

JavaScript を用いた Web アプリケーションにおける CSS の拡張に関する研究

2006MI001 阿部 百合名

指導教員 野呂 昌満

1 はじめに

多くの Web アプリケーション開発技術があり、中でも RIA(Rich Internet Application) が注目されている。RIA は一般的に、文書構造を HTML、見栄えを CSS、振舞いを JavaScript と、分離して記述することが推奨されている [2]。分離することで、アクセシビリティの確保やデザインとプログラミングの開発の分業ができる。しかし、現実には、JavaScript ファイル内で CSS を制御する記述が含まれており、見栄えが CSS だけではなく JavaScript にも記述されており、振舞いと見栄えの記述が混在している。

本研究は、見栄えと振舞いの記述を分離することを目的とし、HTML 要素の状態を考慮した CSS の拡張方法を提案する。JavaScript には振舞いのみを、CSS には見栄えを記述し、見栄えと振舞いを分離することで、HTML、CSS、JavaScript を役割ごとに分離できる。また、Web アプリケーションのデザイナーは HTML 要素の状態と状態遷移を意識しながら HTML、CSS を作成し、プログラマはその状態遷移に対してロジックを記述することで、互いに HTML 要素の状態を意識するのみで分業できる。なお、以下では提案する CSS の記述を「拡張 CSS」と呼ぶ。また、本研究では HTML、JavaScript の分離を容易にするために、jQuery[1] を用いた Web アプリケーションを対象に議論する。

2 Web アプリケーションの問題点

2.1 見栄えと振舞いの混在

CSS は、見栄えと文書構造が混在した HTML の保守性の悪さを改善するために、見栄えを独立して記述する枠組として提案された。しかし、RIA では、CSS の制御が JavaScript の中で記述されており、JavaScript と CSS の記述が独立していない。振舞いと見栄えの記述が混在している具体例を図 1 に示す。下線部分は見栄えに関する JavaScript の記述である。

```
function judgeAnswer() {  
  var total=$('#total>div.number:first').text();  
  var answer=$('#answer>div.number:first').text();  
  if (total==answer){  
    $('#img#judge').attr('src','correct.png').show();  
  }else{  
    $('#img#judge').attr('src','mistake.PNG').show();  
  }  
  $('#img#judge').attr();  
}
```

図 1 見栄えと振舞いの混在例

図 1 は数当てゲームのプログラムの一部である。下線部は、id 属性が judge の img タグに対して、if 文で場合分けして src 属性の値を切り替え、表示画像を変更している。このように JavaScript で見栄えに関する記述が行なわれている。

2.2 CSS による見栄えの記述

色盲などの障害を持つ人も含め、誰もが使いやすく情報を共有できることをアクセシビリティと呼ぶ。振舞いと見栄えが JavaScript ファイル内で混在すると、アクセシビリティの確保が難しいという問題が生じる。一般に、別の CSS ファイルを切り替えることでアクセシビリティを確保できる。振舞いと見栄えが混在することで、従来の CSS を切り替えてもロジックに制御される見栄えは切り替えられず、ロジックと依存関係にあるので見栄えを切り替える方法も複雑になり、アクセシビリティの確保が難しい。

2.3 Web アプリケーションの開発プロセス

Web アプリケーションの開発プロセスは一般的に、デザインとプログラミングの 2 つで構成されている [4]。図 2 に従来の Web アプリケーションの構成を示す。

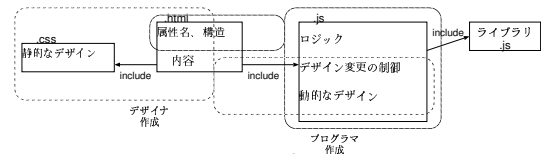


図 2 現在の Web アプリケーションの構成

デザインはデザイナーが画面構成をもとに HTML、CSS を記述し、プログラミングはプログラマが JavaScript でロジックを実装する。しかし実際には、見栄えの制御が JavaScript に含まれるので、デザイナーとプログラマの双方が JavaScript を編集する必要があり、デザインとプログラミングの分業が困難という問題がある。

3 HTML 要素の状態を考慮した CSS の拡張

3.1 HTML 要素の状態を考慮した CSS

JavaScript で見栄えを変更するタイミングを複数の Web アプリケーションで確認したところ、HTML の属性値が変化したときや、HTML 要素自身または親要素にイベントが起きたときなど、HTML 要素の状態が変化したときに変更されていた。そこで、HTML 要素の状態に着目して見栄えと振舞いを分離する方法として、HTML 要素の状態名を記述でき、見栄えに関する JavaScript コードを記述できるように拡張した CSS を提案する。また、JavaScript で状態遷移を明示的に記述するための state メソッドも提案する。拡張 CSS を採用した Web アプリケーションの構成を図 3 に示す。

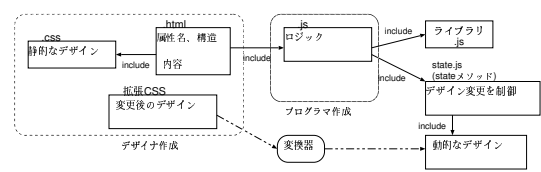


図 3 拡張 CSS を適用した Web アプリケーションの構成
なお、本研究では、拡張 CSS を JavaScript ファイル

に変換して Web アプリケーションに使用する。

3.2 拡張 CSS

拡張 CSS では、従来の CSS のセレクタの代わりに、セレクタと状態名の組を記述する。また、各状態が持つ見栄えを表すプロパティでは、従来の CSS のプロパティのほかに、JavaScript を用いた値の計算式、状態遷移による他の HTML 要素の見栄えを変化させる JavaScript コードを記述する。図 4 に拡張 CSS の具体例を示す。

```
( #answer, focus ){
  display : block;
  top:({ $( "#total" ).offset().top + $( "#cor_mis" ).height() });
  left:({ $( "#total" ).offset().left + $( "#total" ).width() });
}
( .judge, correct ){
  display : none;
  ({$("#correct").show();})
}
```

図 4 拡張 CSS の記述例

図 4 での (#answer, focus) のように、セレクタと状態名の組は、セレクタと状態名をカンマで区切り括弧で囲い記述する。中括弧内にプロパティ名とプロパティの値を記述し、各状態における見栄えの変更を記述する。拡張 CSS のプロパティの値には、従来どおりの定数と、JavaScript による計算式の 2 種類がある。計算式は、図 4 の top のように、({, }) 内に JavaScript コードを記述する。また、図 4 の id 属性 correct の JavaScript コードのように、HTML 要素の状態遷移により他の要素の見栄えを変化させる場合、プロパティ名は書かず、({, }) 内に JavaScript コードを記述する。

3.3 state メソッド

一方、ロジックには CSS の制御の記述はせず、状態遷移のみを記述する。state メソッドは、セレクタ名および状態名の一致する拡張 CSS を検索し実行する jQuery オブジェクトのメソッドである。また、JavaScript 内には見栄えを記述せず、state(“状態名”)と記述する。

4 実装と評価

4.1 実装

本研究の実現可能性を確かめるために、jQuery の拡張と変換器の実装を行なった。なお、jQuery は、HTML、JavaScript の分離を実現する上で広く利用されており、本研究で採用した。

JavaScript ファイルに変換された拡張 CSS の仕様を決めた。拡張 CSS から JavaScript の変換器は、Perl で 40 行程度で実装できた。state メソッドは jQuery のプラグイン形式で 30 行程度で実装できた。

4.2 評価

4.1 節で実装したものを使い、実際に Web アプリケーションに適用して、評価を行なった。

4.2.1 拡張 CSS によるアクセシビリティの確保

評価で用いた Web アプリケーションは、合計値から数字の組合せを当てるゲームで、JavaScript で約 80 行程度である。セレクタと状態名の組は 8 個、state メソッドの呼び出しは 8 箇所である。

アクセシビリティを確保するために、Web アプリケーションに見栄えを変更できる仕組みを追加した。追加方法として、ページ表示後にセレクトボックスにより別の拡張 CSS に動的に切り替える方法を採用した。選択した数字の色を変更するように拡張し、実際に色が変更でき

ることを確認した。

4.2.2 開発プロセスの分離

評価は 4 つの Web アプリケーションで行なった。本章では、メニューにマウスが重なるとサブメニューを表示する機能を持つ、JavaScript ファイルは 10 行程度のもを例としてあげる。評価は、従来の Web アプリケーションと拡張 CSS を適用した Web アプリケーションの見栄えと振舞いを修正し、双方の修正箇所、および修正箇所の開発プロセスを比較して行なった。

拡張 CSS を採用することで、見栄えと振舞いの影響範囲を局所化でき、分業できる。サブメニューの表示位置を変更する際、従来の Web アプリケーションでは、JavaScript 内で CSS のプロパティの値を計算し見栄えを変更していたので、ロジックを書き換える必要があった。しかし、拡張 CSS を適用した Web アプリケーションでは、拡張 CSS を修正することで表示位置を変更できる。同様に、ロジックのみを変更する場合、見栄えに影響せずにロジックを修正できる。しかし、見栄えと振舞いのどちらで記述すればいいか判断が困難な事例があった。メニュー外にマウスを移動したことでサブメニューを非表示にする際、ロジックで hide メソッドを用いて制御するか、状態遷移として拡張 CSS に非表示を記述するか、判断が曖昧となる箇所があり、問題として残った。

また、実装した形式では、拡張 CSS は 1 つのセレクタと状態の組で見栄えを定義する。しかし、違うセレクタの同じ状態の見栄えを定義するには同じプロパティの定義を記述する必要があり、保守性の観点から複数のセレクタを記述できる記法の導入が望ましいと考える。

4.3 関連研究

アスペクト指向技術を用いて、JavaScript における横断的関心事を分離する研究 [3] があるが、JavaScript 内での振舞いと見栄えの分離は考慮されていない。Unobtrusive JavaScript では、jQuery を用いることで HTML、JavaScript は分離できるが、JavaScript、CSS を分離できない。本研究では、考慮されていない振舞いと見栄えの分離を目的としている。

5 おわりに

本研究では、HTML 要素の状態を考慮した CSS の提案を行なった。また、実装を行ない、有効性について評価した。今後の課題として、4.2.2 節で述べた問題の検討があげられる。

参考文献

- [1] J. Resig and the jQuery Team, “jQuery,” <http://jquery.com/>, 2009.
- [2] S. Willison, “Unobtrusive JavaScript with jQuery,” <http://simonwillison.net/static/2008/xtexh/>, 2008.
- [3] 久保淳人, 他, “AOJS:アスペクトを完全分離記述可能な JavaScript アスペクト指向プログラミング・フレームワーク,” ソフトウェア工学の基礎 XV, pp.145-154, 2008.
- [4] 福田浩章, 山本喜一, “RIA におけるデザイナーと開発者の完全分業を実現するフレームワーク,” 情報システム学会誌, vol.4, No.2, pp.28-43, 2009.