

MVCアーキテクチャを採用したWAFを使う開発を補助する Emacs 拡張の開発

1. 背景

近年、増加傾向にある Model View Controller (以降 MVC) アーキテクチャを採用した Web アプリケーションフレームワーク (WAF) を使った開発において、コードを書くという生産的な作業以外に費やされている時間は無視できないものである。

たとえば、コントローラから呼び出されるモデルへ、モデルから対応するコントローラへ、コントローラから最終的に描画されるテンプレートへ、といったファイル間の移動などの何度も繰り返される動作である。また、それらの動作は各種フレームワークに共通する動作でもある。

2. 目的

本プロジェクトでは、この共通点を抽象化することによって、多くの Web アプリケーションフレームワークを使ったプロジェクトの開発補助を実現する Emacs 拡張を開発する。

3. 開発の内容

本プロジェクトでは以下の開発を行う。

- ユーザにストレスを感じさせないインタフェース
- エディタ拡張の専門知識の無いプログラマが簡単に機能拡張することのできるエディタ拡張システム
- エディタ固有のユーザインタフェースと開発補助のための本質的な処理とを適切に分離した再利用性のあるエディタ拡張システム

ユーザインタフェース以外の処理を行うプログラムを、Emacs に依存しない単体のアプリケーションとして開発する。このアプリケーションはユーザーのローカルでウェブアプリケーションとして動作し、Emacs とは RESTful な API を通して HTTP プロトコルで通信する。

開発したシステムは3つのパートに分けられる。それぞれ再利用性を高める設計となっている。

- コア処理部

プロジェクトのファイルシステムをスキャンし、各ファイルごとの依存関係をスコアつきで計算する。これは、ファイルが更新するたびに実行される。各ファイルから依存関係を抽出する部分はプラグインとして実装され、拡張が容易になっている。現在、以下のフレームワークに対応済みである。

Perl: ARK, Amon2, Catalyst, Kamui

PHP: Symfony, CakePHP, Zend Framework

この部分は今後新規にフレームワークがリリースされた場合でも即座に対応することができる。依存関係の情報はデータベースに格納され、高速な

検索に耐えられるようにしている。またこの依存関係データベースはスコアを常時更新することで、精度を高めるようになっている。スコア更新のためのAPIも用意している。

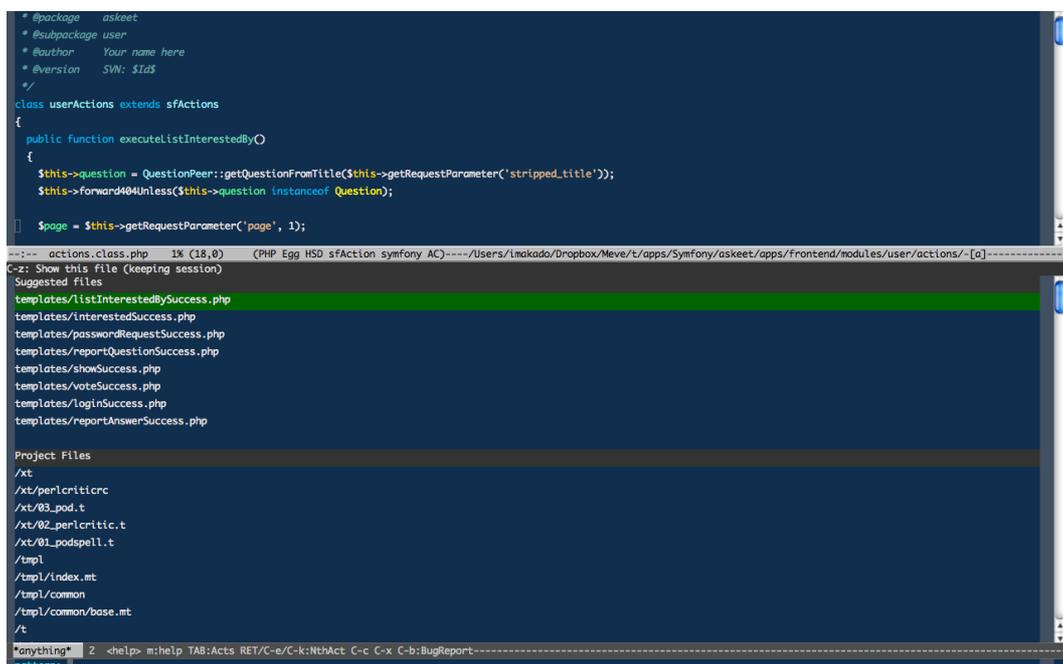
- 通信部

Emacs からのリクエストに応じて、上記処理の結果を返答する。また、Emacs のユーザインタフェースからのフィードバックをコア処理部に送信する。これはローカルに起動している Web サーバを利用して動作している。Web サーバを利用した理由としては、Web 開発においてローカルにテスト用の Web サーバを起動するケースが多く、インストールの手間を最少にできるためである。

- ユーザインタフェース部

Emacs 上のプログラムとして、各 Web アプリケーションフレームワークの編集環境に機能を追加する形で実装している。動作としては、ポインタが指し示している名前の上でファイル遷移キーを押すと、バックグラウンドでコア部と通信し候補のファイル一覧を獲得し、別バッファ上でそれらのファイル一覧を提示する。ユーザはそのバッファ上で遷移したいファイルを選択し、それを開く。同時に、選択されたファイルをフィードバックとしてコア部と通信する。

開発したシステムのスナップショットを以下に示す。



```
* @package askeet
* @subpackage user
* @author Your name here
* @version SVN: $Id$
*/
class userActions extends sfActions
{
    public function executeListInterestedByO
    {
        $this->question = QuestionPeer::getQuestionFromTitle($this->getRequestParameter('stripped_title'));
        $this->forward404Unless($this->question instanceof Question);

        $page = $this->getRequestParameter('page', 1);
    }
}
----- actions.class.php 1K (18,0) (PHP Egg HSD sfAction symfony AC)-----/Users/Imakado/Dropbox/Meve/t/apps/Symfony/askeet/apps/frontend/modules/user/actions/-[a]-----
C-z: Show this file (keeping session)
Suggested Files
templates/ListInterestedBySuccess.php
templates/InterestedSuccess.php
templates/passwordRequestSuccess.php
templates/reportQuestionSuccess.php
templates/showSuccess.php
templates/voteSuccess.php
templates/loginSuccess.php
templates/reportAnswerSuccess.php

Project Files
/xt
/xt/perlcriticrc
/xt/03_pod.t
/xt/02_perlcritic.t
/xt/01_podspell.t
/tmp1
/tmp1/index.mt
/tmp1/common
/tmp1/common/base.mt
/t
*anything* 2 <help> m:help TAB:Acts RET/C-e/C-k:NthAct C-c C-x C-b:BugReport
pattern:
```

図 1: コマンドキーによる遷移候補ファイルの選択

図 1 はリアルタイムに遷移候補ファイルを表示するのではなく、コマンド (key の押下) によって候補を選択する動作の様子を示している。

```

templates/listInterestedBySuccess.php 11
templates/InterestedSuccess.php 10
templates/passwordRequestSuccess.php 10
templates/reportQuestionSuccess.php 10
templates/showSuccess.php 10
templates/voteSuccess.php 10
templates/loginSuccess.php 10
templates/reportAnswerSuccess.php 10

-:~%* *Albatross:Files* All (1,0) (Files E2)
Buffer History [4]
1 actions.class.php
2 *home-keys:*
3 .zsh_history
4 index.html *

-:~: actions.class.php Top (20,0) (PHP E2w(albatross) Egg H50 sFAction symfony AC)----/Users/mokado/Dropbox/Meve/t/apps/

```

図 2: リアルタイム遷移候補ファイル提示

```

templates/InterestedSuccess.php 12
templates/listInterestedBySuccess.php 11
templates/passwordRequestSuccess.php 10
templates/reportQuestionSuccess.php 10
templates/showSuccess.php 10
templates/voteSuccess.php 10
templates/loginSuccess.php 10
templates/reportAnswerSuccess.php 10

-:~%* *Albatross:Files* All (1,0) (Files E2)
Buffer History [4]
1 actions.class.php
2 *home-keys:*
3 .zsh_history
4 index.html *

-:~: actions.class.php Top (23,37) (PHP E2w(albatross) Egg H50 sFAction symfony AC)----/Users/mokado/Dropbox/Meve/t/apps/

```

図 3: リアルタイム遷移候補ファイル提示 (カーソル移動後)

図 2 は左上のウィンドウにリアルタイムに遷移候補ファイルを表示するモードで動作させた様子を示している。図 3 では、カーソルをメソッド `executeInterested()` に移動させた後に、左上の遷移候補ファイルの表示が適切なものに更新されている様子を示している。

4. 従来の技術（または機能）との相違

開発したプラグインの高い拡張性、柔軟性は、今後対応する WAF フレームワークを容易に追加可能なものとなっている。加えて開発したコア機能は、Emacs 以外のエディタに対しても、エディタごとの通信部分を追加で実装するだけで利用可能なものとなっている。

5. 期待される効果

本プロジェクトの成果により、対応するフレームワークの多さから、非常に多くの恩恵を受けるユーザー生まれる。

6. 普及（または活用）の見通し

本プロジェクトの成果はオープンソースソフトウェアとして公開する予定である。配布方法については以下を想定している。

- Emacs 拡張

コードホスティングサイト「Github」にて配布する。

- Perl で開発されたアプリケーション部分

アプリケーションの配布は Perl のパッケージングシステムである CPAN を利用する。CPAN を利用することによって、最小限の作業でインストール・アップデートが行える。また、アプリケーションが利用するライブラリの依存関係を解消してくれる。そのため、頻繁にバージョンアップが行われると予想されるリリース当初の時期でも、ユーザーに負荷をかけることなく最新のものに更新することが可能となる。

そのほかに「CPAN Testers」というサービスを利用できるという利点もある。これは CPAN にアップロードしたアプリケーションのテストを様々な環境でテストし、その結果を通知してくれるサービスである。つまり、アプリケーションのテストを完璧に書いておけば、一人では難しい多様な環境での動作確認を行えるということである。

7. クリエータ名（所属）

今門 研爾（フリー）