

Apache Struts の設定削減機能の試作評価

倉持 和彦[†], 熊井 秀憲[‡], 五十嵐 政志[†], 米田 貴雄[†]

三菱電機株式会社情報技術総合研究所[†] 三菱電機株式会社神戸製作所[‡]

1. はじめに

近年、オープンソースソフトウェアが業務システムに適用されるようになってきている。中でも Apache Struts(以下 Struts とする)は Java 言語による Web 業務システムを構築する際のデファクトスタンダードとして基幹系業務システムのような大規模システムにも利用されるようになってきている。しかし、Struts を大規模システムに適用すると設定ファイルが肥大化するという問題点がある。本論文では、Struts の設定ファイルの記述量を削減する機能を試作し、その有効性について評価した結果について報告する。

2. Struts 設定ファイルの肥大化

Struts 設定ファイルには、struts-config.xml と validation.xml が存在する。struts-config.xml は、Struts を構成する Action や Form、ActionForward を定義するとともにその関係を設定するファイルである。また、validation.xml は Form の入力精査のルールを記載するファイルである。

図 1 に業務アプリケーションの画面例を示す。この画面では、ひとつの Form 内に入力フィールドが 2 つ、ボタンが 4 つ配置されており、各ボタンの入力精査ルールが異なるものとする。

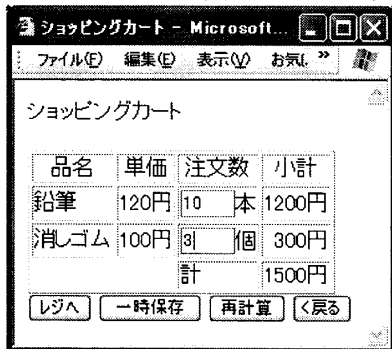


図 1 画面例

各ボタンの入力精査ルールが異なる場合、Struts では、EventDispatchAction と

MappingDispatchAction を組み合わせて利用することになる。この場合、struts-config.xml にアクションの定義がフォームに対してひとつ (EventDispatchAction 用)、ボタンひとつに対してひとつ (MappingDispatchAction 用) 必要になる。Validator.xml に定義する入力精査のルールとフォーム定義についてもボタンごとに必要になる。図 1 の画面を構成するために必要な設定ファイルの平均サイズを表 1 に示す。

表 1 設定ファイルサイズ

種類	定義数	平均行数	合計
Action 定義	5	5	25
Form 定義	4	1	4
Validator 定義	4	6	24

このように、業務システムのような複雑な画面になると Struts 設定ファイルが肥大化することとなる。たとえば、5KStep 程度のアプリケーションで Struts-config.xml が 400 行に達することもある。

3. Struts 設定削減機能

Struts 設定削減機能は Struts 設定ファイルの肥大化の原因となっている Action 定義、Form 定義および Validator 定義の削減を主目的とし、Struts 設定情報を設定するアノテーションとアノテーションを解釈する Struts のプラグインと ActionMapping クラスの拡張クラスから構成される。Struts 設定削減機能は以下の 4 機能からなる。

- Action 名や Form 名の自動命名規約
- Action、Form、JSP の自動マッピング
- アノテーションによる Action 定義と Validation 定義
- 物理パスによるとび先指定機能

3.1. 自動命名規約

特定のパッケージ配下に存在するアクションクラスやフォームクラスに対して自動でアクション名やフォーム名を命名する。たとえば、アクションクラス名が jp.co.xxx.cart.CartAction の場合、アクション名として、/cart/Cart と命

[†]Evaluation of a function to reduce items of configuration files of Apache Struts

[†]Mitsubishi Electric Corporation, Information Technology R&D Center

[‡]Mitsubishi Electric Corporation, Kobe Works

名する。また、フォームクラス名が `jp.co.xxx.cart.CartForm` の場合、`cart.CartForm` と命名する。このようにクラス名から自動で論理名を命名する機能である。

3.2. 自動マッピング

特定のアクションに対して対応するフォームおよびアクションからのとび先となる JSP のマッピングを自動で行う機能である。たとえば、`/cart/Cart` アクションに対して `cart.CartForm` フォームが対応するとともに、“success”のとび先を `/cart/cart.jsp` とする。これにより、アプリケーション開発者が `struts-config.xml` に設定をしなくても自動的に設定が行われることになる。

3.3. アノテーション

自動マッピング機能でマッピングしたフォーム以外を利用したい場合などに、フォームの対応やスコープなどを Action のクラスもしくは、`MappingDispatchAction` のメソッドに指定する。また、Validator 設定情報をフォームクラスにアノテーションで定義する機能である。

3.4. 物理パス指定機能

`struts-config.xml` でフォワード定義を行う代わりにアクションクラスのメソッドで物理パスを指定することにより直接飛び先を指定できる機能である。

4. 評価

Struts の設定削減機能の有効性を確認するため、サンプルアプリケーションを作成することによりコード量の評価を実施した。

4.1. サンプルアプリケーション

今回作成したサンプルアプリケーションは、懇親会を管理する Web アプリケーションである。アプリケーションのユーザは、懇親会の幹事と参加者である。幹事ユーザは、懇親会の開催情報を登録でき、参加者ユーザは、それに対し参加登録を行うことができる。アプリケーションの特徴は以下のとおりである。

- 画面数が 10、ボタン数が 15
- 業務アプリケーションを想定し、`EventDispatchAction` と `MappingDispatchAction` を利用して作成
- 画面単位で Form を作成

4.2. サンプルアプリケーションによる評価

4.1 で説明したサンプルアプリケーションを、Struts 設定削減機能を利用した場合と利用していない場合の 2 パターンで作成した。アプリケーションの作成コード量とその比率は表 2 に示すとおりとなった。結果として以下が示された。

- アノテーションの増加により Java コードが

4.2%増加した。

- 設定ファイルを 60.8%削減することができた。

表 2 アプリケーション作成コード比

種類	削減機能非利用	削減機能利用	削減率 [%]
Java	1692	1764	-4.26
JSP	480	418	12.92
設定ファイル	342	134	60.82

Java ソースコード増加の原因となったアノテーションの数についてカウントしてみると、表 3 に示すように 63 行となった。

表 3 アノテーション数

アノテーション	数
@StrutsAction	30
@Dispatch	33

1 KL あたりに換算すると約 35.7 行のコードが増加することが予想される。

次に、設定ファイルのライン数について詳細を表 4 に示す。

表 4 設定ファイルライン数

ファイル名	非利用	利用
/WEB-INF/web.xml	65	65
/WEB-INF/struts-config.xml	230	22
/ibatis/conf/SqlMapConfig.xml	12	12
/ibatis/dao/SqlMap-Template.xml	35	35

その結果、Struts 設定削減機能非利用の場合、設定ファイルの 67%を `struts-config.xml` が占めているが、本機能を利用することにより、16%まで削減することができた。設定ファイルの削減量は今回ターゲットとした `struts-config.xml` が占めており、`struts-config.xml` だけで見ると、その削減率は 90.4%に達している。アノテーションによるコード増加量をすべて設定にカウントしたとしても設定ファイルの削減率は 64.0%におよび、本機能は設定ファイル削減に有効であることが示された。

5. まとめ

本稿では、Struts 設定情報削減機能はサンプルアプリケーションによる評価において有効であるという結果が得られたことを述べた。今後は、`Validation.xml` についても同様の評価を行うとともに、本機能の実システムへの適用・評価を行う。

参考文献

- [1] Ted Husted ほか、Struts インアクション、ISBN4-7973-2344-2
- [2] Seasar プロジェクト、S2struts
<http://s2struts.seasar.org/ja/>