

プロトタイプ開発支援ツールによる CakePHP フレームワーク向けコード生成実験

京谷 和明 伊藤 恵 奥野 拓

近年、システム開発では効率化やソースコードの可用性を高めるために、様々な自動生成ツールやフレームワークが導入されている。一方で、顧客から依頼を受けてのシステム開発は、早い段階でプロトタイプシステムを開発し、システムの詳細な要件の確認を行う場面が増えている。しかし、プロトタイプシステムの開発に自動生成ツールやフレームワークを導入する場合、学習コストがかかってしまうことが多く、導入が難しい。そこで、本研究ではシステム開発を行う際に作成されることが多く、理解もしやすい画面遷移図と ER 図からフレームワーク用のコードを生成することで、迅速にプロトタイプのシステムを作成することができ、仕様変更があった際にも図を書き換えればすぐにシステムに反映させることのできるツールの開発を行った。本稿では複数の被験者にツールを使用してもらい、有用性の確認を行うことについて述べる。

Recent years, various code generation tools and development frameworks are introduced to smooth system development and enhance availability of source codes. On the other hand, in system development by clients' requests, it is increased that the detail requirements are checked using prototypes in early development phase. However, if code generation tools and frameworks are introduced in development of prototype system, learning cost are often increased. Therefore, this research creates a tool that can generate prototype system codes rapidly and the cost of learning this tool is low. The input models of this tool are ER diagram and screen transition diagram which are often created by system developments. The output of this tool is source codes for CakePHP framework. This tool enables developer to develop prototype system smoothly, and to modify a system rapidly when the requirements are changed. In this paper, we describe the experimental confirmation usefulness of this tool to be used by subjects.

1 背景

システム開発ではより一層の短納期化、高品質化が求められており、クライアントがベンダーに対し、迅速且つ正確に要求通りのシステムを構築することを求めている。この目的達成のために、システム開発ではソースコードの自動生成ツールの導入が行われたり、プロトタイプシステムの開発が行われていたりする。

ソースコードの自動生成ツールを導入するメリットとしては、設計図やドキュメントを活かすことで開発工数を減らせることである。デメリットとしては、生

成率に重点を置く自動生成ツールが多いため、設計を細部まで行わなければうまく自動生成を行えないことが多いことである。また、自動生成ツールを扱えるようになるために1ヶ月といった工数がかかる場合もある。

プロトタイプシステムは早期に大まかに動くものが開発され、それを顧客に見せ、フィードバックを得ることで顧客の真の要求を引き出していくことができる。そのため、自動生成ツールをプロトタイプシステム開発の場で導入することは難しい。

2 関連研究

2.1 UML を入力とするソースコード自動生成ツールの開発

河村, 浅見らの研究[1]では Web アプリケーションの3層アーキテクチャの構造に着目し、クラス図

An Experiment of Code Generation for CakePHP Framework Using a Prototype Development Tool.

Kazuaki Kyoya, Kei Ito, Taku Okuno, 公立はこだて未来大学システム情報科学研究科, Graduate School of Systems Information Science, Future University Hakodate.

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フレームワークを扱う場合、ある程度そのフレームワークを使ったシステム開発の経験が無いと作成が難しいことがある。

2.2 フレームワークを用いた JavaWeb アプリケーション自動生成システムの試作と評価

菰田, 中所らの研究[2]ではシステムエンジニアの生産効率を向上させるために、Web アプリケーションフレームワークである Struts と O/R マッピングフレームワークである Hibernate を用いて、Excel のワークシートからデータベースとビジネスロジックを自動構築できるシステムの開発を行った。このシステムはコーディング時間を削減することを意図したものであるが、開発対象の規模が大きくなると Excel による設計書では視認性などの問題が発生し、記述が困難であると予想される。

2.3 Web Performer

本研究の関連製品としては、キャノン IT ソリューションズ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 100%の自動生成を行うには細部に渡って設計を行わなければならない、プロトタイプシステム開発の導入には向いていない、

3 目的

既存の自動生成ツールではソースコードの自動生成を行うためのモデルやドキュメントを作成するのに時間がかかってしまうものが多い。そこで本研究ではプロトタイプシステムを迅速に開発できるツールを作成することで迅速に顧客の真の要求を聞き出すことができるようにすることを目的とした。具体的には設計を容易に行うことのできるモデルから、迅速にフレームワーク用のソースコードを自動生成できるよ

うにし、修正も迅速に行うことができるようにする。なお、本論文におけるプロトタイプシステムとしては、主にデータアクセス周りにおける機能を考慮しているため、レイアウトや画面内における振る舞いは考慮していない。

4 アプローチ

本研究では、プロトタイプのシステムを迅速に開発できることに主眼を置く。単にコード生成率を重視するのではなく、必要とするモデルを簡単にし、少ない手順でフレームワーク用のコードを自動生成できるようにする。

4.1 アプリケーションフレームワークとモデル

本研究で扱うモデルは画面遷移図と ER 図である。データベースを利用した Web アプリケーションの開発を行う際に作成されることが多く、クラス図やシーケンス図と違い、扱うフレームワークに詳しくなくても作成を行いやすい。画面遷移図は UML のステートマシン図を基に作成を行う。

本研究で扱うアプリケーションフレームワークは CakePHP である。公式サイトからファイルをダウンロードし、開発したいシステムに合わせてファイルを追加し、コードを記述することでシステム開発が行える。CakePHP は MVC アーキテクチャに則ったフレームワークであり、データの処理を Model、画面表示処理を View、データの受け渡しを Controller に記述する。本研究ではこの MVC アーキテクチャの構造に着目し、画面遷移図と ER 図から、CakePHP のソースコードを自動生成する。

4.2 ツール構成図

ツールは Java 言語を用いた Web アプリケーションとして作成を行う。画面遷移図と ER 図は、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*[4] を使って作成する。astah*で記述された画面遷移図と ER 図の astah*ファイルをブラウザで選択することで自動生成を行うことができる。各画面の名前や、テーブル名からファイルが生成され、ツール内に埋め込まれる CakePHP のソースコードがそ

のファイルに記述され、それらのファイルが ZIP 圧縮されるため、その ZIP ファイルをダウンロードすることで利用することができる。図 1 に本ツールの構成図を示す。

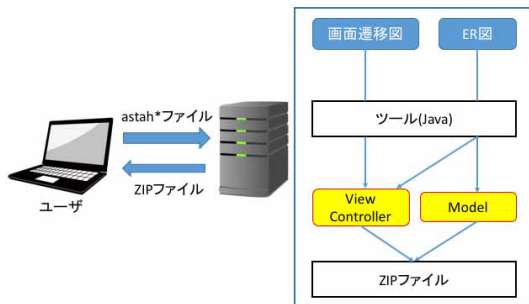


図 1 ツールの構成図

4.3 ツールによるソースコード自動生成

ER 図からは、主に Model に関するソースコードの自動生成を行う。ER 図のテーブル名から各 Model ファイルを生成し、リレーションシップが存在した場合は、それらに関するコードも自動生成する。また、astah*ではカラムのデータの型や、空の値を許すか否かを記述することもできるので、バリデーションに関するコードも Model に生成する。

画面遷移図からは、View と Controller に関するソースコードの自動生成を行う。画面の名前に「一覧」と記述されていたら Controller と View にテーブルの中身を一覧表示させるコードを生成し、「追加」と記述されていたら入力フォームを表示させるコードを生成し、フォームの中身をテーブルに追加するコードを生成する。本ツールは現在、一覧表示、追加、削除、全削除、編集、履歴表示、ログイン機能を生成することができる。ツールに予め登録されている「一覧」、「追加」等の単語が画面の名前に含まれているか調べることで、生成される機能が決定される。画面遷移図からは他にも遷移情報を読み取り、どの画面や機能を使うことで、どの画面に遷移するのかを決定する。画面遷移図だけではどのテーブルに対して処理を行うのが不明なため、各画面のプロパティ領域の定

義の項目に ER 図と対応するテーブル名を記述しておく。プロパティ領域への記入例は図 2 の下線が引いてある場所に示す。

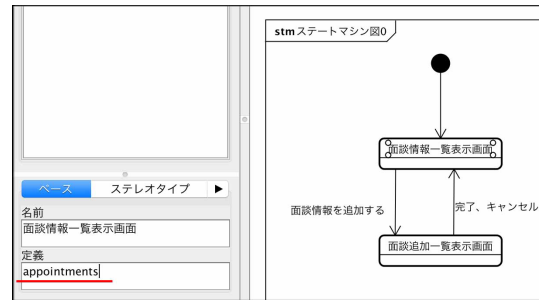


図 2 プロパティ領域への記入例

開発者は CakePHP の知識が少なくても、astah*でこれらの画面遷移図と ER 図を作成し、本ツールでそれらのファイルを選択することで、即座に自動生成が行えるようになる。また、astah*では ER 図から SQL へエクスポートすることも可能なので、データベースの構築も自動化を行うことができる。

4.4 マニュアルの作成

本ツールを円滑に使用できるように、本ツールのマニュアルの作成を行った。本ツールでは画面遷移図と ER 図の対応付けをさせる必要があったり、テーブル名は複数形で記入する必要があるなどの制限が存在したりするため、マニュアルを読み理解する工数をなるべく減らすことができるように作成した。

5 実験 1

本ツールの有用性を確認するために、面談予約管理システムの自動生成を行い、顧客側からシステム変更の依頼を受けた際に、どの程度機敏にシステムの改変を行えるのかの評価を行った。また、CakePHP のソースコードを簡便に自動生成でき、プロトタイプ開発にも使用されることのある bake と呼ばれるツールでも同様にプロトタイプ生成を行い、本ツールとの比較を行った。

5.1 実験の手順

面談予約管理システムとは、学生が教員に対して面談の予約を行いたいときに利用できる Web アプリケーションである。教員が顧客として面談情報の登録、削除、編集、一覧表示、全削除、履歴表示、面談候補日の登録、削除、編集、ユーザの追加、ログインができるシステム開発の依頼をし、それに対して画面遷移図(図5)とER図(図6)の作成を行い、それらのモデルに本ツールを適用した。

生成されたプロトタイプを見て顧客から、ユーザ情報に性別情報を追加したい、との追加要求を受け、ER図の修正を行い、再度本ツールを使って自動生成を行った。さらに、面談タイトルも追加や削除ができるようにしてほしい、との追加要求を受け、画面遷移図の修正を行い、再度本ツールを使って自動生成を行った。画面遷移図とER図の修正箇所は図7の通りである。

同様に bake を使って自動生成を行い、修正時間や生成量等を比較した。

5.2 本ツールによる生成結果

図5と図6のモデルに対して本ツールを適用したところ、21個のファイルが自動生成された。仕様通りのコードが全て生成されたわけではないが、必要となるデータアクセス周りの機能は全て自動生成を行うことができた。仕様通りに生成されなかった点は、面談情報の一覧表示の際に本来は今日以降の情報のみを表示する仕様だが全ての面談情報を表示するよう生成された点や、本来は教員と学生とで閲覧できる画面に差があるがすべて共通で生成された点などであった。

続いて、ER図に修正を加えて自動生成を行った結果、ファイル数に違いはなかったが、コードの行数に違いが出た。違いが出た箇所としては、ユーザの登録の際に性別の登録や編集ができるよう、2行のコードが生成されていた。Modelには、性別を登録する際にフォームが空の値を許さないように、バリデーションが1行生成された。他にも面談情報の一覧表示や詳細表示画面で性別が表示できるように、6行のコードが生成された。

画面遷移図に修正を加えて自動生成を行った結果

は、ファイル数にも違いが出た。1回目で生成されたファイルに加え、3つのファイルが自動生成されており、行数としては129行のコードが生成されていた。面談タイトルのViewと、Controllerに面談タイトルの追加、一覧表示、削除の処理ができるようになっており、面談情報と、面談候補日の一覧表示画面には、面談タイトルの一覧表示画面に遷移できるよう、それぞれ1行のコードが生成されていた。

5.3 bakeによる生成結果

bakeでも同様の手順で自動生成を行った。bakeではモデルを用意する必要はなく、データベースを用意し、環境変数などを設定することで扱えるようになる。bakeで自動生成できる機能は、本ツールと同様にデータアクセス周りの機能であり、追加、削除、編集、一覧表示、詳細表示の5つの機能を一括で生成する。

1回目の生成では、まずデータベースとの接続の設定を行った。ドライバや接続ポートなどの10個の項目について問われるため、それらに答えていくことでデータベースとの接続を行った。その後MVCのどの部分を生成するのかを尋ねられるので、それに答えていくことで自動生成を行った。例えばModelを生成する場合はModelを選択し、どのテーブルを生成対象とするか、バリデーションを設定するか、アソシエーション(関係)を生成するかどうかの3つの項目を尋ねられた。バリデーションは各カラムに対し、1つずつ指定する必要がある。Model同士のアソシエーションも、各テーブルに対し1つずつ答える必要があった。設計書がないため、頭のなかで考えてアソシエーションを答える必要があった。ControllerやViewも同様に対応するテーブルを指定し、scaffoldingやCRUDを作るか否か、ヘルパーやコンポーネントを扱うかなどを答えていく必要があった。Controllerは7つの項目、Viewは2つの項目について尋ねられた。モデルの作成をしなくても自動生成を行うことができたが、上記の手順を踏まなければならないため、本ツールで自動生成した場合より少し手間がかからない程度であった。

2回目の自動生成ではデータベースの設定を行い、

ユーザの Model と View ファイルに対して自動生成を行った。バリデーションに関するコードが2行、性別の登録フォームに関するコードが2行生成されていた。全ての Model, View に対して自動生成を行う必要はなかったが、修正が及ぶ Model と View に対しては、また最初からどのように自動生成を行うのかを指定しなかなければならなかった。

3回目の自動生成では面談タイトルの View と Controller に対して新規に自動生成を行った。この自動生成において317行ものコードが自動生成されたが、詳細表示や編集などの必要のない機能やそれらへの画面遷移が生成されていたため、不要なコードが多かった。

5.4 考察

1回目の自動生成では bake のほうがやや早く自動生成を行えたが、本ツールではログイン機能や全削除機能、履歴表示機能も自動生成できていた。また、モデルの存在により、可視化や意識共有といった観点から見ても本ツールのほうが有用性が高いと思われる。2回目の自動生成では、本ツールでは ER 図に性別カラムを1行追加するだけですぐに自動生成を行うことができ、複数の箇所の修正も一括で行われていた。bake では開発者が修正が及びそうな箇所を考え、1つずつ自動生成しなければならなかったため、本ツールのほうが迅速に修正を行うことができた。3回目の自動生成では新たに画面を3つ作成しなければならなかったため、手間としては bake による自動生成とあまり変わらなかった。しかし、画面遷移図があるおかげで本ツールのほうが柔軟に必要な機能だけを自動生成できた。bake ではプロトタイプをそのまま実システムの開発に使うことになった場合、不要な機能を削除する作業が入る。また、修正が広範囲に及ぶ場合、bake では修正のし忘れが発生してしまう恐れがあるが、本ツールではモデルの存在により修正のし忘れを予防することができる。これらのことから、bake と比較すると、画面遷移図と ER 図から直接迅速に自動生成を行うため、本ツールのほうが有用性が高いと思われる。

6 実験 2

学生5名、教員1名に本ツールを利用してもらい、問題点の抽出を行った。自動生成を行うのは、予めこちらで用意した画面遷移図と ER 図である。約1.5時間をかけてマニュアルに則った自動生成を行ってもらい、生成コードを含む ZIP ファイルをダウンロードし、実際にサーバにデプロイしてアプリを動作させるまでの手順で不明点がないか、マニュアルに不備がないかなどの確認を行った。なお、動作確認に用いたのは PaaS の1種である C4SA [5] である。今回は本ツールのマニュアルとは別に、C4SA でアプリケーションを動作させるまでの手順を書いたマニュアルも用意した。

6.1 実験結果

CakePHP を利用したことのある被験者1名はアプリケーションの動作まで行うことができたが、他の被験者は時間内にアプリケーションの動作を行うことができなかった。原因としては CakePHP を利用したことのある被験者は CakePHP の環境構築に慣れているためすぐに動作させることができたが、CakePHP を利用したことのない被験者は環境構築に慣れておらず、アプリケーションを動作させるまでの過程に時間がかかってしまったことが挙げられる。

具体的に CakePHP を動かすまでの過程で時間がかかってしまった場面としては、本ツールでは Model, View, Controller 部分のコードのみを生成していたので、CakePHP 本体とマージするまでの手順が発生してしまっていたことが挙げられる。アプリケーションの動作を行うことのできた被験者は CakePHP のファイル構造を把握していたため、どのファイルをどのフォルダに追加すればよいか把握できていたが、他の被験者はあまり把握できていなかった。

他には、astah*では ER 図から SQL の自動生成を行うことができたが、別途 CakePHP 用のデータベースの設定を行わなければいけなかったことが挙げられる。CakePHP では主キーである id を AUTO_INCREMENT にする必要があるが、astah*ではその部分を自動生成できない。C4SA ではデー

データベースサーバに MySQL を用いており、クライアントツールに phpMyAdmin を用いていたため、マニュアルには phpMyAdmin 上で設定するように記述していたが、phpMyAdmin の操作をしたことのない被験者は設定に時間がかかってしまっていた。また、CakePHP 上でデータベース接続用のコードも自分で記述する必要があった。

他にも CakePHP ではハッシュ化や暗号化を行うために、予め記述されている Security.salt の値と Security.cipherSeed の値を修正する必要があることが挙げられる。マニュアルではどの箇所を修正すればよいのかを記述し、各自でエディタを用意し修正するように記述していたため、その部分に関しても時間がかかってしまっていた。

6.2 ツールの改善

これらの結果を受けて、本ツールの改善を行った。具体的には CakePHP 本体を本ツールに内蔵することで、自動生成を行ったファイルとのマージを本ツール上で自動で行うようにした。本ツールの利用者は Model, View, Controller ファイルのみをダウンロードするか、CakePHP 本体ごと全部ダウンロードするか選択できるようにした。

また、SQL の生成は astah* の ER 図上で SQL エクスポートを行うのではなく、本ツールで ER 図から読み込んだデータを基に自動生成を行い、生成コードを含む ZIP ファイルと同時に SQL ファイルもダウンロードできるようにした。phpMyAdmin 上では SQL のインポートのみを行うことで、必要最低限の設定でデータベースとの接続ができるようにした。

他にも、Security.salt の値と Security.cipherSeed の値も本ツールの利用者が各々エディタを用意し修正を行うのではなく、ブラウザ上で入力フォームを追加し、自動で該当箇所を修正するよう自動生成を行うように修正した。改善を行った本ツールは図 3 に示す。

7 実験 3

改善を行った本ツールを用いて再度被験者を募集し、生成実験を行った。今回の被験者は学生 2 名で、3 時間以内を目安に自動生成を行ってもらった。今回



図 3 改善を行ったツールの画面

扱う画面遷移図と ER 図も予め用意した面談予約管理システムのものであるが、被験者に画面遷移図と ER 図の修正を 2 回行ってもらい、そこから自動生成を行いアプリケーションの動作を行ってもらった。予め用意した ER 図は図 6 と同じものであるが、画面遷移図は図 7 を用いた。動作環境は実験 2 と同様の C4SA である。まず最初に本ツールのマニュアルを読んでもらい、それを読み終え次第予め用意した画面遷移図と ER 図を基に自動生成を行ってもらった。その後ユーザ情報に性別情報も登録できるようにしてほしいという要求を与え、自動生成を行い、動作確認を行ってもらった。さらに面談タイトルについても変更を加えられるようにしてほしいという要求を与え、自動生成を行い、動作確認を行ってもらった。実験の様子は図 4 に示す。

7.1 1 回目の自動生成

1 回目の自動生成では、まず本ツールのマニュアルを読んでもらった。マニュアルの読解にかかった時間は被験者 1 は約 15 分程で、被験者 2 は 20 分程であった。その後本ツールを用いて自動生成を行ってもらった。自動生成にかかった時間はいずれの被験者も 5 分程であった。その後 C4SA のマニュアルを基に、C4SA 上に自動生成を行ったソースコードをアップロードし、動作確認を行ってもらった。被験者 1 はアップロードした ZIP ファイルが C4SA 上でうまく



図 4 実験の様子

展開できなかった問題が発生し、被験者 2 は自動生成を行うときに Model, View, Controller ファイルのみをダウンロードするように選択してしまっていた。動作確認を行うまでかかった時間は被験者 1 が 1 時間程で、被験者 2 が 55 分程であった。

7.2 2 回目の自動生成

2 回目の自動生成では、ユーザ情報に性別情報を登録できるように、ER 図を修正してから自動生成を行い、性別情報を登録できるかの動作確認を行うように要求を与えた。この性別情報は入力が必要であることも伝えた。被験者 1、被験者 2 のいずれも予め用意した ER 図の情報を基に、新たに users テーブルにカラムを追加していた。ER 図の他のカラムの情報を見ながら作成を行っていたため、どの部分に情報を付加すれば入力が必要になるかすぐに気づいたようである。被験者 1 は修正を行ってから C4SA 上で動作確認を行うまでに 15 分程の時間を要した。被験者 2 は users テーブルの名前を user と変えてしまい、うまく動作確認を行うことができていなかった。この修正は意図的に行ったものではなく、誤って修正してしまったとのことであった。被験者 2 は正常に動作確認を行えるようになるまで 30 分程の時間を要した。

7.3 3 回目の自動生成

3 回目の自動生成では、面談タイトルの変更機能を作成するように、画面遷移図を修正してから自動生成

を行い、面談タイトルが変更できるようになるかの動作確認を行うように要求を与えた。被験者 1、被験者 2 のいずれも予め用意した画面遷移図の情報を基に、画面遷移図に新たに「面談タイトル変更画面」という画面を追加し、別の画面から遷移するように作成していた。被験者 1 は面談タイトル追加画面から変更画面に遷移するように記述を行っていたため、うまく動作させることができていなかったが、画面遷移図を確認し、すぐに面談タイトル一覧表示画面から遷移するように修正を行っていた。被験者 1、被験者 2 共に新たに作成した画面のプロパティ領域に対応する ER 図のテーブル名を記述していなかったため、うまく動作させることができていなかった。その後本ツールのマニュアルを読み気づいたようであり、画面遷移図に対応するテーブル名を記入し、うまく動作させることができていた。正常に動作できていることが確認できるまでにかかった時間は被験者 1 は 40 分、被験者 2 は 45 分であった。

7.4 考察

1 回目の自動生成ではいずれの被験者もマニュアルの読解も含めて 1.5 時間以内に動作確認を行うことができていた。実験 2 で行ったツールの改善により、実験 2 と比較し、CakePHP を動作させるまでの手順を減らすことができ、動作確認を行うまでの時間を減らすことができた。

2 回目の自動生成では被験者 1 は ER 図にカラムを追加し、すぐに自動生成を行うことができたため、短い時間で要求通りの自動生成を行うことができていたが、被験者 2 は ER 図を誤って修正してしまった部分があり、正常に動作確認を行うまでに時間を要してしまっていた。本ツールで自動生成を行う際に、ER 図が正しく記述されていない場合に利用者がすぐ気づくようにどの部分が正しく記述されていないかエラー表示などをする必要があると思われる。

3 回目の自動生成では被験者 1、被験者 2 共に ER 図との対応付けを行うのに時間を要したため、正常に動作させるまでに時間がかかってしまっていた。マニュアルには ER 図との対応付けの仕方が載っていたため、すぐに気づくことができるようにマニュアルの

改善も行う必要があると思われる。

以上のことから 2 回目, 3 回目の自動生成では時間がかかってしまう場面があったが, エラー表示を含むツールの UI やマニュアルの改善によって時間を削ることができると思われるため, 本ツールをさらに改善する必要があることが示された。

8 まとめと今後の課題

本研究では既存の自動生成ツールではプロトタイプシステムの開発に時間を要してしまうという問題から, 画面遷移図と ER 図から CakePHP 用のソースコードを容易に自動生成できるツールの開発を行った。実験 1 では bake との自動生成実験の比較を行い, 本ツールの有用性を確認することができた。実験 2 では被験者を募集し, 本ツールを利用して自動生成を行ってもらったが, 時間内に正常に動作することができた被験者が少なかったため, ツールの改善を行った。実験 3 では再度被験者を募集し, 改善した本ツールを利用することで, 動作確認を行うまでにかかる時間を減らすことができていることを確認することができた。また, 画面遷移図と ER 図の修正を行ってもらい, そこから自動生成を行ってもらったが, マニュアルの修正も含めて本ツールをさらに改善していく必要があることが示された。

今後の展望としてはさらに本ツールの修正を行い, 面談予約管理システム以外のシステムにも本ツールを適用してみる生成実験を行いたい。その後実際の要求獲得の場で本ツールを用いることで, 本ツールの有用性を示していきたい。また, CakePHP 用のコードだけでなく, スマートフォンアプリ等の他の MVC アーキテクチャを用いたフレームワーク用のソースコードの自動生成を試みることで, 他のフレームワークでも同様なことが行えることを示していきたい。

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, 一般社団法人情報処理学会, 2010, p337-p338
- [2] 菰田直樹, 中所武司, フレームワークを用いた JavaWeb アプリケーション自動生成システムの試作と評価, 全国大会講演論文集, 第 71 回, 一般社団法人情報処理学会, 2009, p289-p290

- [3] キヤノン IT ソリューションズ株式会社, Web Performer, http://www.canon-soft.co.jp/product/web_performer/, 参照 (2016-8-16)
- [4] 株式会社チェンジビジョン, astah*, <http://astah.change-vision.com/ja/>, 参照 (2016-8-16)
- [5] ニフティ株式会社, ニフティクラウド C4SA , <http://c4sa.nifty.com/>, 参照 (2016-8-16)

A 付録: 本稿で用いたモデル

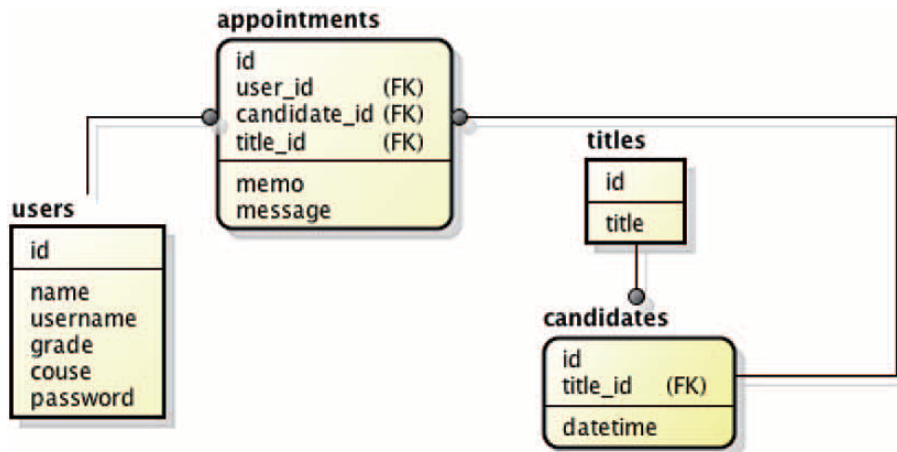


図 6 面談予約管理システムの ER 図

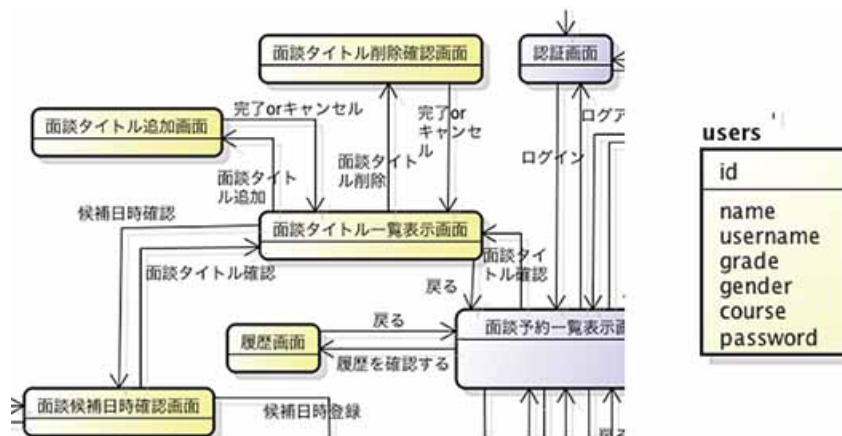


図 7 修正された画面遷移図と ER 図の一部