

軽量 Java による Web アプリケーションの MDA ツール

佐野 篤史† 紫合 治†

東京電機大学 情報環境学部†

1 はじめに

近年, Webアプリケーション開発において, フレームワークを駆使することにより, 開発工程を減少し, 開発効率を高めることができる軽量 Java[1]が注目されている. また, モデルから自動生成による開発を行うMDA (Model Driven Architecture) の実現が研究されている. 本論文では軽量JavaによるWebアプリケーションをMDAによって開発する方式とツールについて報告する.

2 従来方式

2.1 軽量 Java

今回述べる軽量Javaはプレゼンテーション層にJSF, ビジネスロジック層にSpring, インテグレーション層にHibernateのフレームワークを用いる. JSFはJSR127で仕様策定され標準化された, UIコンポーネントフレームワークである. Springは依存性を外部から操作するDI (Dependency Injection)機能, オブジェクト群に対して機能挿入するAOP (Aspect Oriented Programming)機能のあるフレームワークである. HibernateはオブジェクトとRDB (Relational Database)の透過的な変換をするO/Rマッピングのフレームワークである.

2.2 AndromDA(アンドロメダ)

AndromDA[2]はオープンソースのMDAツールで, モデルとなるクラス図を入力することによって, Struts, Spring, Hibernateのフレームワークを用いたWebアプリケーションを自動生成する.

MDA tool for Web application using Light Weight Java

† Atsushi Sano and Osamu Shigo, School of Information Environment, Tokyo Denki University

3 AndromDAの問題点

AndromDAは短時間でWebアプリケーションを開発できるが, 生成されたアプリケーションは現実的なものではなく, プレゼンテーション層の修正や独自の機能追加等が不可欠となる. その場合, 2つの問題がある.

(1) ビジネスロジック層が複雑

AndromDAで生成したビジネスロジック層は図1のようにクラス数が多く複雑で, 機能の追加に多くのクラスを参照, 修正する必要がある.

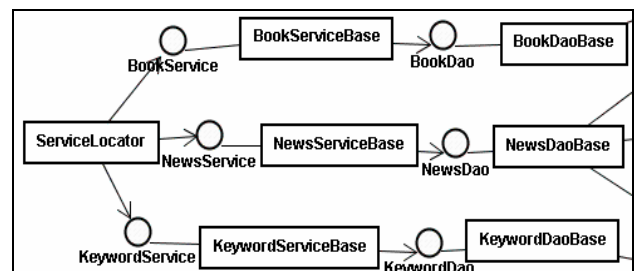


図 1. AndromDAで生成されたビジネスロジック層のクラス図

(2) パラメータ受け渡しの引数が多い

AndromDAで生成したインタフェースメソッドの引数は, データベースの列名と型により定まるので, データベースの設計によっては引数が非常に多くなり, 読みづらく, 修正が困難になる. 表1に自動生成されたインタフェースメソッドの一部を示す.

表 1. AndromDAで生成されたコードの例

```

public void create(String book_id, String isbn, String title, String company, String year, String proprietor, String author, Long id);
    
```

4 軽量 Java による MDA ツール

4.1 プレゼンテーション層の入れ替え

AndroMDA の問題点を解決するため、プレゼンテーション層を Struts から JSF に入れ替える方式を提案する。それに応じてビジネスロジック層も修正する。これによって、図 2 のようにビジネスロジック層のクラス数が減少し、DTO(Data Transfer Object)を用いることにより、表 2 のように引数の数をデータベースの設計に依存しないようにできる。

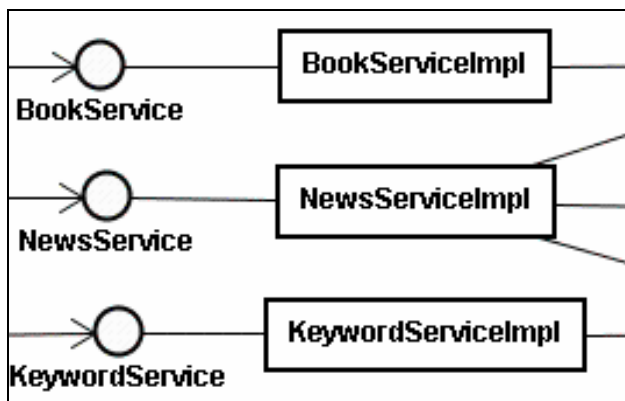


図 2. 変換後のビジネスロジック層のクラス図

表 2. 変換後のコードの例

```
public void create(Book book);
```

さらに、プレゼンテーション層の Struts を JSF に置き換えることで、JSF の特徴であるセッターインジェクションの利用により、ビジネスロジック層へのアクセスが外部ファイルから指定可能になり、ServiceLocator の必要性がなくなる。

4.2 ソース変換ツール

4.1 の方式を実現するため、AndroMDA で生成したソースを自動変換するツールを開発した。まず、AndroMDA により Web アプリケーションを自動生成し、生成されたプロジェクトのフォルダパスを変換ツールに入力する。変換ツールは Hibernate の設定ファイルを解析し、軽量 Java を利用したプロジェクトに変換する。変換されたプロジェクトは以下の特徴をもち、3 で述べた AndroMDA の問題点を解決している。

- ・ インテグレーション層はそのまま使用

- ・ ビジネスロジック層を 4.1 のように DTO 利用の JSF 向きに変換
- ・ プレゼンテーション層、JSP ファイルの生成
- ・ Spring, JSF, Web アプリケーション設定ファイルの生成

5 適用例

ソース変換ツールの有効性を確かめるため、本システムを応用して書籍情報管理システムの開発を行った。システムは、書籍情報・キーワード・記事情報を、データベースに登録・表示・編集・削除する Web アプリケーションである。まず、モデルとなるクラス図を作成し、AndroMDA で自動生成したプロジェクトを変換ツールで変換した。次に、書籍情報を検索する機能を AndroMDA によって生成されたプロジェクトと、変換ツール適用後のプロジェクトに追加し、追加にかかるコストを比較し、変換ツールの有効性を確認した。

表 3. 変更コストの比較

	変換前	変換後
修正箇所	5	3
追加行数	42	17

6 おわりに

本論文では、Web アプリケーションの MDA ツールである AndroMDA により生成された Web アプリケーションの問題点を分析し、解決するための変換ツールを開発した。また、変換ツールによって、機能追加が容易になることが確認できた。

今後の課題として、ツールの利用性の向上とより現実的なシステムへの適用によるツールの有効性の検証などがある。

参考文献

[1] 岡本隆史, 吉田栄嗣, 金子崇之, 権藤夏男, 「Light Weight Java」, 毎日コミュニケーションズ
[2] WINGSプロジェクト 佐藤 治夫, “AndroMDA でMDAの世界を体験する”,
<http://codezine.jp/a/article.aspx?aid=132>
<http://codezine.jp/a/article.aspx?aid=133>