータベースと、ルールを厳密に記述したプログラムが必要となり、高価なシステムとなる。この解決策として本研究では、履修登録エラーがある状態でも履修登録申請を提出できる仕組みを導入した。これにより、提出された申請の内容を教務担当者が確認できると同時に、システムに存在する各種の問題点を効率的に絞り込み、把握できるスクリーニング機構を実現できた。書類による履修登録のときと同様、担当者が学生の提出してくる問題のある履修登録申請にも目を通すことができる結果、各学生の事情を配慮して個別に指導するなどきめ細かな対応を行うことが可能となった。また、2002年度、2003年度と実際に学生約3,200人に対して運用した際の、学外からの履修登録を可とした場合のアクセス件数、履修登録期間中の提出件数、アクセスが集中する時期のサーバのロードアベレージなどの具体的運用データを示した。

キーワード 学務情報システム、スクリーニング機構、履修登録エラー、運用データ

1. ま え が き

大学環境における情報化も着実に進んでおり、Webや電子メールを活用した学務情報システム、学生情報システムなどが各方面で導入されつつある[1], [2]。本工学部でも2001年度後期から2003年度にかけてそれまでの紙による履修登録を改め、学生に、より質の高いサービスを提供すべく独自開発のシステムにより履修登録を実施してきた[3]~[5]。本論文ではこの履修登録システムの構築とその運用について述べる。

履修登録時のエラーの判定はルールが非常に複雑であり、質の高いデータとプログラムがそろって初めて正しい判定が可能となる。ここでいうデータとは例えば履修課程や履修条件のことであり、学科の改組・新設などの再編が繰り返し行われる中、数少ない過年度生

の情報も含むすべてのルールを表すデータを準備することは不可能に近い状態にある。また、ルールをデータとして表現できない部分もあり、それらは個別のプログラムコードとして埋め込まれることとなる。つまり、現在運用されている特別なケースまでをすべて含むデータとプログラムを備えることは困難に近いのが現状である。現在稼動しているシステムにおいても、特殊なケースは窓口対応としているなど、厳密に対応できていないシステムも見られる。本論文では、履修登録エラーがある状態でも履修登録申請を提出できる仕組みを導入することでこの点の解決を図り、学生が窓口に並ぶことなく履修登録を完結できるなど、質の高いサービスを提供できるシステムの構築を目指す。

学外からの利用を可とするか否かの判断や、それに関連して、円滑に運用できるシステム実現のための同時アクセス数の見積りも重要である。このように既に一般化している履修登録システムであるが、解決すべき点は多く存在する。また、設計にあたって参考となる公表された具体的運用データが少ないのが現状である。本論文では、学生数約3,200人に対して運用を行った際の具体的運用データを示す。

[†] 千葉大学大学院自然科学研究科情報科学専攻, 千葉市
Division of Information Science, Graduate School of Science
and Technology, Chiba University, 1-33 Yayoi-cho, Inage-ku,
Chiba-shi, 263-8522 Japan

^{††} 千葉大学工学部, 千葉市
Faculty of Engineering, Chiba University, 1-33 Yayoi-cho,
Inage-ku, Chiba-shi, 263-8522 Japan

^{†††} 千葉大学文学部, 千葉市
Faculty of Letters, Chiba University, 1-33 Yayoi-cho,
Inage-ku, Chiba-shi, 263-8522 Japan

2. 履修登録システムの問題点と解決策

2.1 一般的履修登録の流れ

一般的履修登録システムにおける履修登録の流れはおよそ次のとおりである。

(1) 学生個人を特定できる ID、及びそれが学生本人であることを認証するためのパスワードを指定して履修登録システムにログインする。

(2) 登録しようとしている授業を特定し指定する。具体的な方法としては、(a) 時間割形式で表示された一覧から授業を選択、(b) 学科・学年ごとの授業一覧から選択、(c) 各授業に割り振られたコードで入力、などが一般的である。

(3) ルールに基づき登録の正当性のチェックを受ける。チェックされるべき点としては、(a) 登録しようとしている授業間の開講時限の重なり(曜日時限チェック)、(b) 既に履修済み科目の 2 重登録(既修チェック)、(c) 前期の授業を後期に登録しようとするなどの登録時期の正当性(開講期チェック)、(d) 年間履修登録単位数の制限(制限単位数チェック)などがある。

(4) 登録操作を行う。

(3) における曜日時限チェックにおいては、再履修である場合は重なっていても登録を許すケースもある。そのほか、履修登録システムがもつべき機能として、過年度の履修状況の参照や進級卒業の要件をチェックするため、科目区分ごとの単位数の集計表示の機能も必要である。

2.2 正確な履修登録判定の難しさ

履修登録処理は、処理内容を記述したプログラムによって実現されているのはもちろんであるが、それだけでは機能することはできない。参照すべきデータとして、正確な情報をもった表 1 のようなテーブルが備わって初めて機能する。実際にはこれらのほかにも必

表 1 履修登録に関連するテーブル
Table 1 Tables for registration.

テーブル名	項目
科目	科目コード, 科目名称, 単位数, 開講学科コード
授業	授業コード, 科目コード, 担当教員コード, 成績受理日, 開設年度
時間割	授業コード, セメスタ, 曜日, 時限, 開講区分
履修課程	入学年度, 学科コード, 科目コード, 科目区分
制限単位数	入学年度, 学科コード, 単位数
履修	履修年度, 学籍番号, 授業コード, 評価
学生	学籍番号, 入学年度, 所属学科コード

注) 学科コード, 教員コードなどのコードテーブル関係は省略した。

要な情報があるが、ここでは履修登録の動作に直接関係ある情報のみに簡略化し、単純化した分かりやすい名称で示した。大幅な履修課程改変時に、少数しか在学していない過年度学生用のデータを省略した場合や、特別な事情のある学生への特例を認める場合、実際は運用でカバーできる開講時限の重なりなど、設計時に想定した処理だけでは対応できないケースがどうしても発生する。データの不備をなくし、いかなるケースにも対応できるよう、データとプログラムを整備することでこれらの例外を小さく抑えることも可能であるが、現実的にはそのために必要となる莫大なコストを無視できず、ゼロにすることは困難であるといえる。また、履修登録に関連するルールは毎年のように変更が加えられ、それらに追従するためのシステム維持費も膨大なものとなる。

2.3 解決策の提案

通常の履修登録システムでは、2.1(3)におけるチェックをすべて満たさないと履修登録できない仕組みとなっている。発想を転換し、ひとりのチェックを実施し、その結果(エラー内容)を利用者に提示した上で、それでも利用者がその履修登録を行いたいと判断した場合はそのリクエストを処理担当者にそのまま伝えることのできるシステムを本論文で提案する。このようなケースでは、データやシステムの処理の方に問題があり、チェックがエラーとなっている場合が含まれるからである。リクエストを受け取った担当者は、その内容をチェックし、リクエストに問題がなければそのリクエストを受け付ける。データやシステムの処理に不備がある場合は、それを修正しチェックがエラーとなる状況を改善していくことができる。

このように、エラーがあってもリクエストを処理担当者に伝えることができる仕組みとすることにより、処理担当者がエラーの状況をより具体的に把握することができ、よりきめ細かなサービスを提供できると同時に、結果的に、これらの仕組みは、プログラムやデータの不具合も含む、システム全体の問題点を効率的に洗い出すことができるスクリーニング機構として機能することが期待できる。

3. システム設計・実装

3.1 システムの目的・要件

本システムの目的は、紙による履修登録を行っている状況を改善し、学生に、より質の高いサービスを早期に提供することである。そのための要件として以下

を設定した。

(1) 学生が窓口に向くことなく履修登録を完結できること；本学部においては、主として夜間・土曜日の授業を履修するコース（Bコース）の学生の場合、来校しても毎日ほとんどの時間帯が授業で埋まっており、窓口を訪れる時間の確保が困難な現状である。また、昼間の授業を主として履修するコース（Aコース）の学生にとっても、昼間は授業に集中することができ、より質の高いサービスの実現につながる。

(2) 規則をシステムに合わせ変更・制限するのではなく、これまで同様、従来の規則（特例も含む）で運用できるシステムとすること；学生にとって、本システムの導入が原因で、これまでと異なるルールで運用されることになるのは迷惑であり、処理の合理化に伴うサービスの低下ととらえることができる。

(3) 短い準備期間でできるだけ早くサービスを開始できること；紙による履修登録の現状をできるだけ早期に改善すべきである。

(4) 自宅からも24時間利用可能とすること；オンラインショッピング、インターネットバンキング等が一般化している現在、より質の高いサービス実現のためには、インターネットから24時間利用できるシステムとすべきである。

(5) 常に安定した動作が行え、アクセス集中時にも適切なレスポンスが得られること；利用者の動作環境に制限があったり、動作不安定になる場合があると、質の高いサービスの提供は望めない。また、サーバの負荷分散の目的で、学年や学科別にアクセス時間を制限することはサービスの低下ととらえることができる。

3.2 システム設計

一般的に新しい履修登録システムを導入するためには、十分な準備期間と、特別な予算措置が必要である。3.1の要件(3)に対応するため、本システムでは高価で、かつ導入までに仕様の検討、必要なデータの整備などがすべて完了する必要がある商用パッケージの採用は行わず、筆者らで独自にシステムを開発し、短期間でできる限り安価に履修登録システムを構築することとした。業務に直接関わる担当者が直接設計に加わるため、仕様策定や設計時間の短縮ができる、これまで開発してきた他のシステム[6]の開発経験や既存モジュールを活用し、短期間で開発できる等の理由のほか、必要最小限の基本機能と基本データから運用を開始し、必要な機能やデータを順次整備していく構築とすることで、早期に履修登録を実施できる見込みが

あった。また、独自開発することによる教育・研究上の効果も期待できた。

3.1の要件(1),(2)を満たすため、2.3で述べた、履修登録エラーがある状態でも履修登録申請を提出できる仕組みを導入し対応することとした。旧来の紙の書類との連続性が分かりやすいようにこの仕組みを実装するため、作成、提出、受理、処理、廃棄などの概念を設けた。3.7で詳細を述べる。他システムでは履修登録時のチェックがエラーで窓口対応となるような場合でもこの仕組みでは、窓口を訪れることなく、学生にきめ細かなサービスを提供するように構成することが可能となる。その詳細については3.8で述べる。この仕組みの導入により、一部不完全なデータや処理が含まれていても、スクリーニング機構で順次整備していくことが可能となるため、要件(3)の実現にも効果的である。このような独自の仕組みの導入のためにも、独自開発のシステムとする方が好都合であった。

要件(4)を満たすため、十分なセキュリティを確保できる設計とした。詳細は3.4で述べる。要件(5)については、信頼性の高いソフトウェアで構成し、かつそれらを適切な設定で利用することにより対応した。また、無償で利用できるソフトウェアで構成することにより、システム開発に掛かる費用を抑えることが可能となった。詳細は3.5で述べる。

3.3 システム構成

図1に本システムの構成図を示す。システム全体としては、授業編成支援やシラバス作成・公開の機能[5],[7]、電子メールによる連絡用メーリングリストの機能[8]などをもっている[3]が、この図では履修登録に関連する情報の流れを中心に示した。“教務システム”は、本部によって運用され、全学の学務・教務

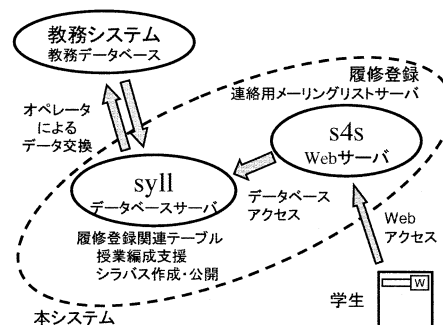


図1 システム構成

Fig.1 System configuration.

係が使用しているもので、1996年稼動以来、部分的な手直しを加えながら今日に至っている。

本システムは syll, s4s という二つのサーバを中心に構成されている。syll は教務システムのフロントエンド的な役割を果たすサーバで、オペレータ（教務係）が Web ページを操作することにより、教務システムとの間で必要なデータの交換が行われる。syll には表 1 で示したものに相当するテーブルが保持されており、履修登録時はデータベースサーバとして機能する。s4s は連絡用メーリングリストサーバ、アーカイバとして利用され、すべての学生がアカウントをもっており、利用者専用ページをアクセスできるようになっている。履修登録時には s4s は Web サーバとして機能し、その利用者専用ページ内に設置した CGI プログラムで syll のデータベースに接続して処理を行うことになる。

図 1 の構成の特徴としては以下の点が挙げられる。

(1) 基幹システムとは別システムとして構成した；本システムは基幹システムである教務システムとは別システムとして独自に構築し、教務システムがもっていない機能を順次補う方法で拡張を行ってきた。このような構成をとった結果、教務システムの稼動時間に依存することなくサービス時間を設定でき、また履修登録によるアクセスの集中の影響などが重要な基幹システムである教務システム側に及ぶのを防ぐことも可能となる。更に、基幹システムへ手を加える大規模な変更を行うには、十分な作業期間と多くの費用が必要となるが、本構成をとった場合は、システム開発のために基幹システムを停止する必要もなく、本システムの組込みにより、基幹システムが動作不良を起こす危険性もない。反面、利用者（学生）が教務システムとは別システムとして構成されている点を理解せずに利用するようなことがあると、教務データベースへの反映がオンラインでなく時間遅れを生じることに對し混乱を招く結果となるので注意して実装する必要がある。

(2) 連絡用メーリングリストサーバ機能を併設した；本システムの最大の特徴である履修登録エラーがある状態でも履修登録申請を提出できる仕組みを有効に機能させ、最終目標である質の高いサービスを提供するためには、担当係が学生と密接に連絡をとることが重要である。

(3) 授業編成支援やシラバス作成・公開機能を併設した；どの授業を履修するかの判断のための資料となるのがシラバスであり、両者を密接に関連づけた構

成とすることで、履修登録システムの操作性を上げることが可能となる。

3.4 セキュリティへの配慮

現行のシステムでは履修登録に利用できるパソコン端末を学内の端末のみに制限するケースが多く見られる。その主な理由は (1) 部外者の利用の可能性が低くなり、より高いセキュリティを確保できる、(2) 同時接続数を把握しやすく、必要なサーバの能力の見積りを行いやすい、(3) 更に特定の演習室のパソコンなどに制限することで、対象となる OS を限定でき、利用者サポートを行いやすくなる、などである。その反面、学生にとっては (1) 多くの場合、制限された数の端末に待ち行列を作る必要があり、履修登録自体は Web 化されても端末の利用のために並ぶという現実との間にギャップがある、(2) これまで自宅の自室で行っていた授業履修計画作成作業を、限られた時間に限られた場所で行うように強制することになり、サービス低下につながる、などのため、せっかくの Web 化の効果が発揮できない。

s4s の Web サーバでは、3.1 の要件 (4) を満たすべく、学外からのアクセスも許可し、syll, s4s とともに 24 時間稼動としている。十分なセキュリティを確保するため、SSL (Secure Socket Layer) の導入、信頼性の高い OS・ソフトウェアの利用、OS やアプリケーションにセキュリティホールがない状態に保つ運用を実施、適切な ID・パスワードに関する運用等の対策を行っている。OS やアプリケーションにセキュリティホールがなく、SSL により通信の暗号化を行っていても、ID とパスワードの配布や再発行が適切な手段で行われないと不正なアクセスを受ける結果につながるおそれがある。本システムでは、ランダムパスワードを学生証により本人確認の上配布するなど、手間が掛かっても慎重な扱いを実施している [4]。

3.5 ソフトウェア・ハードウェア構成

これらのサーバの OS は FreeBSD [9] であり、Web サーバとして apache [10]、データベースマネージャとして PostgreSQL [11]、通信内容の保護のために OpenSSL [12] などの無料でかつ信頼性の高いソフトウェアを使用している。ハードウェアは syll 用としては CPU が Pentium III 800 MHz 二つの SMP 構成、メモリ 1 GByte、s4s 用としては CPU が Pentium III 1 GHz 二つの SMP 構成、メモリ 2 GByte のものを利用している。ディスク装置は RAID 5 構成である。

締切りや期間を設定して受付を行う期限付きエント

りの場合、締切り直前に大部分のエントリが集中しピークを生じる。自宅からのアクセスも許可した運用を行う場合には特にアクセスの動向を十分に調査する必要がある。参考とすることのできる具体的なアクセスのデータがあればそれをもとに動向を分析し、適切な設計・運用を行うことが可能となる。本システムでは試作版による試行実施期間を設け、更に、2003年度の運用にあたっては2002年度のデータ [4] をもとに調整を行った。

3.6 教務システムとの関係

syll と教務システムとの間の履修登録関連情報の受渡しの詳細を図 2 に示す。syll に蓄積された履修登録の申請データを適当なタイミングでオペレータがローカルの PC に管理者用 Web ページを使ってダウンロードする。その申請データを、教務システムの外部データ読み込み用ユーティリティを使って教務システムに転送し履修登録処理を行う。その結果が反映された履修テーブルの内容をローカルの PC にエクスポートし、それを syll にアップロードして履修テーブルを更新する。

3.7 履修登録の流れ

本システムでは Web による履修登録を旧来の紙の書類に見立て、表 2 に示すような作成、提出、受理、処理、廃棄などの概念を設けた。履修登録は図 3 に示すような流れとなる。学生は教務データベースに追加登録、またはそこから削除すべき授業のコードを本システムで入力することにより追加・削除コードのリストを“作成”する。このときコードからではなく、時間割や授業一覧から作成することも可能である。作成したリストに基づき“確認”により内容を確認すると同時に、開講時限の重なりや削除授業のチェックが行われる。この結果 2.1 (3) で示したようなエラーが検出されればその旨、メッセージが表示される。この場合は作成のステップに戻り修正を行う。その後この

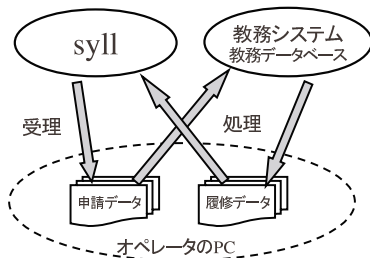


図 2 syll と教務システムとの関係
Fig. 2 Syll and the academic affairs system.

履修登録申請の“提出”を行う。教務係は適当なタイミングで“受理”の処理を行い、提出されている履修登録申請を図 2 の申請データとしてダウンロードし、教務データベースへ更新をかける。この時点で教務システムへの履修登録処理が行われ、教務システムとしてのエラーチェックも行われる。履修テーブルの内容をエクスポートして本システムに転送することにより“処理”が完了し、学生が s4s の Web サーバで履修登録の結果を確認できる状態となる。受理される前であれば、新しい申請を提出することによって、何度でも申請の差替えを行うことができる。また、受理前であれば、申請の取下げも可能である。

履修登録は一括して行う必要はなく、履修登録期間中であれば、確定した授業から順に複数回にわたって登録することができる。登録が終了したものから順に、担当教員がその学生を受講者リストで確認できるようになる。

3.8 未解決の履修登録エラーがある状態での提出の許可

既に 2.2 で述べたように、発生率の低い特別なケースについては、システムにすべてを仕組みとして組み込むことをせず、個別対応とすることが多い。すなわち、履修登録の“確認”の段階で判定困難なものについてはすべてエラーとしておきオンラインでの登録は拒否し、学生を窓口に来させて対応するという方法が従来一般的に行われてきた。これに対し本システムでは、未解決の履修登録エラーがあっても、学生が必要と認

表 2 履修登録における操作の種類
Table 2 Operations in registration.

操作	説明
作成	履修登録情報を準備すること。受取り手側(教務係)にはまだ伝わらない。
提出	受取り手側(教務係)が内容を見ることが出来る状態にすること。
受理	受取り手側(教務係)が内容を認め、処理すべき有効なデータとすること。
処理	受理されたデータにより、教務システムに対して履修登録の操作を行うこと。
廃棄	受理しなかった申請を無効とすること。

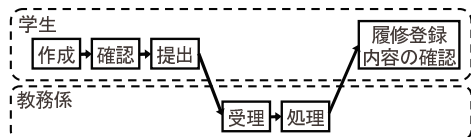


図 3 履修登録の流れ
Fig. 3 Flow of registration.

めた場合はそれを提出できる方式をとった。もちろん本システムにおいても、“確認”の段階で、教務データベースから転送した授業テーブル、履修課程テーブル、履修テーブル、過年度の履修テーブルなどの内容を参照し、一般的な履修登録チェックを実施している。その上で更に、エラーと判定されたものについて、学生が必要と判断した場合は提出も可能な仕組みとしている。エラー未解決のまま提出された登録申請へはオペレータが容易にアクセスできるインタフェースを設け、エラーの状況を把握しやすいようにした。“受理”においては、その内容によっては(1)エラーがあってもそのまま受理する、(2)その登録申請は受理せず、学生に適切な指導を行い、修正の上再提出させる、などの対応をとる。また、明らかに受理できない申請を無効とするため“廃棄”の状態を設けた。

図4にオペレータ(教務係)用の履修登録申請一覧のページを示す。学生の提出している履修登録申請について、このような一覧で状況を把握できる。すべての申請はひとりのエラーチェックが行われ、その結果(エラーの件数)も表示されている。そしてエラーがあるものについては、許可や廃棄のチェックボックスが現れ、操作できるようになっている。提出された個別の申請書を表示して詳細を調査し、状況に応じて必要な操作を行うことが可能となっている。

このように、作成、提出、受理、処理、廃棄の概念を導入することにより、基幹システムと別システムとした構成をより自然に表現でき、履修登録エラーがある状態でも履修登録申請を提出できる仕組みを分かりやすく高機能に実装することが可能となる。

4. 運用と評価

4.1 本システムによる履修登録の経緯

表3に示すように本システムによる履修登録は2001年度の後期から試行という形で開始された。本格的に実施されたのは2002年度からであるが、この年度は新入生の前期の登録については、情報処理教育をまだ行っていないことから様子を見るため、従来どおりの書類による方法を残した。2003年度は書類による履修登録を完全廃止し、本システムによる履修登録だけにより実施した。

4.2 Webサーバアクセス記録

表4はWebサーバ(s4s)へのWebのアクセスログをまとめたものである。2003年度の前期履修登録期間を中心とする4月8日から4月30日にかけてのアクセスについて調査したものである。これより学生がどこから利用したかなど利用の様子を知ることができる。この結果によると、学外(自宅や職場)から利用した学生は67%であり、社会人の多いBコース学生の場合は79%が学外から利用している。

4.3 履修登録申請提出件数の推移

2003年度前期履修登録期間(4/8~4/28)と後期履

表3 本システムによる履修登録の経緯
Table 3 History of registration using this system.

年度	実施状況
2001	前期は従来どおりの書類による履修登録、後期の登録変更申請から本システムの試作版にて試行実施。
2002	新入生(739人)の前期履修登録を除き、工学部全学生(3,343人)が本システムにてオンライン履修登録実施。新入生前期は従来どおりの方法。
2003	工学部全学生(3,253人)が本システムでオンライン履修登録完全実施。

学生証番号-通番	状態	作成	提出	受理/取下	処理	削除	追加	エラー	科目	単位	申請書	許可	廃棄
1 T03X003A-6	提出中	10/13 14:23	10/13 14:23					1 4 3	30 50.0	[表示]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2 T02X067B-3	提出中	10/13 16:12	10/13 16:12					0 5 0	27 51.0	[表示]	-	-	
3 T98Z005B-5	提出中	10/13 17:31	10/13 17:32					0 10 0	27 52.0	[表示]	-	-	
4 T01W073A-2	提出中	10/13 18:07	10/13 18:08					0 9 0	21 32.0	[表示]	-	-	
5 T01V011A-2	提出中	10/13 18:06	10/13 18:08					0 12 2	11 23.0	[表示]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6 T02Y046A-6	提出中	10/13 18:49	10/13 18:51					0 2 0	30 50.0	[表示]	-	-	
7 T02Z044B-8	提出中	10/13 19:53	10/13 19:53					0 2 0	29 46.0	[表示]	-	-	
8 T01Y065A-5	提出中	10/13 20:02	10/13 20:02					2 0 0	18 38.0	[表示]	-	-	
9 T01V091A-3	提出中	10/13 20:47	10/13 20:49					0 6 0	20 41.0	[表示]	-	-	
10 T01W049B-3	提出中	10/13 20:58	10/13 20:58					1 0 0	17 33.0	[表示]	-	-	
11 T01Y026A-5	提出中	10/13 21:00	10/13 21:01					3 0 0	22 42.0	[表示]	-	-	

ここに示した個人情報は架空のもので実在しない。

図4 履修登録申請一覧のページ

Fig. 4 List of registration requests.

修登録期間 (10/1 ~ 10/14) の履修登録申請提出件数の様子を図 5, 図 6 に示す. また, 締切り前後の詳細を図 7, 図 8 に示す.

前期の締切りのアナウンスは 4 月 28 日 18:00 とした. そのため, この日に向けて提出数が増加しているのが図 5 から読み取れる. 図 7 によると締切りが近くなった 4/28 の 14 時台と 16 時台に最大の提出数 (124 件) を記録している. また, 4/24 や 4/25 の時間別提出数を見てみると, 昼間のピークのほか, 夜間 23 時台にもピークがあるのが観測できる. これらは自宅からのアクセスによるピークである.

後期の締切りのアナウンスは 10 月 14 日 24:00 とし

た. 前期の履修登録は比較的分散傾向にあったが, 後期については締切りに集中しており, 日中午後 16:00 のピーク (109 件) のほかに夜間の 24:00 直前に最大のピーク (112 件) が観測される. 昼と夜の両方に大きなピークを生じた結果, ピーク値は前期のものより小さいが, 1 日の提出数としては前期の締切日よりも大きな値となっている.

4.4 アクセス集中時のロードアベレージ

図 9 に時間当りの提出数が最大を記録した, 前期 4 月 28 日における syll のロードアベレージ (5 分平均) の測定値を示す. 同時に図 7 の該当部分の提出件数の推移を重ねて示した. この場合, PostgreSQL 7.2.3 を

表 4 Web アクセスログから求めた利用場所
Table 4 Web server access log.

	全学生 (3,108 人)	A コース学生 (2,735 人)	B コース学生 (373 人)
学内だけから利用した	1,025 人 (33%)	948 人 (35%)	77 人 (21%)
学外だけから利用した	1,292 人 (42%)	1,106 人 (40%)	186 人 (50%)
両方から利用した	791 人 (25%)	681 人 (25%)	110 人 (29%)
学内から利用した	1,816 人 (58%)	1,629 人 (60%)	187 人 (50%)
学外から利用した	2,083 人 (67%)	1,787 人 (65%)	296 人 (79%)

(前期履修登録期間 2003 年 4 月 8 日 ~ 4 月 30 日)

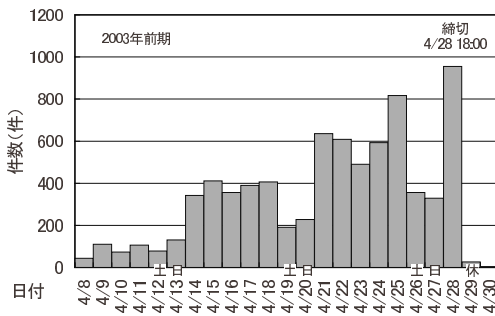


図 5 履修登録申請提出件数 (2003 年度前期)
Fig. 5 Request frequency in first half of 2003.

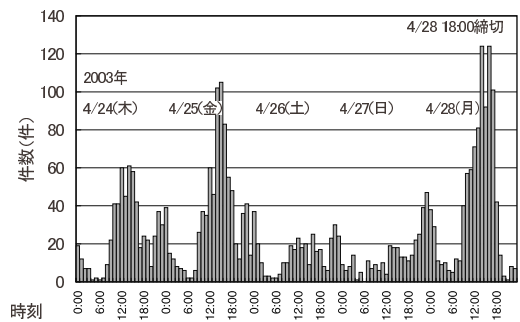


図 7 履修登録申請提出件数 (2003 年度前期)
Fig. 7 Request frequency in first half of 2003.

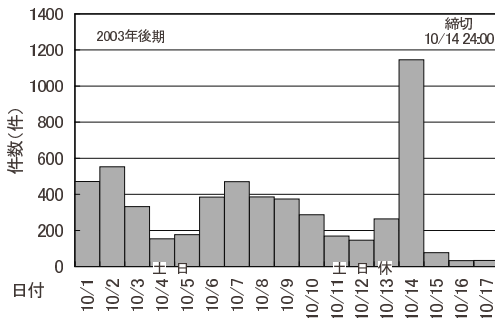


図 6 履修登録申請提出件数 (2003 年度後期)
Fig. 6 Request frequency in second half of 2003.

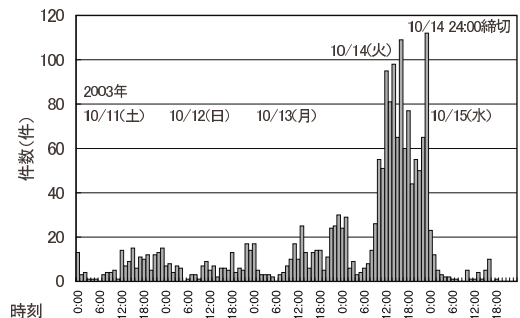


図 8 履修登録申請提出件数 (2003 年度後期)
Fig. 8 Request frequency in second half of 2003.

搭載した syll (Pentium III 800 MHz × 2, 1 MByte) がデータベースサーバとして動作し, その部分がボトルネックとなると考えられる. 時間当りの提出件数の最大を記録した 14 時台と 16 時台においても, ロードアベレージは 0.6 以下の十分低い値を示している. CPU 数が 2 であることを考えると, 一般にロードアベレージが 2 位までは十分に余裕のある処理が行えるとされている.

このときのおおよその処理内容を示すため, 図 10 に各ステップで実行される SQL 文の概要を示す. ステップの番号は 2.1 で述べた処理の流れの番号に対応している. from, join の後の括弧内にテーブルのおおよその行数を示す. この部分の select 文は副問合せを意味する. コード入力 (2), チェック (3), 提出 (4) の操作はそれぞれの繰り返しを伴いながら, 通常連続して実行される. 図 9 の時刻の範囲内において各人の操

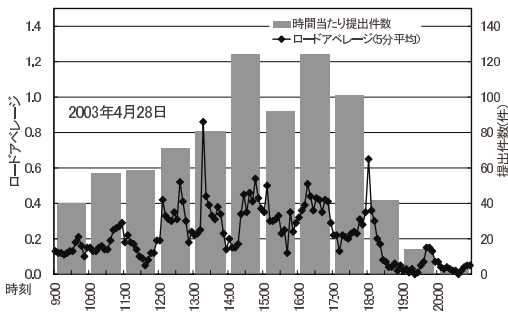


図 9 データベースサーバのロードアベレージ
Fig.9 Load average of database server.

(2) 作成 (コード入力)

小規模 select 文 (7 回)
delete 文 (削除項目ごとに 1 回)
select ~ from(少数) join(3 万) join(1 万) join(3 万); (1 回)

(3) 作成 (チェック)

小規模 select 文 (10 回)
select ~ from(少数) join(3 万) join(3 万) join(1 万)
left outer join(select ~ from(8 万))
left outer join(select ~ from(4 万) join(1 万)); (1 回)
select ~ from(少数) join(3 万) join(3 万) join(1 万)
left outer join(select ~ from(20 万))
left outer join(select ~ from(14 万))
left outer join(select ~ from(4 万) join(1 万)); (1 回)
select ~ from(select ~ from(8 万) join(3 万) join(3 万))
join(select ~ from(8 万) join(3 万))
left outer join(select ~ from(14 万)); (1 回)
insert 文 (追加項目ごとに 2 回 + エラー数ごとに 1 回)
delete 文 (削除項目ごとに 2 回)

(4) 提出

小規模 select 文 (8 回), update 文 (2 回)

図 10 履修登録時に発行される SQL 文

Fig. 10 SQL statement executed in registration.

作の様子について調べた(注1)ところ, 作成用 CGI 利用開始から平均 6 分で提出を行っている. また, 9 割は開始から 20 分以内に提出を行っている. ただし, コード入力でなく, 時間割や科目一覧から選ぶ方法をとった場合は, 時間割や科目一覧ページの表示操作とチェックボックスにチェックマークを入れる選択作業が作成用 CGI 利用開始前に必要となるが, その部分がシステム負荷に与える影響は小さいと考えられるのでここでは無視した. また, 同調査によると, 一度の提出のため (2) と (3) の操作を平均計 7 回行っている.

4.5 エラー未解決の履修登録申請の提出状況

表 5 に 2002 年度と 2003 年度における, エラー未解決の履修登録申請の提出状況を示す. 4.1 で述べたように, 2002 年度前期は新入生については原則紙ベースの履修登録としたため, 全提出人数は完全実施の 2003 年度の方が 8%ほど増加している. 全提出件数は 37%増とこれに対し大幅に増加している. 同様に受理件数も 32%増となっている. 1 人当りの平均の提出件数は 2002 年度が 3.6 回であったが, 2003 年度のそれは 4.6 回に増えている. 学生が, 複数回に分けて履修登録できるという本システムの仕組みを理解し, 履修を決めた授業から順次登録を行うようになった様子がうかがえる. エラー未解決の申請を提出した件数は 16%減で若干少ない程度であるが, それを受理した件数は 1 けた減少している. ほとんどの場合は, 提出後に学生自身が教務係や学科の教育委員の指導のもと, 各自でエラーを解決して再提出したことがうかがえる.

エラー未解決のまま提出された履修登録申請のエラーの内容は次のようなものであった.

(1) 担当教員が前年度の成績を保留していたため, 新年度の履修登録ができない状況にあった.

表 5 エラー未解決の履修登録申請の提出状況

Table 5 Number of registration request with errors.

	2002 年度	2003 年度
全提出人数	2,935 人	3,157 人
全提出件数	10,547 件	14,480 件
エラー提出人数	402 人 (14%)	336 人 (11%)
エラー提出件数	622 件 (6%)	541 件 (4%)
受理・処理人数	2,934 人	3,157 人
受理・処理件数	7,561 件	9,967 件
エラー受理人数	217 人 (7%)	22 人 (0.7%)
エラー受理件数	225 件 (3%)	22 件 (0.2%)
廃棄件数	7 件	14 件

(注1): 直接記録されていないので, Web のログをもとに, 10 件に 1 件をサンプル抽出し調査した.

(2) 年間履修単位の上限値を超えた登録を行おうとした。自分の単位が足りないため、無理に計算だけ合わせて登録しようとした例も見られたが、事情により、制限を越えざるを得ないケースも見られた。

(3) 他の授業との開講時限の重なり。実際は履修可能であるが、授業編成上の都合でエラーとなってしまったケースがあった。データの登録ミスが原因の場合もあった。

(4) 履修課程テーブル不備のため、正しい科目区分が設定されていなかった。

(5) 後期に、前期に単位がとれなかった授業を削除しようとする例が見られた。また、後期に前期の授業を登録しようと試みたケースも見られた。特別な理由があつた場合もあった。

(6) 後期に、通期の授業の追加を行おうとした。特別な事情のある学生で、この登録は認められた。特に(2)が多く見られた。これらについては、学科の担当の判断で、各個人の事情によっては単位制限を個別に引き上げることができることとし、システムの方もそれが可能なように仕様変更した。

4.6 評価

(1) 本システムは、独自開発し、そのスクリーニング機構を活用することで、数か月という短い準備期間で運用開始することができた。また、基幹システムである教務システムとは別システムとして構築したことも短期間で実現できた理由の一つである。無料でかつ信頼性の高いソフトウェアと安価になった高性能なハードウェアを活用してシステム構築を行った結果、定常的な予算の範囲内で、かつ現員のスタッフの業務の範囲内で開発・運用を行うことができた。本システムのように独自開発するためには、そのためのスタッフと体制が必要であり、どのような場合にでも独自開発が可能であるわけではない。また、継続して運用を行うためには、サポートのための体制の整備も重要となる。

(2) 本システムの基本部分は、2.1 に述べた流れに沿っており、表 1 に相当する情報にも一般性がある。それに対し、基幹システム(教務システム)との入出力部分は各大学における基幹システムのデータベースのデータ構造に直接依存する。また、一部コード設計の問題等から汎用性が失われている部分があり、これらの点について修正を行えば、本システムは他大学に対しても利用可能である。

(3) 本システムでは、履修登録コードによる登録

だけではなく、学生の授業計画用ツールとして利用できるよう、シラバスと完全連動させた時間割形式の一覧や、授業一覧からの登録も可能としたが、大多数の学生がコード入力ではなく、両一覧から登録を行っており、期待どおりの効果が得られた。

(4) できる限り学生が窓口に出向くことなく履修登録を完結できるよう設計を行った結果、従来の紙による履修登録時に比べ、締切り間際のピーク時における窓口の混雑の具合は、従来よりもおよそ2けたのレベルで改善され、窓口に行列ができることがなくなった。また、本システムの導入が原因となる規則の変更を生じることはなかった。

(5) 基幹システムと別システムとしたことから運用時間が自由に設定できるようになったが、深夜にも一つのアクセスのピークを生じるという運用結果から、24 時間サービスできることの重要性が改めて認識される結果となった。

(6) 2 年間の実運用において、セキュリティ上の問題は発生しなかった。いくつかのポートスキャンやセキュリティホールを突こうとしたと見られる攻撃が観測されたが、いずれもセキュリティ対策済みであり、問題は発生しなかった。

(7) 本システムの導入により、これまで紙の書類で処理されていたものが電子的に処理されるようになり、事務処理削減の効果が得られた。本システムの最終目標は質の高いサービスの提供であり、投入したコストは、同時に質の高いサービスの実現のために消費される。このシステムの導入により、学生にとっては窓口の長蛇の列に並んでいたのが改善され、24 時間自宅からシラバスを見ながら、履修する授業の計画を立てることが可能となった。教務係にとっては、煩雑な事務処理から開放された一方、学生が電子的に提出した履修登録申請の内容に目を通し、そのエラーの原因を種類別にまとめて解決するという作業を効率的に行えるようになった。学生はその結果、質の高いサポートを受けることが可能となった。

(8) 2003 年度前期におけるアクセス集中時のサーバの運用において、前年度の運用経験 [4] から、学年・学科別に締切りを変えるなどのピーク値を抑える措置はとらなかったがサーバの処理能力の範囲内に収めることができた。特に後期においては、締切りを 24:00 とアナウンスすることで、締切り日のピークを分散させる結果となった。

(9) 2003 年度にエラー未解決の受理が激減した

この理由として、本システムのスクリーニング機構が有効に機能し、このようにして把握できた問題点を順次解決して行った結果、プログラムやデータの質が向上し、学生への指導内容の改善も行われ、トータルとしての運用が安定したことが挙げられる。本システムのスクリーニング機構は、十分に満足できる効果が得られたと判断できるが、これを更に活用するためには、学生と教務係、学科の担当者とは、よりスムーズかつ確実に連絡がとれる体制を整えることが望まれる。本システムでは、s4s がもつ個人連絡用アドレスの機能 [8] を用いてメールを中心とした仕組みによって学生と連絡をとる設計となっているが、実際には学生と連絡がとれず、十分に機能しない場合が見られた。この点については、学生と担当係とのより密接で確実な連絡手段を実装するなど改善の余地がある。

5. む す び

ルールが複雑で個別対応するケースも多く、すべてを忠実に反映することが困難な履修登録システムにおいて、エラー未解決の履修登録申請の提出を許し、それらの状況を管理者が的確に把握できるスクリーニング機構を組み込むことでこの問題を解決した。(1) 本システムにより、教務係は窓口を介することなく、書類による履修登録のときと同様、学生の提出する履修登録申請に目を通すことができ、それまでとルールを変更することなく、各学生の事情を配慮して個別に指導するなどきめ細かな対応を行うことが可能となった。(2) 2002 年度、2003 年度における運用で、スクリーニング機構が有効に機能した。その結果、システム全体の質が向上し、学生への指導内容の改善が行われた。(3) 学外からの利用を許すことで半数以上の学生が学外からこれを利用した。そのアクセスの記録から、昼間の利用のピークのほかに、自宅からの利用が夜間にピークを作ることなどが確認できた。以上の点において、学生に質の高いサービスを提供することのできるシステムを構築することができた。スクリーニング機構により、きめ細かなサービスを更に効率的に実現するためには、より密接な学生と担当係との確実な連絡手段の実装が必要である。

文 献

- [1] 熊本大学教務課, “学務情報システム—SOSEKI,” <http://www.kumamoto-u.ac.jp/contents/stu/souseki.htm>
- [2] 北村 充, “学生情報システム「もみじ」の動向,” 広大フォーラム, no.377, Aug. 2003.
- [3] 檜垣泰彦, 板倉 勇, 池田宏明, “教務事務支援システムの開発,” 2002 信学総大, D-9-2, March 2002.
- [4] 檜垣泰彦, 阿由葉努, 土屋 俊, “履修登録システムの構築と運用,” 信学技報, OIS2003-10, May 2003.
- [5] 檜垣泰彦, 土屋 俊, 伊藤公一, 宮崎 清, “Web ベース授業支援システムの構築と運用,” (社) 日本工学教育協会平成 15 年度工学・工業教育研究講演会論文集, no.203, pp.609-612, Sept. 2003.
- [6] 檜垣泰彦, 藤田 隆, 池田宏明, “大規模国際会議における IT システムの設計と運用,” 2000 信学総大, D-9-6, March 2000.
- [7] 檜垣泰彦, 阿由葉努, 土屋 俊, “授業運用支援システムの構築と運用,” 信学技報, OIS2003-93, March 2004.
- [8] 檜垣泰彦, 阿由葉努, 土屋 俊, “電子メールによる学生連絡システム,” 信学技報, OIS2003-30, Sept. 2003.
- [9] The FreeBSD project, <http://www.freebsd.org/>
- [10] The apache software foundation, <http://www.apache.org/>
- [11] The PostgreSQL global development group, <http://www.postgresql.org/>
- [12] R.S. Engelschall, “mod_ssl: The apache interface to OpenSSL,” <http://www.modssl.org/>
(平成 16 年 5 月 20 日受付, 9 月 15 日再受付)



檜垣 泰彦 (正員)

昭 57 千葉大・工・電子卒。昭 59 同大学院工学研究科(修士課程)電子工学専攻了。同大・工助手。平 15 同大学院自然科学研究科(博士後期課程)入学。現在同大・工・都市環境システム学科助手。大学環境における情報システムに興味をもつ。



阿由葉 努

平元千葉大・工・工業化学卒。同年より同学部教務第二係を振出しに各学部の教務係。現在、同大工学部教務係。授業管理、授業情報提供、学生情報提供などに興味をもつ。



土屋 俊

千葉大学文学部行動科学科教授。心の哲学, 言語哲学, 論理学史, 状況意味論, 技術史・技術論, 情報倫理, マルチメディア文書処理, 対話構造及び対話理解・生成等に関する研究に従事。