

フレームワーク

岡田 典子, 雨宮 明日香
Noriko OKADA, Asuka AMAMIYA

1 はじめに

近年, インターネット技術の普及にともない, 受注管理システム, および物品管理システムなどの企業内の基幹業務システムを, Web アプリケーションとして構築する企業が増えてきている. また, ビジネスの急速な変化にともない, Web アプリケーション開発は大規模化, および複雑化が進み, その一方で開発の短期化が求められている. そこで, 現在の Web アプリケーション開発において, フレームワークを用いた開発スタイルが注目されている. フレームワークとは, アプリケーションソフト開発の際に必要なとされる汎用的な機能をまとめて提供するアプリケーションの雛形であり, これを用いることで, 開発工数の削減だけでなく, 技術リスクを減らし, 品質を向上させることが可能となる.

本報告では, フレームワークの中でも特に Web アプリケーションフレームワークについて着目し, その概念, 機能, および各プログラミング言語におけるフレームワークについて解説する.

2 フレームワーク

ソフトウェア開発におけるフレームワークとは, アプリケーションソフトを開発する際に必要とされる汎用的な機能をまとめて提供し, アプリケーションの土台として機能するソフトウェアのことである. プログラマはフレームワークの決まりにしたがって実装を行うだけで, 一定の品質をもつアプリケーションを作ることが可能となる.

フレームワークを導入することにより, 保守やアプリケーション統合を容易にしたり, ノウハウを次の開発に流用することなどが可能となるため, 開発工数の削減, 開発リスクの回避, および品質の向上が期待できる. 一方で, 開発初期時には一定の工数がかかることが問題として挙げられる.

3 Web アプリケーションフレームワーク

Web サイト, および Web サービスなどの開発をサポートするために設計されたフレームワークを, Web アプリケーションフレームワークという. Web アプリケーションフレームワークは, Ruby, および Python など多数のプログラミング言語に対して存在し, その多くが MVC モデルをベースとしている.

3.1 MVC(Model-View-Controller) モデル

MVC モデルとは, アプリケーションの構成要素を命令の処理, 表示, および制御の 3 つに分けて設計開発を

行うという考え方である. 以下に MVC モデルの各要素について述べる.

モデル ユーザから入力されたデータなどを用いて, データベースの更新などの処理を行う. データベースとの接続機能はモデルのみが持ち, この機能を O/R(Object/Relational) マッピング・ツールによって実装している. O/R マッピング・ツールについては, 3.2 節で述べる.

ビュー 画面の表示, および出力を行う. 画面の表示には, テンプレート・エンジンという仕組みを用いている. テンプレート・エンジンについては, 3.3 節で述べる.

コントローラ ユーザの画面操作による入力を受け取り, モデル, およびビューの制御を行う.

Fig. 1 に MVC モデルの構造を示し, 以下に MVC モデルの処理の流れを示す.

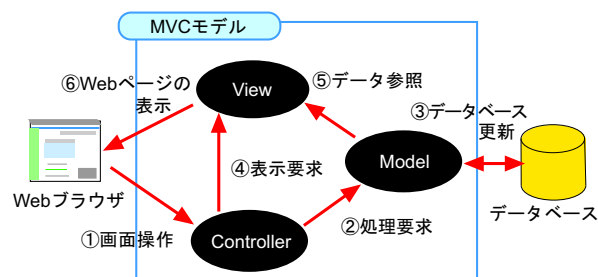


Fig.1 MVC モデルの構造 (出典: 自作)

1. ユーザの画面操作による入力をコントローラが受け取る.
2. コントローラが入力内容を判断し, モデルに処理を依頼する.
3. モデルがデータベース上のデータを参照, および更新しながら処理を実行する.
4. コントローラがビューに表示の更新を依頼する.
5. ビューが更新に必要な処理結果をモデルから参照する.
6. モデルから参照した情報を元に, ビューが画面表示の更新を行う.

MVC モデルを用いることによるメリットとして, 機能を分離することで各機能の独立性が確保されること, および仕様変更の影響を軽減できることなどが挙げられる.

3.2 O/R マッピング・ツール

O/R マッピング・ツールは, データベースにデータを格納する際に必要なモデルのオブジェクトとリレーシ

ナル・データベースのテーブルを関連付けるものである。O/R マッピング・ツールを内蔵したフレームワークは、データの登録、読み出し、変更、および削除などの処理に対して適切な SQL 文を組み立て、データベースとやり取りを行う。そのため、開発者は形式化されたデータ処理をフレームワークにまかせ、アプリケーションの設計や論理の実装に専念することができる。

3.3 テンプレート・エンジン

テンプレート・エンジンとは、HTML で記述された複数の Web ページの共通部分からなるテンプレートをあらかじめ用意しておき、Web ページを出力する際にテンプレートに実データの埋込処理を行うことで最終的な HTML を完成させるというものである。これをフレームワークに組み込むことで、Web ページのデザインとプログラムの分離が実現し、Web デザイナーとプログラマーという異なるスキルを持った人がそれぞれの得意分野で能力を発揮することが可能となる。

4 各言語環境におけるフレームワーク

フレームワークは、Ruby、および Python などの各プログラミング言語に対して存在する。ここでは、軽量型言語で記述されたフレームワークとして、Ruby の Ruby on Rails、Python の TurboGears、および Django について述べる。

4.1 Ruby on Rails(RoR)

RoR は、オブジェクト指向のスクリプト言語である Ruby によって記述された Web アプリケーションフレームワークである。Ruby の作者が日本人であるため、日本では特に盛んに利用されている。RoR では、コードの自動生成機能が整備されており、開発者は少ないコードの記述量で簡単に開発を行うことができる。一方で、消費するメモリ量が多いという問題が挙げられる。

RoR は Convention over Configuration、および Don't Repeat Yourself という 2 つの大きな方針に基づいて設計されている。両者について以下に詳しく述べる。

- Convention over Configuration(CoC)

従来のフレームワークでは、XML 形式の設定ファイルを大量に作成する必要があることによる負担が大きいという欠点があった。RoR では、この負担を軽減するために、開発者によって設定を指示されなかった場合の挙動を、CoC（設定に勝る規約）として定めている。この規約により、不要な設定ファイルの準備や重複したコードの記述が自然かつ大幅に抑制され、その結果、開発工数およびコストの削減が可能となる。

- Don't Repeat Yourself (DRY)

DRY は、同じ作業を繰り返すべきではないという設計思想である。RoR では、ディレクトリの配置や命令規則などが詳細に定められている。それにしたがって開発を進めることにより、ある情報に対するコードが 1 カ所に集約される。これにより、冗長

なコードを省略し、内容の変更を一括して行うことが可能となる。

4.2 TurboGears

TurboGears とは、Python における Web アプリケーションフレームワークである。TurboGears は、フレームワークを構成する部品として、既存のツール、およびライブラリを組み込んで活用しているメガフレームワークである。また、O/R マッピング・ツール、およびテンプレート・エンジンなどの一部の部品は、必要に応じて取り替えることができる。

4.3 Django

Django は、TurboGears と同様に Python で記述されたオープンソースの Web アプリケーションフレームワークである。ニュースサイトを運営するために作成されたフレームワークであるため、データベース主体の Web サイトの構築を容易に行うことができる。また、既存のツールを組み合わせで構成されている TurboGears に対し、大半の機能が Django プロジェクト内で開発されているため、機能間の統一性がとれている。

5 まとめ

本報告では、アプリケーションソフト開発の際に必要な汎用的な機能をまとめて提供し、アプリケーションの土台として機能するソフトウェアであるフレームワークについて調査した。フレームワークを用いてアプリケーション開発を行うことにより、開発工数の短縮化、および開発リスクの回避などのメリットが見込める。Web アプリケーションフレームワークの多くは、MVC モデルをベースとしており、O/R マッピング・ツール、およびテンプレート・エンジンなどの機能を備えている。今後、現在日本で多く利用されている Ruby のフレームワークである RoR に続き、Python の TurboGears、および Django などをはじめとする軽量型言語で記述されたフレームワークがさらに普及していくと予測される。

参考文献

- 1) Web アプリケーション・フレームワーク入門
<http://itpro.nikkeibp.co.jp/article/COLUMN/20071011/284283/?P=1&ST=develop>
- 2) ThinkIT, Rails が注目されている理由
<http://www.thinkit.co.jp/free/article/0605/2/1/>
- 3) もっと知りたい Python, 第 3 回 Web フレームワーク TurboGears, Django, Pylons
<http://gihyo.jp/dev/feature/01/python/0003>
- 4) TECHSCORE, MVC モデル
<http://www.techscore.com/tech/J2EETPattern/introduction.html>
- 5) 法林浩之他, 最新 LL フレームワークエクスペローラへ Ruby on Rails, Maple/Ethna, TurboGears, Catalyst 5 大フレームワーク徹底攻略, 技術評論社, 2006 年