

# ルールセットの類似性を用いた CSS リファクタリング手法の提案

山本 優三<sup>†</sup> 福安 直樹<sup>†</sup> 満田 成紀<sup>†</sup>  
吉田 敦<sup>‡</sup> 松延 拓生<sup>†</sup> 鯨坂 恒夫<sup>†</sup>

本研究では、CSS 記述に含まれる類似したルールセットの共通部分を一つのルールセットにまとめるリファクタリング手法を提案する。オブジェクト指向言語に適用される「Template Method の形成」リファクタリングを応用することにより、CSS 記述の保守性の向上を目指す。また、CSS 記述の中から類似したルールセットを見つけるための手法として、二つのルールセット間の類似度を測定する手法を定義する。この値が高いルールセットの組合せほど、提案するリファクタリングの対象となり得る。

## CSS Refactoring Method using Similarity of Rulesets

YUZO YAMAMOTO<sup>†</sup> NAOKI FUKUYASU<sup>†</sup> NARUKI MITSUDA<sup>†</sup>  
ATSUSHI YOSHIDA<sup>‡</sup> TAKUO MATSUNOBE<sup>†</sup> TSUNEO AJISAKA<sup>†</sup>

In this paper, we propose a refactoring method to integrate similar rulesets in CSS descriptions. We try to improve the maintainability of CSS descriptions by applying the refactoring “Form Template Method” which is usually applied to the object-oriented language. We define a technique for measuring the similarity between two rulesets of CSS descriptions. The pair of rulesets with a higher similarity becomes a target of our refactoring method.

### 1. はじめに

Web 設計において、視覚的構造を定義する技術として CSS と呼ばれるものが存在するが、その開発手法は非効率的な部分が多く、未成熟である。CSS では、HTML 構造の中でスタイルを適用する対象にあたるセレクタと、そのスタイル情報を設定するためのプロパティと属性値の組で構成される宣言の集合とを一つのルールセットとして、複数のルールセットを重ね合わせて目的のデザインを実現させる。その際、似たようなデザインを実現させたい場合、安易にコピー&ペーストしてしまい、複数の類似した箇所を作ってしまうがちである。それらは結果として、保守における作業効率の低下や、CSS ファイルのデータサイズを余分に大きくしてしまう恐れがある。

そこで本研究では、それらの問題を解消するために、CSS 記述におけるリファクタリング<sup>1)</sup>手法を提案し、ルールセットの類似度を定義する。そして、それを用いて実際の CSS 記述に対してリファクタリングを実施する。

### 2. CSS リファクタリング

JavaやC++などのオブジェクト指向言語では、パターン化されたアルゴリズムを抽出し、無駄な重複を排除させるためのリファクタリングとして、「Template Methodの形成」<sup>1)</sup>と呼ばれるものが存在する。このリファクタリングの手順をCSS記述に応用した手法を提案する。

#### 2.1 提案手法

まず初めに、それぞれのCSS中で指定方法が複数あるプロパティ<sup>\*1</sup>が存在した場合、それらのプロパティに対して、個別に指定する形式に変換し、表記法の統一を行う。その上で統合の対象となる類似したルールセットの組を見つける。ルールセットの類似性については、2.2 節で議論する。次に、その対象ルールセットの中で共通ではなかった宣言を個別に分割し、宣言し直す。その際、互いの対象ルールセットのセレクタ名が統一可能であった場合、セレクタ名を「名前の変更」<sup>1)</sup>により統一する。これは、セレクタ名の変更対象となったルールセットを含むCSSを参照しているHTMLも影響を受けるため、HTML自身にも変更作業が必要となる。そして最後に、もともとのCSSファイルの先頭で、import宣言の追記によって新たに作成するCSSファイルを読み込ませると共に、統合可能部分のルール

<sup>†</sup> 和歌山大学システム工学部  
Faculty of Systems Engineering, Wakayama University

<sup>‡</sup> 和歌山大学システム情報学センター  
Center for Information Science, Wakayama University

\*1 margin{ top, right, bottom, left }に対して margin など

表 1 2つの CSS で定義されたルールセットの類似度  
Table 1 Similarity of CSS rulesets.

	table.attendance	tr.odd	tr.even	table.attendance td	td.title, td.date
table.report	1.00	0	0	0	0
tr.odd	0	1.50	0.50	0	0.17
tr.even	0	0.50	1.50	0	0.17
table.report td	0	0	0	1.00	0
td.title, td.step	0	0.17	0.17	0	1.50

セットを新規 CSS ファイルへと引き上げる。

## 2.2 ルールセットの類似度

ルールセットの類似度は、任意の二つのルールセットを比較して、同じ値を持つプロパティがいくつ指定されており、それが二つのルールセットで宣言されているプロパティのどの程度を占めているかを示す値である。ルールセット  $r_1$  と  $r_2$  の類似度  $S(r_1, r_2)$  を、 $P(r)$  をルールセット  $r$  で宣言されているプロパティ数、 $C(r_1, r_2)$  をルールセット  $r_1, r_2$  に共通のプロパティ数、 $D(r_1, r_2)$  をルールセット  $r_1, r_2$  の編集距離(= $(P(r_1) - C(r_1, r_2)) + (P(r_2) - C(r_1, r_2))$ )として、

$$S(r_1, r_2) = \frac{C(r_1, r_2)}{D(r_1, r_2) + 1}$$

と定義する。ただし、プロパティによっては、まとめて指定する方法が用意されているため、このようなプロパティについては個別に指定する形式に変換した上で、重み\*<sup>2</sup>を付けて評価する。

## 3. 適用実験と考察

2節で述べた CSS リファクタリングを、実際の CSS 記述に適用した。類似度を求めた結果の一部を表 1 に示す。二つの異なる CSS で定義されているルールセットのセクタを縦軸と横軸に記述し、それぞれについて、それらの間の類似度を求めている。表の類似度の高い組み合わせから、双方の **tr.odd** と **tr.even** の組が、また、**td.title** および、**td.step** と **td.date** の組が、それぞれ類似した表現で出力されていることがわかる。今回はその中で、**td.step** と **td.date** のセクタ名を **td.count** へと統一し、新たな CSS ファイル(C.css)へと引き上げた例を図 1 に示す。

今回挙げた CSS ファイルの例に関して表 1 を見てみると、主対角線の類似度が、それ以外の部分に比べて高くなっているが、対応するセクタが対象とする HTML の構造も似ており、二つの CSS は同じ表現を別々のルール

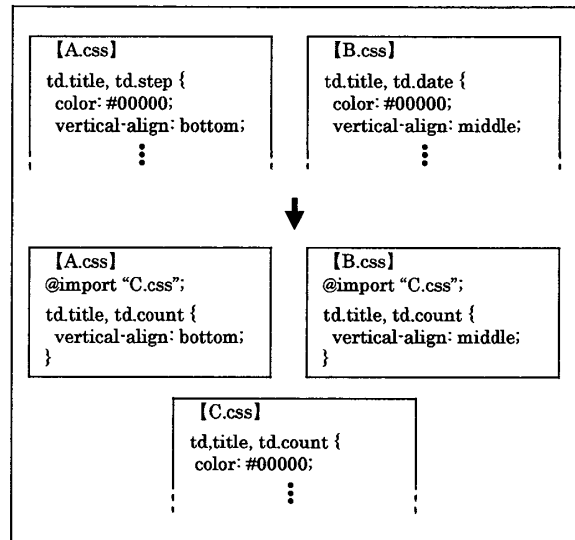


図 1 CSS 記述へのリファクタリングの適用例  
Fig 1 Result of our refactoring.

セットで実現していることが予想される。そこで、ルールセットの 1 対 1 の関係で類似度を議論する他に、グループ間での類似度の議論も必要ではないかと考えられる。また、個別指定に変換可能なプロパティにおいて、今回は重みを仮に 0.5 としたが、より多くの事例を用いて更なる検討が必要であると考えている。これらは共に今後の課題である。

## 4. おわりに

CSS の保守性を向上させるためには、重ね合わせの特徴を活かし、無駄な重複は避けるべきである。そこで、本研究では、そのような重複を取り除くためのリファクタリング手法を提案し、リファクタリングの対象となる類似したルールセットを見つけるために、類似度を定義し、実際の CSS 記述に適用した。CSS 開発の際に本研究の手法を用いることで、CSS 記述の整合性が保たれ、開発効率の向上につながると考えている。

## 参考文献

- 1) Fowler, M.: *Refactoring: Improving the Design of Existing Code*, Addison-Wesley(1999).

\*2 margin, padding などの個別の指定について 0.5