

フレームワークを用いた学習・教育支援システムの構築について

大塚 貴史 白土 浩

近畿大学産業理工学部
sira@fuk.kindai.ac.jp

概要

本稿では、既存のデータベース連動型の教務システムで管理されている学生の成績情報等と連携した学習・教育支援システムの開発と改良を目指す。すなわち、学生の成績管理のみにとどまらず、各週毎の学生の出欠状況の確認、JABEEにおける学生向け総合達成度評価、学生面談記録（学生カルテ）等の機能を有する独自 Web アプリケーションとして開発する。そして、拡張性やセキュリティを考慮した、フレームワークプログラミングとして構築したシステムについて紹介する。

1 まえがき

大学における教務システム等のオンラインデータベース化は急速に進んでおり、学生の個人情報や履修・成績管理など様々なサービスを Web アプリケーションとして実装した市販ソフトも開発され、多くの大学で導入されている。しかし、これらのシステムの多くは事務作業の効率化を目的としているため、指導教員による学生指導という観点からみた場合、必要な情報が得られないという問題がある。例えば、担当科目単位での成績・出席状況については容易に取得可能であるが、学年・学生ごとの履修状況の取得は難しい。このため、ある学年における出席の傾向を知るために、曜日・時限毎の出席率を取得するといった作業が必要な場合、個別に導入ベンダと調整する必要がある。

学生指導において、学生の成績や個人情報、出欠情報、指導履歴等の情報管理は非常に重要である。本学科においても教務から得られる紙媒体の成績表や個人連絡カード、サイボウズや Word/Excel を活用した出欠管理表・学生指導履歴を用いた学生指導を実施している。しかし、各種情報が分散しているため、全教員が統一的なデータを参照することが難しい。したがって、これらの情報を一元的に閲覧・

管理できれば、効率的かつ充実した学生指導が可能になると考えられる。

本稿では、全学的に導入されている教務システムの情報と学科単位で保有している出席・学生指導履歴等の情報を統合し教員向けの学習・教育支援システムについて検討する。この学習・教育支援システムでは、学生の履修状況を把握するために教務システムに保存されている履修履歴と学科教員が毎週記録する出席簿情報を統合する。これにより、例えば、各学年毎の直近二週間における出席状況や学生毎の履修科目全体の出席情報、過去の学期における出席状況と単位修得状況等を一覧表示することが可能となる。

2 学習・教育支援

大学では多様な学生を受け入れているため、大学進学に合わせて独り暮らし等を始める学生も多く、中には生活リズムの崩れや体調不良などの要因によって遅刻・欠席が増える学生の早期対応は重要な課題である。このような学生をいち早く知るための手段として出席状況のチェックは重要であるといえる。

本学科では、4年前まで、出席情報を科目間で共有するためにエクセルを用い、シート毎に担当科目

の出席を記入し、マクロによって状況を把握する仕組みを利用していた。この方法では、休みがちな学生の検出は可能であるが、欠席が連続的なものか分散的かを見分けることはできない。さらに、学年毎、年度毎に別ファイルとして管理していたため、記録に連続性がないという問題もあった。そこで、出席を管理するための仕組みを PHP による Web アプリケーションとして実装し、DB(MySQL) 上に出席記録を保存するようなシステムを構築した。さらに、出席情報だけでなく教務システムから得られる成績情報を参照することで、過去の履修結果と出席状況の関連性が可視化できるようになり学生との面談において有効利用されている。

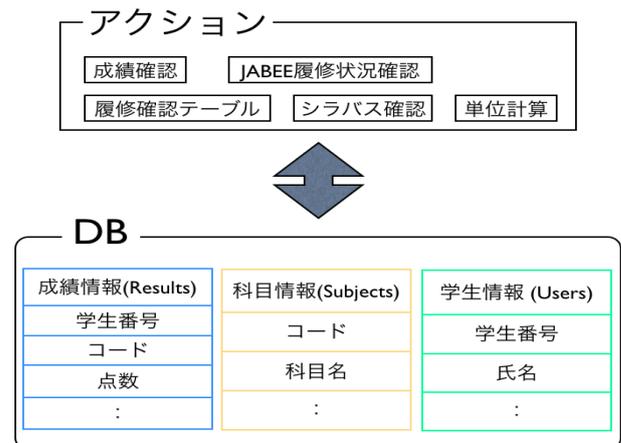


図1 機能構成図

3 履修支援システム

前節で述べた、学習・教育支援システムは教員による学生指導での利用を目的としているが、ここでは、学生自身による履修計画や学期毎の自己評価として利用する方法について考える。まず、既存の一般的な教務システムにおいても、履修登録機能は実装されており、学生が未履修の科目を時間割表に登録することで履修計画が可能である。しかし、履修コースが決定していない段階でのコース必修・選択等を考慮した履修計画や JABEE における学習・教育目標の到達に必要な科目の履修といった詳細な計画は困難である。さらに、シラバスの参照やコース毎の必修選択の別、取得予定単位数といった情報が表示されないシステムも存在する。この場合、学生は履修登録に必要な情報を自分自身で収集する必要がある。

そこで、学習・教育支援システムに科目毎のシラバス情報やコース毎の単位数、JABEE における学習教育目標に関する情報を登録し、学生自身による履修計画が円滑に進むようなシステムを開発した。

4 学習・教育支援システムの構築

ここでは、本学科で開発した学習・教育支援システムの仕様、実際の利用について述べる。

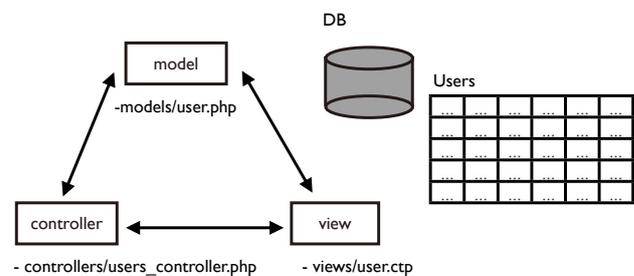


図2 MVC モデル概要

4.1 システム仕様

システムが提供する機能としては、1. 成績確認、2. 科目別出席入力、3. 出席確認、4. シラバス登録・閲覧、5. 学生面談、6. JABEE 履修状況確認からなっており、全ての機能へのアクセス時に LDAP によるユーザ認証を必要とし、アクセスレベルに応じた表示範囲の設定を可能とする。開発に用いる言語は PHP、MVC フレームワークとして CakePHP、データベースには MySQL を採用する。

4.2 フレームワークと MVC モデル

CakePHP では ソフトウェア設計モデルとして、MVC モデルを採用しており、プログラム全体を Model, View, Controller の 3 要素に分けて開発する。すなわち、まず、Model では DB 上のテーブルおよびテーブル間連携について記述し、Controller

や View からの呼び出されるメソッドを規定する。次に、View ではユーザからの入力イベントを Controller に伝え、Model 内の関連するデータを取得し、画面出力を更新する。そして、Controller では View からの入力イベント処理をイベントハンドラ等により実装し、ユーザからのアクションに応じた Model メソッド処理を担当する。

CakePHP は命名規約やディレクトリ構成などの規則が統一されており、例えば、図 2 のように DB 内にあるテーブル名を Users とすると Model は user.php , Controller は users_controller.php , View は user.ctp と一意に決定される。この命名規則に遵守すれば、セキュリティや保守性の向上やコードの再利用性も高まるため、開発効率的は高くなると考えられる。

以上のように、CakePHP による Web アプリケーション開発では、MVC モデルの採用や命名規則、ディレクトリ構成の統一によって、コードが一定の品質を保つことが可能となって、開発効率だけでなく保守・メンテナンス性についても向上すると考えられる。

4.3 データベース設計とアソシエーション

CakePHP では、主キー・外部キーにより DB 内のテーブル同士を関連づけるアソシエーション機能を持っている。この機能にも命名規約が定められており DB の要素名が規約に従っていれば、Model 内で自動的に定義される。すなわち cakephp コマンドにより外部キーと一意に対応する主キーを持つ Model は hasOne , 1 対多の関係に有る Model は hasMany , 外部キーを持つ Model は belongsTo なる定義が記述され [1], DB 内のテーブル同士を関連づける機能を持っている。

図 3 のように、DB 内の Users と Subjects テーブル内の id を主キー、Results テーブル内の学生番号 (user_id) , 科目コード (subject_id) を外部キーと定義した場合についてのアソシエーションを考える。まず、Results テーブルに対応する Model 内に belongsToUsers, belongsToSubjects が定義される。そして、主キーである Users テーブルの学生番号 (id) および Subjects テーブルの科目コード

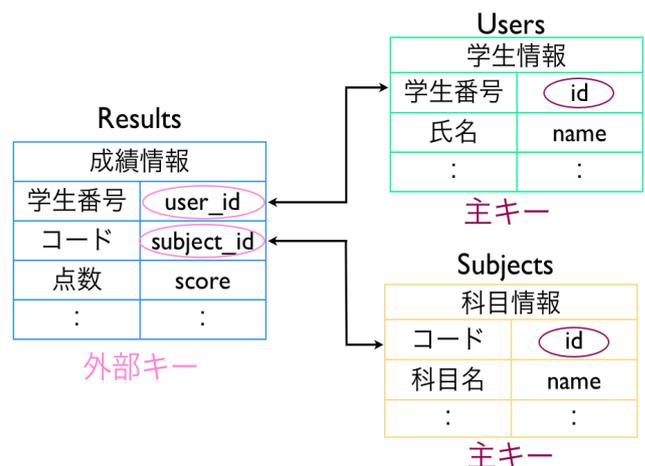


図 3 データベース間のアソシエーション規定

(id) に対応する外部キー (user_id, subject_id) を持つ Results モデルに対して、hasMany 定義が付与される。そして、Results モデルへの問い合わせの際、結果レコードに外部キーと対応する Users , Subjects の各フィールドを再帰的に取得することになる。

このように、アソシエーションでは、belongsTo 関係にあるテーブルの要素を自動的に再帰探索するため、通常の DB 利用で必要となる結合定義が不要となる。このため、予め、主キーと外部キーを指定しておくだけで、開発者は必要な情報を容易に取得可能となる。

4.4 ユーザ認証

本システムでは学生・教員等の成績や出席状況などの個人情報を取り扱うため、不正なリクエストに対する対応が重要となる。まず、ユーザ認証についてはシステム側でパスワード情報を DB 上に保持する場合、その取り扱いや他システムとの連携が必要となるため煩雑となる。次に、通常のようにログインページを経由せず、直接 URL がリクエストされることを防ぐために、ページ毎に認証チェックが必要となり、必要なセキュリティレベルを保てなくなる可能性がある。前者については、CakePHP の標準機能として、AuthComponent と呼ばれる認証モジュールが用意されているが、DB 上にパスワード情報を用意することが前提であるため上述の問題

を解決できない。

そこで、新たに Unix 等で広く用いられている LDAP (Lightweight Directory Access Protocol) 認証に対応した Component として、LdapAuthComponent を作成した。また、後者について、CakePHP で提供されるフィルタチェーンと呼ばれる全てのアクションの前後に共通処理を追加する機能を利用する。すなわち、フィルタチェーン beforeFilter() 内で LdapAuthComponent を利用したユーザ認証処理を構成することで、既存の全ページだけでなく将来作成されるページに対してもユーザ認証処理を付与することが可能となる。

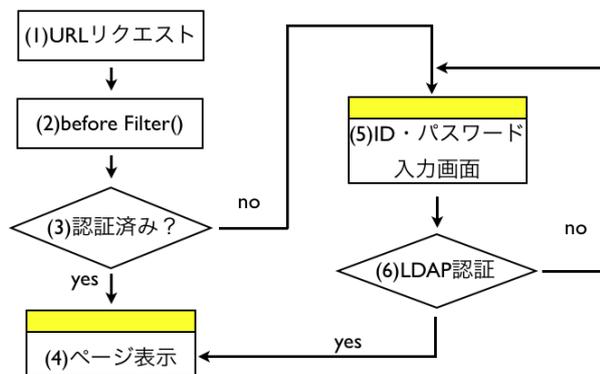


図4 ユーザ認証機能図

新たに作成した LdapAuthComponent を beforeFilter() 内で適用する流れを図4に示す。(1) ユーザから送信された URL リクエストに対し、(2) フィルタチェーンである beforeFilter() により認証処理が呼び出される。(3) ユーザが認証済みであるかを評価し、(4) 認証済みであればリクエストページの閲覧が可能となり、(5) 未認証であれば、ログイン ID とパスワードの入力画面を表示し、(6) 入力された情報が認証されない場合、再度入力を表示する。

5 まとめ

本稿では、既存の Web 履修支援システムをフレームワークを採用している CakePHP を用いて再構築する主案について提案した。また、プログラ

ム作成内容の主な詳細については当日発表する。

参考文献

- [1] CakePHP: the rapid development php framework URL:<http://cakephp.org/>
- [2] 岡田佳典, 「CakePHP(ポケットリファレンス)」, 株式会社フルーオーシャン (2009)