

CAPTER 11

Model, View, Controller

ウェブアプリを model, view, controller という三つの要素で組み立てる方法を紹介します。一見複雑に見えますが、この枠組みで作ると、後の修正や拡張、部品の再利用が容易になります。

11.1 MVC とは何か、どう実装するのか

11.1.1 Monolithic JSP

10 章 (p.119) で作成したウェブアプリのサーバ側は、ただ一つの JSP (zip.jsp) からできていました。zip.jsp が担っていたのは次の処理です。

1. クライアントからデータを受信する
2. データベースへ問い合わせる
3. 結果をクライアントに返す

このように、処理のすべてを担うような JSP は、Monolithic JSP と呼ばれ、ウェブアプリ開発のアンチパターン（よい設計であるデザインパターンの対極）とされています（図 11.1）[47]。アプリケーションが複雑になると、JSP は手に負えないくらい複雑になってしまうからです。そうならないために、MVC というデザインパターンに沿ってアプリケーションを実装します。



図 11.1 Monolithic JSP

11.1.2 Model-View-Controller

ウェブアプリにおける MVC とは、おおざっぱに言えば、アプリケーションの各役割を次のように分担することです（図 11.2）。

Controller：クライアントからのデータを受け取ります。実装は Servlet です。

Model：メイン・タスク（ビジネスロジック）を担います。実装は JavaBeans です¹⁾。

View : クライアントに返す画面を生成します . 実装は JSP です .

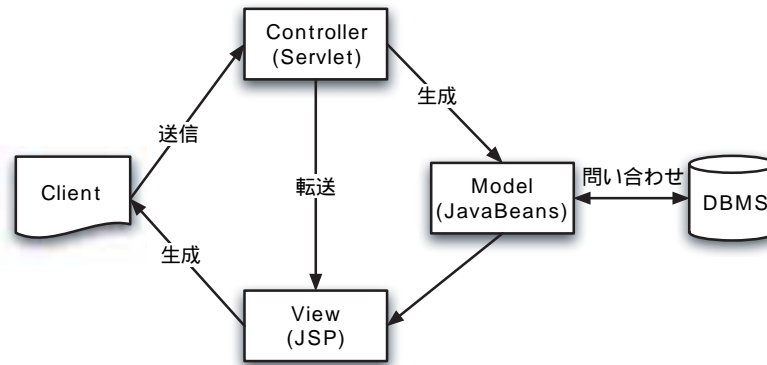


図 11.2 Model-View-Controller

11.2 郵便番号検索の MVC による実装

先の郵便番号検索システムを MVC で実装すると、各コンポーネントの動作は次のようになります (図 11.3) .

1. Client: zip.Controller にデータ (code=194) を送信する .
2. Controller: zip.Model を生成する .
3. Controller: zip.Model のパラメータを設定する (code="194") .
4. Controller: zip.Model のメイン・タスク (検索) を実行させる .
5. Model: データベースに問い合わせる . (SELECT 文の実行)
6. Controller: zip.Model を一時保管する .
7. Controller: View.jsp に移行する .
8. View: 保管された zip.Model を取り出す .
9. View: zip.Model から情報を取り出しながらページを生成する .

これだけみると非常に複雑になってしまったように思うかもしれませんが、しかし、こうすることで、実装がとても楽になるのです。アプリケーションが複雑になっても、基本的な枠組みはこれ以上複雑にはなりません。

1) Model は JavaBeans ではなく、EJB (Enterprise JavaBeans) で作られることも多いです。しかし、EJB はかなり複雑なため、本書では扱いません (興味のある方は、EJB 3.0 に対応した資料にあたってください)。いずれにしても、Model をウェブアプリから切り離すことで、別のアプリケーションで再利用することができるようになります。