2Z-4

Symfonyを用いたシステム開発：設計と実装

鈴木　美穂† 石井　涼† 小野寺博之† 大谷　真†
湘南工科大学†

1. はじめに

PHP の最新フレームワークである Symfony を使って学内のゼミ室予約システムを開発した。開発を通して Symfony の特徴的機能がどの程度役に立つかを評価した。

2. Symfony

2.1 概要

Symfony とはフランスの Sensio Labs 社が開発した php5 のフレームワークである。2007 年 1 月にバージョン 1.0 が公開し、現在は安定版の 1.4 のほか、2.0(Symfony2)が開発されている。

2.2 特徴

Symfony の特徴には以下のものが挙げられる。本研究ではこれらの特徴機能がシステム実装にどのように適用できるのかに焦点を当てた。

(1) MVC 構造
Symfony は MVC 構造に基づいている。

(2) yaml による設定
yaml によりデータベースの設定などができる。

(3) ORM (ObjectRelationalMapper)
Doctrine と呼ばれる ORM 機能がある。

(4) sfForm 機能
フォーム処理を簡略化できる。

(5) admin ジェネレータ
CRUD（レコードの作成、検索、更新、削除）を行う画面と処理を自動生成できる。管理者向けの簡単な画面の開発などに向いている。

3. 機能設計とデータベース設計

3.1 ゼミ室予約システムの機能

湘南工科大学情報工学科には 8 つ程のゼミ室がある。各部屋はゼミでの利用が主体だが、少人数の授業でも使われることもある。学期の開始時に学科の部屋管理者が授業に割り付け、その後先生方が必要に応じてゼミ室を予約する。この作業を支援するのが今回開発したゼミ室予約システムである。表 1 に示すように一般者（実際には先生）と管理者（部屋管理者）ごとにゼミや授業の登録、部屋属性の設定などができる。

予約は、
・一日の予約で一つの教室・コマを使う
・一日の予約で複数の教室・コマを使う
・複数の日にちに一つの教室・コマを使う
・複数の日にちに複数の教室・コマを使う
ができるようにした。

A System Development using Symfony:
Design and Implementation.
†Miho Suzuki, †Ryo Ishii, †Hiroyuki Onodera, †Makoto Oya,
†Shonan Institute of Technology

表 1　機能一覧

	機　　能	管理者	ユーザ
一般	ゼ　ミの登録・修正・削除・訂正	○	○
管理	授　業の登録・修正・削除・訂正	○	
	部　屋の登録・修正・削除・訂正	○	
	ユーザの登録・修正・削除・訂正	○	

3.2 データベース設計

図 1 に示すように Subject(科目),Reserved(予約),Room(部屋)の 3 つのテーブルを作成した。

・Subject(科目)テーブル：
　・kind：授業かゼミかの区分
　・title：授業名またはゼミ名
　・professor：担当教員名
　・created_at と update_at：作成時間と更新時間
・Reserved(予約)テーブル：
　・date と day_of_week：日付と曜日、
　・koma：何時間目なのかを示す
　・subject_id と room_id：その予約の科目と部屋
・Room(部屋)テーブル：
　・room_name：部屋名
　・large、pc など：部屋の属性（大きさや PC 数）

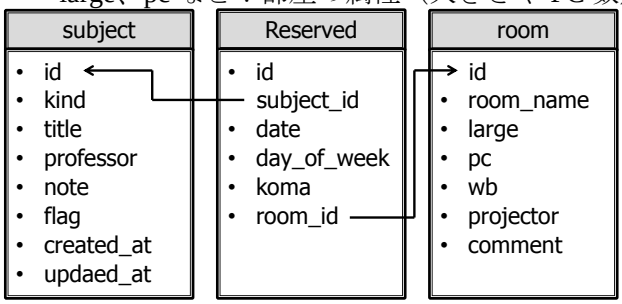

図 1　データベース

4. 実装

4.1 予約操作と MVC 構造

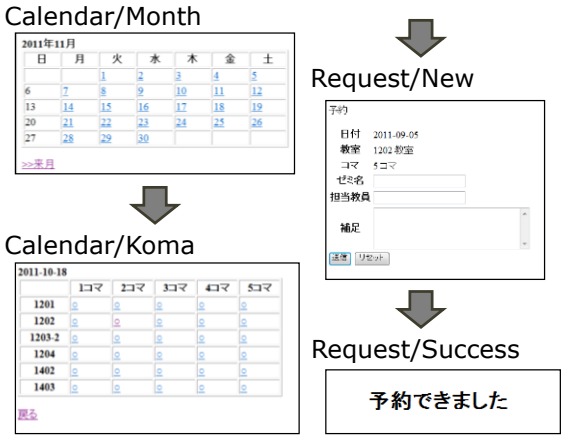

図 2　画面遷移

予約操作はユーザにとってできるだけ簡単であるよう配慮した。図2は画面遷移の例である。カレンダー(Calendar/Month)から日付を選択し、教室とコマ(Calendar/Koma)で希望する教室とコマを選ぶ。

以上のような予約操作をするために Symfony の MVC 規則にのっとり図3のようモジュールを設計した。

Model
subject
reserved
room

	Controller	View
モジュール	アクション	テンプレート
Page	Top	Top
	Success	Success
	Failure	Failure
Calendar	Month	Month
	Koma	Koma
Request	New	New
	Save	-

図3　MVC に基づくモジュール設計

4.2　yaml による DB 設定

図 4 は図 1 の Reserved テーブルを生成した schema.yml である。各カラムの他に Room、Subject 両テーブルへのリレーションも設定している。またデータベース名、パスワードなどは databases.yml に記載した。

```
Reserved:
  columns:
    subject_id:   integer
    date:         date
    day_of_week:  integer
    koma:         integer
    room_id:      integer

  relations:
    Room:
    local:        reserved_id
    type:         one
    foreign:      room_id
    foreignType:  many
    foreignAlias: room_reserveds
    onDelete:     CASCADE

    Subject:
    local:        reserved_id
    type:         one
    foreign:      subject_id
    foreignType:  many
    foreignAlias: subject_reserveds
    onDelete:     CASCADE
```

図4　yaml による DB 設定の例

4.3　Doctrine の利用

本プログラムでのデータベースへのアクセスは Doctrine を全面的に利用した。例えば、図5は Room テーブルから教室名 (noom_name) を Doctrine を使い取得するプログラムである。

```
①　$r = RoomTable::getInstance();
②　$room = $r->findOneById(1);
③　$name = $room->getRoom_name();
```

図5　Doctrine の利用例

①は Room テーブルのインスタンスを作成し、②で該当するレコードを検索し抽出している。③で Doctrine が自動生成したメソッド get_name() を使って、レコードの教室名(noom_name)を取得している。

しかし、一つ問題が発生した。予約をする際 Reserved には対応する Subject_id を渡すのだが、Subject の id は auto increment にしていた。php、MySQL には直前の auto increment で生成された値を取得する関数が用意されているが、Doctrine を使っているのでその関数の使用が出来なかった。このように Doctrine を使うことで普段なら実現可能な機能が一部制限されてしまう。

4.4　sfForm の活用

予約ページ(Request/new)でのフォームは sfForm を利用した。Form クラスに Type1 というフォームを作りウィジェット（Input、Textarea 等）とバリデータ（required、max_length 等）を定義した。その後、図3にある Request モジュールの New アクションから呼び出し、テンプレートで表示した。このフォームはどのモジュール、アクションでも何度でも使用可能である。

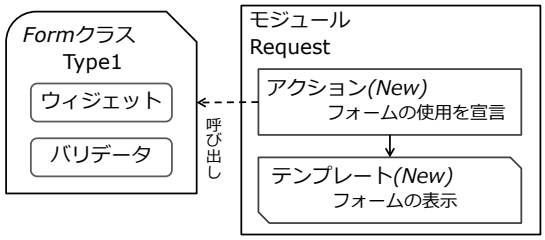

図6　sfForm の流れ

4.5　admin ジェネレータの利用

Room モデルに対する部屋管理ページの作成に、admin ジェネレータを利用した。部屋の一覧画面、新規作成、検索、削除などの機能がコマンドを一行打つだけで実装できた。ただ予約の管理には subject テーブルだけでなく、reserved テーブルの情報も必要だった。admin ジェネレータは二つ以上のテーブルを同時に使うには不便であるため、ここでは使わなかった。

5.　まとめ

本研究ではゼミ室予約システムを開発し Symfony の各特長がシステム開発にどの程度効果があるか評価した。その結果

- ✓ MVC 構造をとることで、機能ごとにファイルが分けられ、一つ一つのプログラムがコンパクトになり可読性が上がった
- ✓ Doctrine が自動生成したメソッドを使うことで、難解な SQL 文を書く必要がない
- ✓ Form を分離することで、同じようなフォームを再度開発する必要がない
- ✓ CRUD 画面の生成だけでなく、表示などのカスタマイズも generator.yml の編集のみでできき、作業が簡略化した

6.　参考文献

[1] 日本 Symfony ユーザ会、Symfony1.4 による web アプリケーション開発、秀和システム、2011