

ウェブサイトデザインのためのフレームワーク組み合わせ手法の提案

島藤 大誉[†] 満田 成紀[‡] 福安 直樹[‡] 松延 拓生[‡] 鯨坂 恒夫[‡]
 和歌山大学大学院システム工学研究科[†] 和歌山大学システム工学部[‡]

概要:近年, HTML5 の普及に伴い, Bootstrap や Foundation などのウェブサイトデザイン用フレームワークを利用することが多くなっている. しかし, フレームワークによって提供されている機能や表現は異なり, サイト開発者が求めるデザインを単一のフレームワークで実現することは難しい. 本研究では, 複数のフレームワークを組み合わせることを目的とし, サイト開発者が異なるフレームワークの機能や表現を選択して利用することを可能とする手法を提案する. この手法によって, 複数のフレームワークを利用する際のコンフリクトを解消し, ウェブサイトの軽量化, CSS の編集自由度の向上を図る.

1 はじめに

近年, HTML5 の普及に伴い, ウェブサイトデザイン用フレームワークを利用することが多くなっている. 特に Bootstrap[1]や Foundation[2]などのレスポンシブ CSS フレームワークの利用頻度は増加している. これらのフレームワークを利用するメリットとして, ウェブサイト制作に必要な枠組みが用意されていることから, ライブラリのように呼び出さずとも, 少ない工数で多様な機能とコンポーネントを整然と施せることが挙げられる[3]. しかし, フレームワークの利用が増加するほど, 似通ったデザインや動作のウェブサイトが蔓延し, オリジナリティが失われることに繋がる[4]. また, シンプルなページ・サイトを作る際には, フレームワークが無駄にデータ量を増やしてしまうというデメリットが挙げられる[5]. これらのデメリットを解決するために, 複数のウェブサイトデザイン用フレームワークの組み合わせを行い, ウェブサイトの軽量化, CSS の編集自由度の向上を図る.

2 組み合わせの問題点

既存のフレームワークを組み合わせる際に生じる問題点として以下が挙げられる.

- ・名前の衝突
フレームワーク毎に設定されている固有の class, id が被って衝突する. もしくは, 開発者が新規に設定した class, id とフレームワークの class, id が衝突する.
- ・無駄な容量
フレームワークのデメリットである大きなデータ量が組み合わせで増大し, 処理速度を下げる.

・解説コスト

共同開発で他者がソースコードを解説する際に, どのフレームワークのどのコンポーネントを使っているのかが分かりにくくなる. しかしながら, 一つ一つコメントにそれらを記述するのは開発者の負担になる.

・デザイン骨格部の衝突

グリッドなどのサイトデザインの骨格を形成するコンポーネントは他のコンポーネントにデザインを継承する. そのため, フレームワークを組み合わせた際に各々のデザイン骨格が衝突し, デザインの崩壊が起きやすくなる.

・記述自由度低下

デザイン骨格部が多様化することにより, デザインの規則性やルールが複雑化する. 従って, 開発者が記述する自由度が低下し, ウェブサイト自体のクオリティの低下や, オリジナリティの喪失の恐れがある.

3 サービス提案

フレームワークの組み合わせに伴う問題点を解消するため, 制作するウェブサイトに合わせて, 開発者がオリジナルのフレームワークを作成できるサービスを提案する.

3.1 サービス概要

オリジナルフレームワーク作成サービスとして利用者は以下の手順を行う[図 1].

1. コンポーネントや機能ごとに細分化された要素から利用したいものを選択
2. class, id などを衝突しないように編集
3. 結合した要素やフレームワーク識別番号を記述したフレームワーク設計書とオリジナルフレームワークをダウンロード
4. 後で追加したい要素が発生した際に, フレームワーク識別番号からサービスで検索し, フレームワークを再構築 (1に戻る)

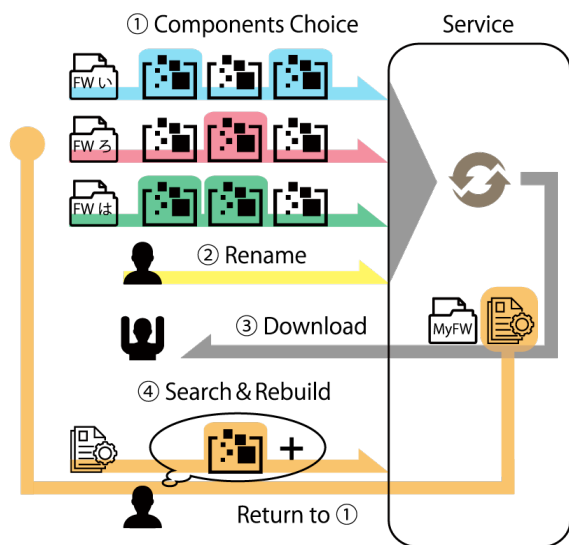


図 1：サービス概要図

3.2 システム概要

サービス内部のシステムとして以下の順番に処理を行う[図 2].

1. コンポーネントや機能ごとにフレームワークの内部を細分化
2. 利用者が選択した要素を抽出し、衝突がないか分析
3. 編集された class, idなどを反映
4. 利用者が選択した要素を結合し、オリジナルフレームワークを生成
5. 結合した要素やフレームワーク識別番号を記述したフレームワーク設計書を生成
6. データベースに生成されたものを記録
7. 追加要素の要求に応じてフレームワークを再構築 (2に戻る)

4 考察

提案するサービスにより、名前の衝突はリネーム処理で回避され、必要な要素だけでフレームワークを構成するため無駄な容量が発生することもない。また、設計書により、解説コストの問題も解消される。デザイン骨格部の衝突、記述自由度低下に関しては、組み合わせるフレームワークの中でどれか一つのフレームワークの条件や規則のみを採用するというルールを設けることで衝突を解消する。それに伴い、フレームワークのどの要素がデザインの骨格部に当たるのかを分析する必要がある。

また、組み合わせる際の、フレームワーク毎の相性の良し悪しを分析する必要がある。例えば、同じ間隔でグリッドを分割しているフレームワーク同士は組み合わせやすいが、そのフレームワークが固有に設定した要素などがある場

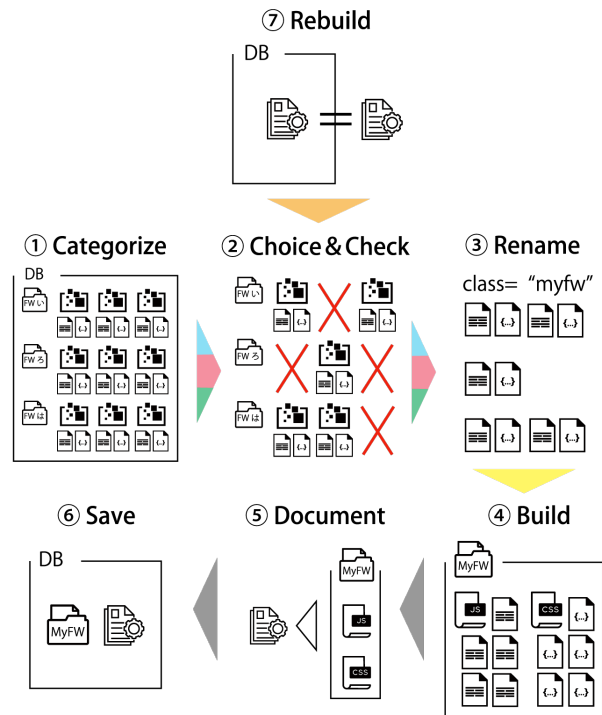


図 2：システム概要図

合、組み合わせの際に非常に大きな障害になることがある。

5 まとめ

本研究のサービス提案により、組み合わせる際の問題点だけでなく、フレームワークそのもののデメリットも解消する。また、開発者が組み合わせたオリジナルフレームワークによって、どこかで見たことのあるような、オリジナリティの喪失したウェブサイトの蔓延に歯止めをかける。ウェブサイト全体のデザインの多様化と処理の軽量化を図る。

参考文献

- [1] Bootstrap, “<http://getbootstrap.com/>”, 2017年1月13日参照.
- [2] Foundation, “<http://foundation.zurb.com/>”, 2017年1月13日参照.
- [3] Kouichi Yasaki, Hidenobu Ito, Kazuaki Nimura, Proposal of a Package Scheme of HTML5 Application by Optimizing Based on Data Characteristics, 2014.
- [4] 谷義人, 満田成紀, 鱒坂恒夫, XSLT を用いたモジュール性の高い WEB アプリケーションの開発手法とフレームワークの提案, 2004.
- [5] 今泉智博, 斎藤裕佑, 西山裕之, HTML5 のセクショニング要素自動付与システムの提案, 第 77 回全国大会講演論文集, 2015.