

# フレームワークを利用した Web アプリケーション開発

原 祐一<sup>A)</sup>、○若松 進<sup>A)</sup>

<sup>A)</sup>名古屋大学全学技術センター 工学系技術支援室 情報通信技術系

## 概要

これまで、多くの Web アプリケーションを開発してきたが、今後効率の良い開発を行っていく上でフレームワークの利用が必須となってくる。今回、Java 言語で利用される代表的なフレームワークである Spring Framework を利用したアプリケーション（会議室予約システム）を開発したので紹介する。

また、現在運用中の工学研究科シラバスシステムの後継システムを開発中であるが、開発が間に合えば合わせて紹介する。

## 1 アプリケーション開発

安定した運用を確保(システム障害は、システム更新時だけ)するため、アプリケーション開発にあたり、開発・動作確認等は全てローカルサーバ (Windows パソコン) 上で行い、完成後に公開サーバ上で運用することにした(図 1.参照)。

使用した開発環境は以下の通りである。

### 開発用

- ・使用 OS : Windows Vista SP2.0
- ・使用言語 : Java(JDK-1.6.18)
- ・サーブレットコンテナ : Tomcat 6.0.20
- ・開発環境 : Eclipse-3.4.1
- ・データベース : Mysql-5.1.41

### 公開用

- ・使用 OS : CentOS 5.4
- ・使用言語 : Java(JDK-1.6.18)
- ・サーブレットコンテナ : Tomcat 6.0.20
- ・データベース : Mysql-5.1.41

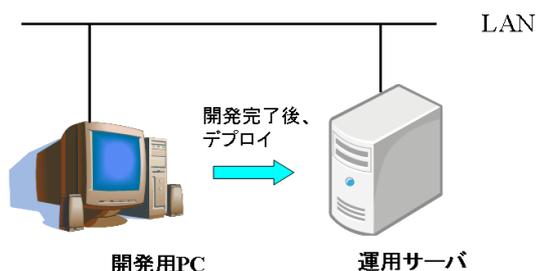


図 1. 開発環境

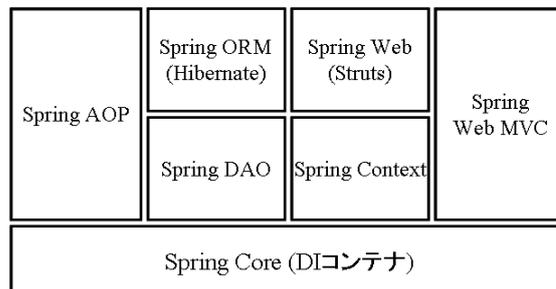


図 2. Spring Framework のモジュール構成

## 2 フレームワーク

アプリケーション開発は、DI(Dependency Injection)を実現する「DI コンテナ」機能を中核として、トランザクション管理、O/R(Object-Relation)マッピングツール、MVC(Model-View-Controller)フレームワークなど多くの機能を提供している Spring Framework<sup>[1]</sup>を利用することにした。

この Spring Framework は、DI コンテナ機能を提供する「Spring Core」を中核として全部で7つのモジュールから構成されており(図2参照)、今回の開発には Core, ORM, DAO モジュールを利用した。

### 2.1 Spring フレームワークのメリット

Spring フレームワークを利用する最大のメリットは、「必要なインスタンスの準備を Spring が行う」ことにある。Spring フレームワークを利用すると、必要なインスタンスを準備する時にその準備を Spring が行ってくれるため、インスタンスの準備に関するコードを記述する必要がなくなる。そして、設定ファイルにインスタンス同士の組み合わせ方を定義し、インスタンス同士をコードの修正なしに組み替え可能な状態にしておくことで、クラス間の依存関係が弱まり、変更に強く、コンポーネントの再利用性が高い柔軟なアプリケーション作成が可能となる。

### 2.2 SpringMVC 処理フロー

Spring フレームワークはフロントコントロールパターンを採用しているため、画面遷移が多い Web アプリケーションを開発するのに向いている<sup>[2]</sup>。

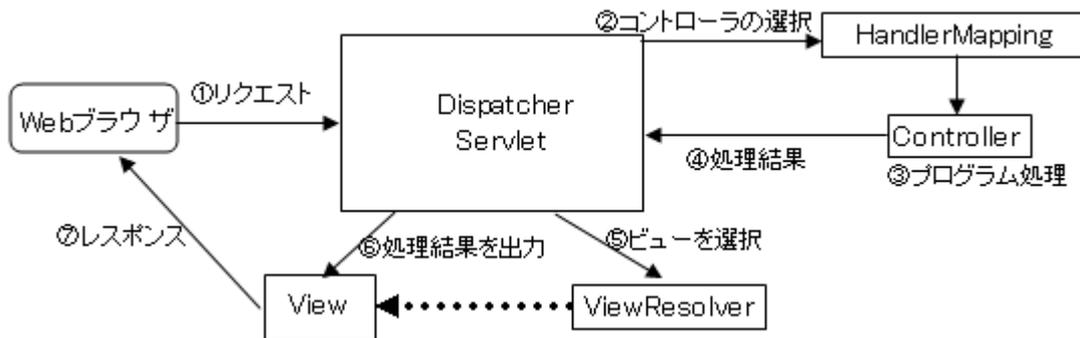


図3. SpringMVC 処理フロー

### 2.3 データベース操作 (Spring JDBC 抽象フレームワーク)

データベースを操作する方法として、java が標準で用意している JDBC が存在する。しかし、JDBC を直接利用したプログラミングには、

- Connection、Statement、ResultSet インスタンスの管理
- 煩雑な例外処理と SQLException の扱い
- Connection インスタンスの取得

という問題点がある。

Spring JDBC 抽象フレームワークは上記問題点をうまく隠しつつ、JDBC の良いところをそのまま残してくれるところに利用のメリットがある。

### 2.3.1 設定ファイル

実際に、Spring JDBC 抽象フレームワークを利用するには、Spring 設定ファイル(applicationContext.xml)へのデータベース接続に利用するデータソースの設定を記述する必要がある。

表 1. applicationContext.xml の設定

「org.springframework.jdbc.datasource.DriverManagerDataSource」クラス	
設定名	説明
driverClassName	JDBC におけるドライバクラスの FQCN
url	JDBC を利用するための接続 URL
username	データベースのアクセスユーザ名
password	データベースのパスワード

表 2. applicationContext.xml ソースコード

```
<beans>
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName">
    <value>com.mysql.jdbc.Driver</value>
  </property>
  <property name="url">
    <value>jdbc:mysql://localhost/hoge</value>
  </property>
  <property name="username">
    <value>springuser</value>
  </property>
  <property name="password">
    <value>springpassword</value>
  </property>
</bean>
```

※ 「hoge」はデータベース名

### 2.3.2 データベースへのアクセス

テーブルにアクセスするメソッドを Dao インターフェイスに定義し、その実装クラスとして DaoImpl クラスを作成する。

DaoImpl では、Dao インターフェイスを実装しつつ、「org.springframework.jdbc.core.support.JdbcDaoSupport」を継承することでデータベースにアクセスする。

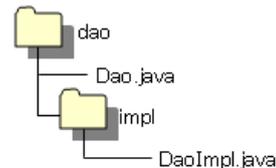


図 4. ファイル構成

※ JdbcDaoSupport クラスには、getJdbcTemplate()メソッドが実装されているため、JDBC を直接使用する場合よりもデータベースへのアクセスを簡略化することができる。

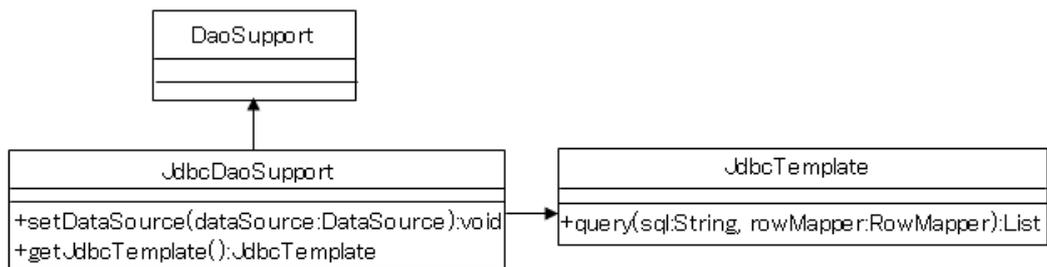


図 5. JdbcDaoSupport クラスと JdbcTemplate クラス

### 3 会議室予約システム

今回、工学研究科の化学系サーバのリプレースに伴い、旧システムで運用されていた会議室予約システム（10 年ほど前に開発されたシステム）を再開発することになった。開発する上で、SpringFramework を利用して作成した会議室予約システムを図 6.に示す。



図 6. 会議室予約システム

### 4 まとめ

今回の報告は、フレームワーク(Spring Framework)を利用したアプリケーション開発をする事を目的としており、フレームワークを利用しない開発方法に比べ、プログラムソース量や開発時間を短縮することができ、フレームワーク利用の有効性を知ることができた。

また当技術系において、今後の技術業務の一つとして Web アプリケーション開発を継続して行うことを予定しており今回報告したフレームワークを利用していく。

なお本報告作成時において、システム設計中の「工学研究科シラバスシステム」については技術研究会において紹介する予定である。

### 参考文献

- [1] 河村嘉之、他、“実戦 Spring Framework J2EE 開発を変える DI コンテナのすべて”,日経 BP 社
- [2] 村山雅彦、他、“Spring による Web アプリケーションスーパーサンプル“,SoftBank Creative