

コード生成を用いたフレームワーク向け Web アプリケーション開発支援ツールの作成

京谷 和明 伊藤 恵

近年, Web アプリケーション開発では, より一層の短納期化, 高品質化が求められている. そのため, Web アプリケーション開発には様々なフレームワークが用いられていることが多い. しかし, フレームワークを用いる場合, 類似したコードを記述することも多く, 開発作業が非効率となる. また, 効率化のため自動生成ツールを用いる場面も増えてきている. しかし, 自動生成ツールを用いる場合は, 生成元のモデルについて深く理解し, 細部まで記述しなければならないことが多く, その自動生成ツールを扱うための学習期間が長くなってしまふことがある. そこで本研究では, Web アプリケーションの作成を行う際に, 一般的に作成され, 理解のしやすいモデルである画面遷移図と ER 図より, CakePHP のアーキテクチャに則ったコードを半自動で生成することで, 機敏に開発を行うことができ, フレームワークによる質の良いコードを生成できるツールの作成を行う.

Recent years, web application development is more needed the short delivery time and high quality. Therefore, various frameworks are used in web application development. But if developers use framework, they often write similar code, so development working is inefficient. Also, web application development use various auto code generation. But for using code generation tools, developers must deeply understand about the model for the tools, and they must write details in the model, and learning period about the model is often lengthy. So this study makes a semi-automatic code generating tool that the conforming CakePHP architecture. The input models of this tool are ER model and screen transition that diagrams are typical created in web application development. By this tool, developers rapidly develop web application and the generated codes are high quality based on the framework.

1 はじめに

Web アプリケーションの開発では様々なフレームワークが用いられることが多い. フレームワークとはソフトウェアを開発する際の枠組みとなるものであり, 頻繁に作成される機能やコードをまとめて提供することで, システム開発の支援をしている. Web アプリケーションのフレームワークとして, 例えば Ruby のフレームワークである Ruby on Rails, PHP のフレームワークである CakePHP などが挙げられる. Web アプリケーションの開発にフレームワークを用いるメリットとしては一般的に開発工数の削減と品質の均質化などが挙げられる. デメリットとして

は, フレームワークごとに定義や作法が異なり, そのフレームワークを扱うのに学習期間が長くなってしまい, 学習したとしても他のフレームワークでは扱うことができないこともあるといったことが挙げられる.

また, Web アプリケーションの開発には自動生成ツールを用いる場面が増えてきている. 自動生成の例としては, 各クラスの名前や属性や関数などを図で表すクラス図やオブジェクト間の処理の流れを図で表すシーケンス図といった, 統一モデリング言語 (UML) から自動生成を行う場合がある. 或いは独自規格を作成し, それに則って画面レイアウトや機能を決定し, そこからソースコードを自動で生成する製品も存在する. 自動生成ツールを導入するメリットとしては, 開発工数の削減と品質の均質化, 設計したドキュメントや図を最大限に活用できるといったことが挙げられる. 同様に, デメリットとしてはモデルから自動生成を行う場合は設計の段階で非常に細かいところまで

機能の洗い出しをして設計しなければならない、自動生成ツールを扱う場合は学習期間が長くなってしまふことが多いことである。

本研究ではこれらのフレームワークと自動生成ツールに関する問題点を解決するために、学習期間があまり長くならず、設計も簡単に行うことができ、さらにコメントも充実させた雛形を生成することで機敏に開発を行うことのできるツールの作成を行うことを目的とする。

2 関連研究

自動生成ツールに関する先行研究として、河村、浅見らの「UML を入力とするソースコード自動生成ツールの開発」[1]が挙げられる。この研究では Web アプリケーション開発における自動生成技術の適用に対して、クラス図とシーケンス図からフレームワーク用のコードを自動生成するツールの作成に取り組んでいる。この研究で作成されたツールを適用することで、9 割以上のソースコードの生成に成功している。しかし、クラス図とシーケンス図はある程度のシステム開発の経験が無いとどう作成すればよいのかわからないことがある。

また、他の先行研究として、八木、中所らの「モデル変換に基づくエンドユーザ主導の Web アプリケーション開発技法」[2]がある。この研究ではエンドユーザが主導して、Web アプリケーションを開発するために 3 つの自動生成ツールを作成している。その 3 つのツールを適用することでシステム開発経験の浅いエンドユーザでも直感的に開発が行えるようにしている。この研究ではこのツールを使って汎用プログラミング言語を用いるよりも、はるかに容易に Web アプリケーションを作成することに成功している。

本研究の関連製品としては、キャノンソフトウェア株式会社が作成した、「Web Performer」[3]が存在する。この製品は GUI 操作で要件定義情報から業務用の Web アプリケーションを 100 %自動生成できるツールであるとのことである。しかし、こちらは 1ヶ月弱の学習期間が必要であり、また製品の値段が 300 万円以上するとのことなので、小さなシステム開発会社にとっては敷居が高くなってしまふ。

その他の関連するツールとしては、株式会社ジャスマインソフトが販売している「Wagby」[4]、オープンソースソフトウェアである「blanco Framework」[5]等が挙げられる。

いずれのツールも自動生成率は高く、Web アプリケーションの開発に適用することで機敏にシステム開発を進めることができると思われる。しかし、設計段階で必要となるモデルを細かく作成しなければならない、学習期間も長くなってしまふツールが多く、これらのツールの導入に手間取ってしまふかもしれない。

3 アプローチ

ここで本研究では、既存のツールと比較して、生成できるソースコードの量は少ないが、必要とするモデルを簡単にすることで、導入をしやすいツールの作成を目指す。本研究で扱うモデルは画面遷移図と ER 図である。これらのモデルは UML を中心としたモデリングを行うことのできるツール astah* を使って作成する。そして作成したファイルを解析し、CakePHP のコードを生成する。

3.1 ツール構成図

ツールは Java を使って作成する。作成した画面遷移図と ER 図のファイルから文字列を抽出し、抽出した文字列からディレクトリやファイルを生成し、そのファイルにコードを記述する。図 1 に本ツールの構成図を示す。

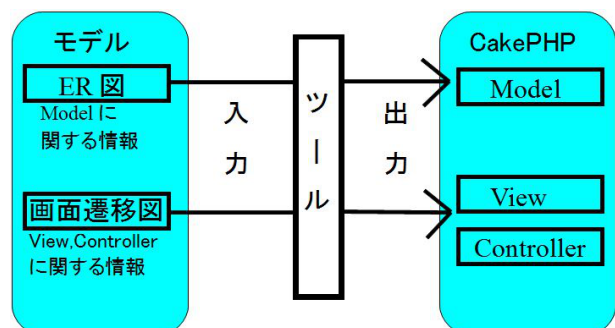


図 1 ツールの構成図

3.2 使用するモデル

本研究ではモデルに画面遷移図と ER 図を扱う。

画面遷移図とは画面の設計を行う際に作成されるものであり、各画面の機能や画面の名前、ボタン押したときや処理を行った際にどの画面に遷移するかが記述されている。

そして ER 図は、和名では実体関連図と呼ばれている。システムのデータを図に表すことで、データベースを抽象的に示したものである。本研究で扱うデータベースは第 3 正規形であり、ER 図は IDEF1X 表記のものを取り扱っていく。また、ER 図は SQL に記述するテーブル名、カラム名と同じ英語表記とする。

画面遷移図と ER 図はいずれも Web アプリケーションを開発する際に作成されることが多く、理解もしやすい。

また、これらのモデリングツールとして、UML を中心としたモデリングを行うことのできる astah* を扱う。画面遷移図は astah* のステートマシン図作成ツールを使い、ER 図は astah* の ER 図作成ツールを用いる。本研究ではこの astah* で作成した画面遷移図と ER 図のファイルを解析することでコードの自動生成を試みる。

3.3 フレームワーク

本研究では Web アプリケーション開発フレームワークである、CakePHP 用のソースコードを生成する。ソースコードは PHP や HTML, JavaScript 等の言語である。CakePHP は MVC アーキテクチャに準拠しており、また多くのリレーショナルデータベース管理システムと連結できる。MVC とは Model-View-Controller の頭文字をとったものであり、Controller にシステムの機能の処理を記述し、View に画面への出力法を記述し、Model にデータの処理を記述する。MVC アーキテクチャを利用することで、それぞれの処理を独立させることができ、コードを均質化することができる。これらのことから本研究では CakePHP のアーキテクチャに則ったコード自動生成を試みる。

3.4 生成手順

本研究では上記で述べたとおり、画面遷移図と ER 図から CakePHP のソースコードを半自動で生成する。図 2 は ER 図の一部である。

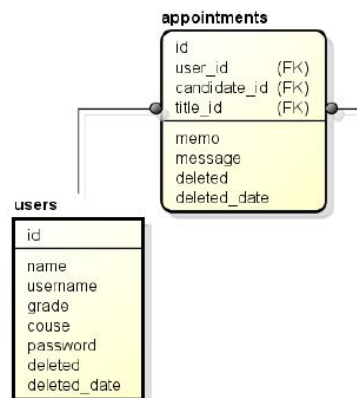


図 2 ER 図の一部

図 2 では各テーブルの名前とカラム名が記述されており、users テーブルと appointments テーブルが 1 対 0 以上の関係で表されている。図 2 のような ER 図の各テーブル名やリレーションシップを参照する。バリデーションに関しては、そのカラムが not null かどうかを判別する。

続いては、画面遷移図よりコントローラーとビューを生成していく。図 3 に面談予約管理システムの画面遷移図の一部を示す。

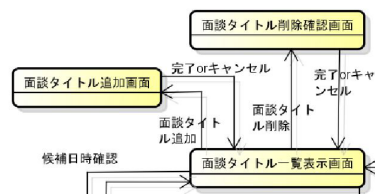


図 3 画面遷移図の一部

図 3 では面談タイトル一覧表示画面で面談タイトル追加ボタンを押すと面談タイトル追加画面へと遷移することが示されている。図 3 で示されているように、このままでは各画面が ER 図のどのテーブル

に値するのかが不明確なため、各画面のプロパティビューに予めテーブル名を記述しておく。図4にプロパティビューの例を示しておく。



図4 プロパティビューの例

画面遷移図からは、各画面の名前や処理を行った際の画面の遷移先を参照する。

3.5 生成されるもの

上記で述べた生成手順を踏むことで、ER図から各モデルのファイルが生成され、クラス名やメソッド名、バリデーションやアソシエーション等が記述される。

また、画面遷移図からは各ビュー、コントローラーのファイルが生成され、そのファイルの中に各機能の処理や画面を表示するためのコードが記述される。具体的な例としては、画面の名前に「追加」とあったらビューに追加画面を表示するため add ファイルを生成し、コントローラーにも add メソッドを記述し、対応するテーブルにデータを追加する処理を記述する。同様に、画面遷移図に「削除」とあった場合、ビューに delete ファイルを生成し、データの削除機能を搭載する。画面の名前に「一覧」と記述してあった場合はテーブルの全てのデータを一覧で表示する、といったように、画面遷移図から文字列を読み取ることで、各機能を搭載する。様々なシステム開発で作成された画面遷移図を参照していくことで、どの文字列

を抽出したらどの機能のソースコードを記述するかを決定していく。

ソースコードを生成する際に、コメントも記述していく。例えば CakePHP の命名規則を示したり、Form ヘルパーの使い方を示したり、モデル、ビュー、コントローラーの対応関係を示したりする。

4 コード生成実験

本研究では本ツールの有用性を示すために面談予約管理システムの作成を行う。この面談予約管理システムとは、学生が教員に面談を予約するときに利用するものであり、教員は面談候補日の登録、削除、編集、一覧表示機能、面談タイトルの登録、削除、一覧表示機能、面談情報の登録、削除、編集、一覧表示機能といった機能を利用することができる。学生はこの機能の一部を利用することができ、面談情報の登録、削除、編集機能を利用することができる。

この面談予約管理システムの画面遷移図と ER 図は図5, 6 に示されている。図5の画面遷移図には画面数が17個存在し、このシステム全体の画面遷移が示されている。また、図6のER図はテーブル数は4個存在し、各テーブル間の対応関係や、カラム名が記述されている。

本研究では画面遷移図と ER 図を作成し、そこから本ツールを適用して、CakePHP のソースコードを生成することでこの面談予約管理システムの作成を試みる。そして、複数の学生たちによって本ツールを使わずに作成された同様の機能を持つ面談予約管理システムのモデルやソースコードと比較することで、本ツールの有用性を確かめる。具体的には CakePHP の全体ファイル数、学生が変更・追加したファイル数、学生が記述したソースコード行数、学生が作成したデータベーステーブル数、画面数・遷移数等を比較する。

5 生成されるソースコード

上記で述べたコード生成実験で生成される想定ソースコードの一部を図7, 8 に示す。図6のER図の titles テーブルから CakePHP のモデルである図7の Title.php が生成され、図5の画面遷移図の面談タイトル追加画面から CakePHP のコントローラーで

ある図 8 の TitlesController.php が生成される。

6 終わりに

ER 図と画面遷移図からのコード生成ではコード生成ですべてのコードを自動生成できない。しかし、コード生成のためだけにモデルに追記する情報はほとんどなく、フレームワークを使用する際にありがちな冗長なコーディング作業は大幅に効率化できると考えられる。またコメントも充実した雛形となっているので、導入がしやすく機敏に開発を行うことができると思われる。学生たちが作成した同じ面談予約管理システムと比較すると、記述されているコードは宣言の仕方やデータの取得の仕方等に統一性があり、コメントも充実しているのも、他者がコードを読んでも理解がしやすいものだと思う。

今後の展望としては、さらに多くのシステム開発で作成されたモデルとソースコードを参照し、生成できるソースコードを増やしていき、さらに実際のシステム開発にこのツールを適用してもらうことで、

どの程度開発を円滑に進めることができるのかを検証していきたい。

参考文献

- [1] 河村美嗣, 浅見可津志, UML を入力とするソースコード自動生成ツールの開発, 全国大会講演論文集 第 72 回平成 22 年 (1), 一般社団法人情報処理学会, 2010, p1-337-p1-338, <http://ci.nii.ac.jp/naid/110008105498#ref>, 参照 (2014-7-17)
- [2] 八木紀幸, 中所武司, モデル変換に基づくエンドユーザ主導の Web アプリケーション開発技法, 情報処理学会研究報告. ソフトウェア工学研究会報告, 一般社団法人情報処理学会, 2009, no.31, p81-p86, <http://ci.nii.ac.jp/naid/110007333794>, 参照 (2014-07-17)
- [3] キヤノンソフトウェア株式会社, Web Performer, http://www.canon-soft.co.jp/product/web_performer/, 参照 (2014-07-16)
- [4] 株式会社ジャスミンソフト, Wagby, <http://wagby.com/>, 参照 (2014-07-16)
- [5] 伊賀敏樹, blanco Framework, <http://www.igapyon.jp/blanco/blanco.ja.html>, 参照 (2014-07-16)


```

<?php
App::uses('AppModel', 'Model');

class Title extends AppModel {
    public $name = 'Title';
    var $hasMany = array(
        'Candidate' => array(
            'className' => 'Candidate',
            'foreignkey'=>'title_id'
        )
    );
}

```

図 7 Title.php の一部

```

public function add(){
    if ($this->request->is('post')) {
        $this->Title->create();
        if ($this->Title->save($this->request->data)) {
            $this->Session->setFlash('面談タイトルが保存されました。');
            $this->redirect(array('action' => 'index', $this->Title->id));
        } else {
            $this->Session->setFlash('面談タイトルを保存できませんでした。もう一度試して下さい。');
        }
    }
}

```

図 8 TitlesController.php の一部