

細粒度マークアップに基づく JSP コーディング検査器の適用実験

吉田勝人, 森川和明, 並河英二, 中元秀明[†]

株式会社野村総合研究所[†]

1. はじめに

企業情報システムの開発では品質向上という観点から、コーディング時に守るべき標準化ルールを定め、全プログラムがこれに準拠したものになるようチェックを行っている。近年は中国などへのオフショア開発が増加しており、オフショアからの納品物に対する品質のチェックが特に重要になっている。こうしたチェックをより正確により少ない費用で実施するために、柔軟にルールを定義でき、そのルールに沿って繰り返しチェックが可能な検査器が必要である。これまでは、ビジネスロジック層の Java プログラムに対してはこのような検査が行われてきたが、プレゼンテーション層の JSP に対しては検査器そのものが存在していなかったため、目視検証によるチェックに頼らざるを得なかった。

そこで筆者らは名古屋大学と共同で JSP が標準化ルールに準拠しているかどうかをチェックする検査器の開発を行った。

この検査器の詳細については参考文献[1]に譲るとし、本稿では、開発した検査器を実際のプロジェクトに適用して有用性を評価する実証実験について述べる。

2. 企業情報システムの開発の現状

2.1 開発の流れ

本稿の内容に関係する部分を抽出して図 1 に示す。

- ① 企業（発注側）は、プログラムが守るべき標準化ルールを定め、開発要件とともにオフショア（受注側）に提供する。
- ② オフショアは、開発要件を基に設計、開発、テストを実施する。このとき標準化ルールに準拠するようにコーディングを行う。納品前にプログラムが標準化ルールに準拠しているか検査を行う。

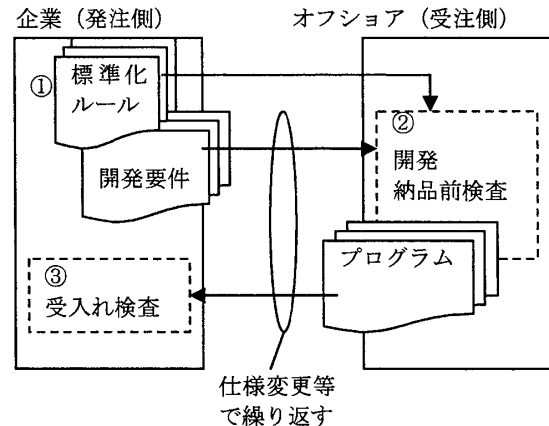


図 1. 開発の流れ

- ③ 企業は、プログラムを受け取り、受入れ検査を行う。この中で、プログラムが標準化ルールに準拠しているかどうかのチェックも行う。

2.2 課題

オフショア開発では、国内の開発と比較して距離も言語も隔たりがあるため、企業から見てオフショア側の活動はブラックボックス化してしまう傾向にある。また開発時・運用時とも、仕様変更や機能追加により何回もプログラム修正と納品が繰り返され、そのたびに納品検査が発生している。そのため企業にとっては、品質を向上させるために受入れ検査をいかにうまく実施するかがポイントとなる。

ビジネスロジック層の Java プログラムでは、機能テスト、標準化ルール準拠チェック、結果のレポートといった一連の受入れ作業を自動化しつつあるが、プレゼンテーション層の JSP ではそこまで行っていないというのが現状であり、特に標準化チェックの部分は目視検証に頼ってきた。

3. 検査器の機能

前述の課題を解決するために、名古屋大学の細粒度マークアップによるプログラム解析器 Sapid (参考文献[2][3])を利用して JSP コーディング検査器を開発した(図 2, 図 3)。ポイントは以下の二つである。

Evaluation of JSP Coding Checker based on A Generic Fine-grained Source Code Markup.

[†]Katsuhito YOSHIDA, Kazuaki MORIKAWA, Eiji NABIKA and Hideaki NAKAMOTO.

[†]Nomura Research Institute, Ltd.

- 日本語で記述していた標準化ルールを、より簡潔に表現でき、かつ検査器で扱えるように、XPath で宣言的に定義するようになった。
- 多くの開発者が XPath でルールを作成できるように、元となる JSP から自動的に XPath の候補を生成し、開発者はその候補を修正（基本的には不要な部分を削除）することにより目的の XPath ルールを作成するという記述支援機能を用意した。

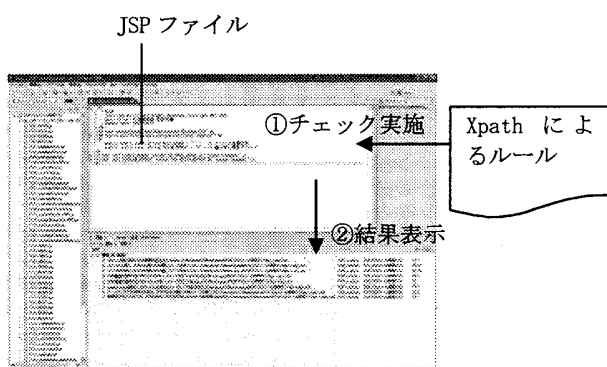


図 2. JSP コーディング検査器

```
<content>一つの JSP では一つの FORM を使用する</content>
<xpath>
//File[count(descendant::Element/STag[name="form"])]>1]
</xpath>
<condition>prohibit</condition>
```

図 3. XPath によるルールの例

4. 実証実験

検査器を社内の二つのプロジェクト（それぞれおよそ 50～60 個の JSP ファイルがある）に適用して、その有用性を評価する。評価の観点と内容は以下の三つである。

(1) 検査器によるチェックの正確性と生産性

過去に標準化ルールが正しく適用されているかどうかを目視で検証した事例があり、そのときのチェック結果と工数のデータが残っている。今回、このときと同じ JSP と標準化ルールに対して検査器で同様のチェックを行い、目視検証のときの結果と比較する。これにより、検査器でどれだけ標準化違反を抽出できるか、またそのときの工数はどれだけ削減できるかを評価する。

(2) 検査器によるルール化の網羅性

二つのプロジェクトの標準化ルールを今回の検査器を使用してルール化する。これにより、検査器のルール記述機能が標準化ルールの何パーセントを網羅する能力を備えているかを評価する。

(3) 検査器の習熟しやすさ

チェック作業自体は、検査器が Eclipse プラグインとして実装されているので、直感的に操作できすぐに使いこなせるようになる。問題は日本語ルールから検査器のルールを作成する部分である。今回、これまでルールを作成した経験のない開発者が実際にルール作成を行なうことにより、ルール化の作業がどれだけ習熟しやすいかを評価する。

現在、これらについて評価を実施中である。

5. おわりに

今回の検査器はプレゼンテーション層の標準化チェックに大きく貢献できると期待しており、社内の複数のプロジェクトに展開して活用していくことを想定している。

今後の課題は、チェック対象を JavaScript やスクリプトレットへ拡張していくことであり、JSP 部分のチェックの普及度合いを見ながら検討していきたいと考えている。

謝辞

本研究の一部は文部科学省リーディングプロジェクト基盤ソフトウェアの統合開発 e-Society 高信頼 WebWare の生成技術の助成による。

参考文献

- [1]金子伸幸, 今井敬吾, 山本晋一郎, 阿草清滋, “細粒度マークアップに基づくカスタマイズ可能なコーディング規約検査器”, 情報処理学会第 70 回全国大会講演論文集 (2008)
- [2]福安直樹, 山本晋一郎, 阿草清滋, “細粒度リポジトリに基づいた CASE ツール・プラットフォーム Sapid”, 情報処理学会論文誌 Vol. 39, No. 6, pp.1990-1998 (1998)
- [3]Sapid Home Page, <http://www.sapid.org/index-ja.html>