

Ruby on Rails における 自動訳文置換えの適応範囲の拡張

佐藤 駿介

松原 俊一

Martin J. Dürst

青山学院大学理工学部情報テクノロジー学科

1 はじめに

Web アプリケーションはその特性から様々な国や地域で利用されるため、国際化対応が重要である。ブラウザに表示する文章の翻訳はもちろんのこと、数値の整形や言語特有の規則など、国際化を開発者が独自に行うのは負担が大きい。開発者の負担を軽減させるために、国際化処理を自動化することは効果的な手段である [1]。

Web アプリケーションフレームワーク Ruby on Rails (Rails) は、標準ライブラリで国際化に対応できる。訳文置換えをする `translate` メソッドなどにより、開発者の国際化の負担を軽減させている。しかし、`translate` メソッドを用いた既存の訳文置換え手法は余分な記述が多くなり、可読性の低下を招く。そのため、苅宿らは Ruby on Rails の `translate` メソッドなしでの自動訳文置換えを実装した [2]。

本研究ではさらに、テンプレート内の訳文箇所での出力用タグの記述を削減した。これにより国際化対応前と同じ記述で国際化が可能となる。国際化対応のためのコード量を削減し、高い可読性を保つことに成功した。

2 Rails の標準の訳文置換え手法

`translate` メソッドを用いた訳文置換えでは、ハッシュのキーと値がそれぞれ訳文キーと訳文に対応したロケール辞書を使用する。訳文キーを `translate` メソッドに渡し、訳文を出力する。`translate` メソッドはテンプレート内で利用しやすいように `t` というエイリアスが用意されている。ソースコード 1 とソースコード 2 は、英語から日本語へのロケール辞書とそれを用いた訳文置換えの例である。本論文ではこの手法を `t` 明示手法と呼ぶ。

訳文置換えは訳文キーをテンプレートの出力用タグ (`<%= コード %>`) で表し、さらにその箇所をメソッド呼出しで明示する必要がある。Rails ではタグ間のコー

ドをテンプレートエンジン Erubis で解釈し、処理している。この出力用タグと `translate` メソッドにより可読性が低下する。

ソースコード 1: ロケール辞書

```
1 ja:
2   plan: '計画'
3   comment: 'コメント %{count} 件'
```

ソースコード 2: 訳文置換え例

```
1 <%= t "plan" %>
2 #=> 計画
3
4 <%= t "comment", count: 3 %>
5 #=> コメント 3件
```

3 先行研究

苅宿ら [2] は、`translate` メソッド無しの訳文置換えを実装した。本論文ではこれを `t` 削減手法と呼ぶ。表 1 は「`plan`」から「計画」への訳文置換えの、`t` 明示手法と `t` 削減手法との比較である。

表 1: `t` 明示手法と `t` 削減手法の記述の比較

手法	記述
<code>t</code> 明示手法	<code><%= t "plan" %></code>
<code>t</code> 削減手法	<code><%= "plan" %></code>

`translate` メソッドによる指定を省略することで記述量が減り、可読性が向上した。しかし、出力用タグで表す必要があることは変わらないため余分な記述は依然として多い。記述量を削減させることで可読性の向上が見込める。

4 自動訳文置換えの適応範囲拡張

前節で述べた問題を解決するため、テンプレート処理の際に HTML タグ間に文字列が存在する場合 `translate` メソッドを呼び出す手法を提案、実装した。

Extending the Range of Automatic Translatable Text Replacement in Ruby on Rails

Shunsuke Sato, Shunichi Matsubara, and Martin J. Dürst
Department of Integrated Information Technology, College of
Science and Engineering, Aoyama Gakuin University
5-10-1 Fuchinobe, Chuo-ku, Sagami-hara, 252-5258, Japan
duerst@it.aoyama.ac.jp

本論文ではこれをタグ削減手法と呼ぶ。表2はt明示手法とt削減手法、タグ削減手法との訳文置換えの比較である。t明示手法では出力用タグで表し、translateメソッドを記述しなければならなかったが、タグ削減手法では訳文キー(多くの場合、元の言語の単語や文節)を直接記述することが可能となる。

表2: 「plan」の「計画」への置換え

手法	記述
t明示手法	<%= t "plan" %>
t削減手法	<%= "plan" %>
タグ削減手法	plan

4.1 変数付き訳文置換え

ソースコード2の4行目のように、translateメソッドによる訳文置換えでは訳文キー以外に変数を渡しても良い。国際化において可算名詞の有無や名詞の性の変化などに対応するための機能である。

タグ削減手法ではこの機能に対応している。ソースコード3はタグ削減手法における変数付き訳文置換えの例である。

ソースコード3: 変数付き訳文置換えの例

```
1 "comment", count: 3
2 #=> コメント 3件
```

訳文キーをダブルクォートで囲み、変数とその値をコロンで区切るよう記述を限定させることで、本機能を実装した。

4.2 訳文置換えが不要な場合の処理

タグ削減手法では、訳文キーが存在しない場合は文字列をそのまま表示する。しかし、訳文キーは存在するが置き換えずに表示させたい場合もある。そのような場合のために、文字列の末尾に.as.isを接続することで自動訳文置換えを行わないよう実装した。ソースコード4はその使用例である。

ソースコード4: .as.isの使用例

```
1 Ruby.as_is
2 #=> Ruby
```

また、既存のプロジェクトにタグ削減手法に対応させる際、テンプレートによって自動訳文置換えを行うのは不適切な場合がある。テンプレートごとに自動訳文置換えを適応するか選択可能にする必要がある。本研究では、テンプレートの1行目にmanualと記述することで自動訳文置換えを行わないよう実装した。

さらに、ErubisはRails以外でも用いられることを考慮し、config/application.rbで設定を有効にすることで本実装の機能を使用可能とした。

4.3 実装の詳細

Erubisではテンプレートを処理する際、正規表現によって開始タグと終了タグの間の文字列を解釈し処理している。それ以外の部分はテキストとして出力する。

タグ削減手法ではテキスト部分をさらに正規表現でマッチングし、HTMLタグ間の文字列を抽出している。これにより自動訳文置換えや変数の受渡し、訳文置換えを行わない場合の処理などに対応させた。

4.4 評価

タグ削減手法により訳文置換え時の記述が削減された。どの程度減らせるか評価するため、オープンソースのWebアプリケーションRedmine¹を調べた。その結果、タグ削減手法を利用すると500か所以上で合計5000文字以上を削減することが可能であった。

さらに訳文置換えをする箇所としない箇所の、少ない方のみに対応することで国際化が可能となった。これはRailsが提唱している「設定より規約」の一例と捉えられる。

5 まとめと今後の課題

本研究ではRailsのテンプレート内の国際化方法を改善することに成功した。Erubisのテンプレート処理を変更することで、Railsにおいて多く利用されるtranslateメソッドと出力用タグの記述量を削減した。これにより、新規Webアプリケーションを作成する際、高い可読性を保ちながら簡単に国際化対応を行うことが可能となった。

今後の課題として、正規表現マッチングの効率化が考えられる。テンプレートのタグをマッチングした後さらにHTMLタグでマッチングを行うのは非効率である。同時にマッチングすることが可能になれば、処理の無駄を減らすことができる。

参考文献

- [1] Martin J. Dürst. Localization in the UNET++ Application Framework. *8th International Unicode Conference, Hong Kong, April 1996.*
- [2] 苅宿航太, 松原俊一, Martin J. Dürst. 訳文の自動置き換えによるWebアプリケーションの国際化方法の改善. 情報処理学会全国大会講演論文集, 第78巻, pp. 1.563–1.564, 2016.

¹<http://www.redmine.org/>