

WebアプリケーションのUI機能テスト のための HTML構造パターンの抽出手法

ITシステムプロジェクト

政策・メディア研究科 修士1年

笹本 将平

読んできた論文

- タイトル
 - WebアプリケーションのUI機能テストのためのHTML構造
パターンの抽出方法
- キーワード
 - Webアプリケーション, テスト, 自動化, UI, ツール
- 書いた人
 - <http://toshihirokamiya.com/index-j.html#basic-info>

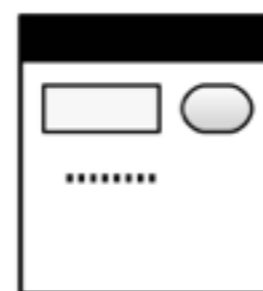
目次

- **研究背景**：なぜこの研究をしているか
- **提案手法**：どうすれば解決できそうか
- **評価**：つくってみてどうだったか
- **まとめ**
- **関連研究**：こういう研究もあります

研究背景

- UIの機能テストはホワイトボックステスト
- ex) WebのUIテスト自動化 - Seleniumを使ってみる
- UIに関するコードのリファクタリングによってテストが壊れやすい

こんなテストが書きたい



前提条件:
トップページが表示されている



操作:
検索ボックスに「とうがらし 玉ねぎ」と
入力し、エンターを押す



期待する動作:
トウガラシと玉ねぎを使った料理
のレシピのリストが表示される

図 2 テストケースの例

Fig. 2 A sample test case

UI機能テストの記述

- 特定の画面が表示されていることを調べる方法
 1. DOMツリーの特定の要素が特定のテキストや属性を持つかチェックする
 2. DOMツリー全体を準備しておいた正解のDOMツリーと比較する

UI機能テストの記述

- 1の問題点は、データの一部しかチェックされていない
 - ex) 余計なものが表示されているとかは別のテストが必要
- 2の問題点は、壊れやすいテストになる
- 1, 2共に一貫性のテストは記述できない
 - ex) 異なるキーワードで検索した結果ページが、内容は異なるが同じレイアウトに従っている

課題をまとめると

特定の画面が表示されていることを確認したい

- 一部しかチェックできない or 全部一致しない
といけない
- 一貫性のテストができない

提案手法

- Webアプリの複数のページのDOMツリーの共通部分をパターンとして表現し、実際に表示されているページがそのパターンに従っているか判定する手法を提案する
 1. DOMツリーからのパターンの半自動生成
 2. パターンを用いたテスト

前回の論文からの変更

- 前回

- HTML構造パターンを作成
- パターンをUIテストの記述に利用する手法を提案

- 今回

- HTML構造パターンのモデル拡張
- 実際のページからパターンを抽出する手法の提案

HTML構造パターンとは

- DOMツリーの要素
 - 要素節点(属性と値を持つ)
 - テキスト節点
- 新たに導入する要素
 - ワイルドカード節点(属性と値を持つ)
 - ワイルドテキスト節点
 - 繰り返し節点

一般的なDOMツリー

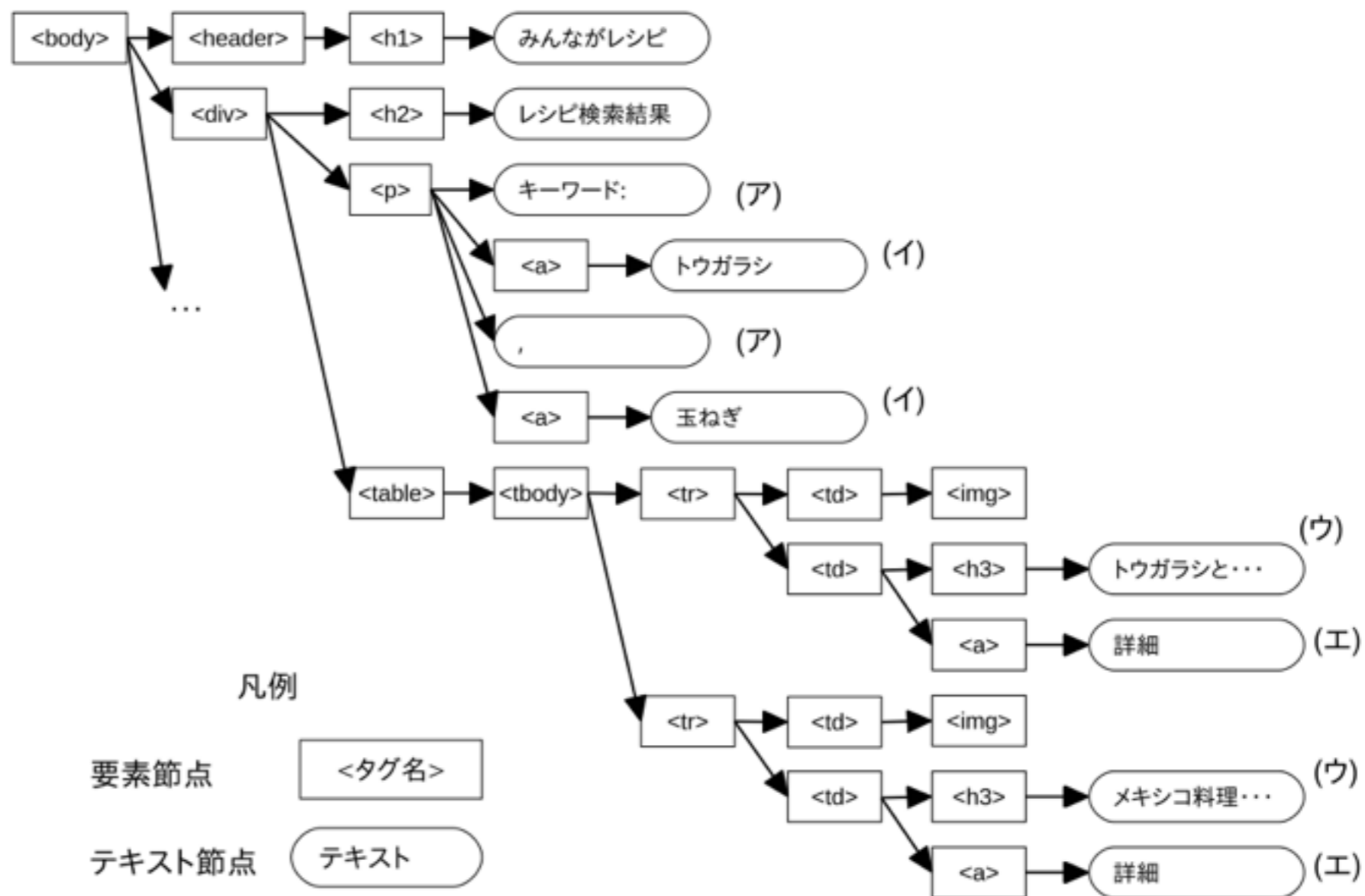


図 1 Web ページの例とその DOM ツリー

Fig. 1 A Web page and its DOM tree

HTML構造パターン

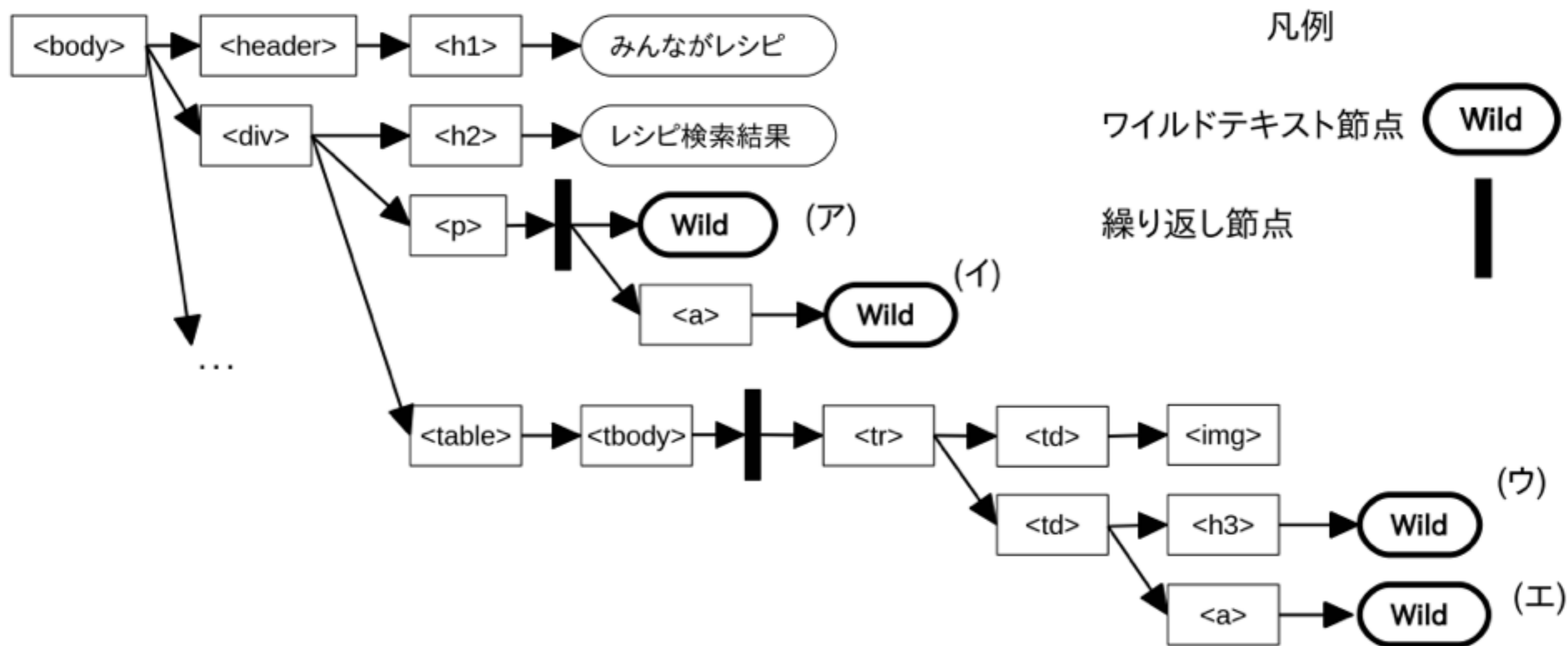


図 3 HTML 構造パターンの例

Fig. 3 A sample HTML structure pattern

HTML構造パターンの生成手法

2つ以上のDOMツリーを入力とする。

1. 入力となったDOMツリーをそれぞれ折りたたむ
 - この際にマージ可能な節点同士はマージされる
2. m を最初のDOMツリーに初期化する
3. 2つ目以降のDOMツリー(t)のそれぞれについて, m を m と t を強制的にマージしたものに更新する

DOMツリーの折りたたみ

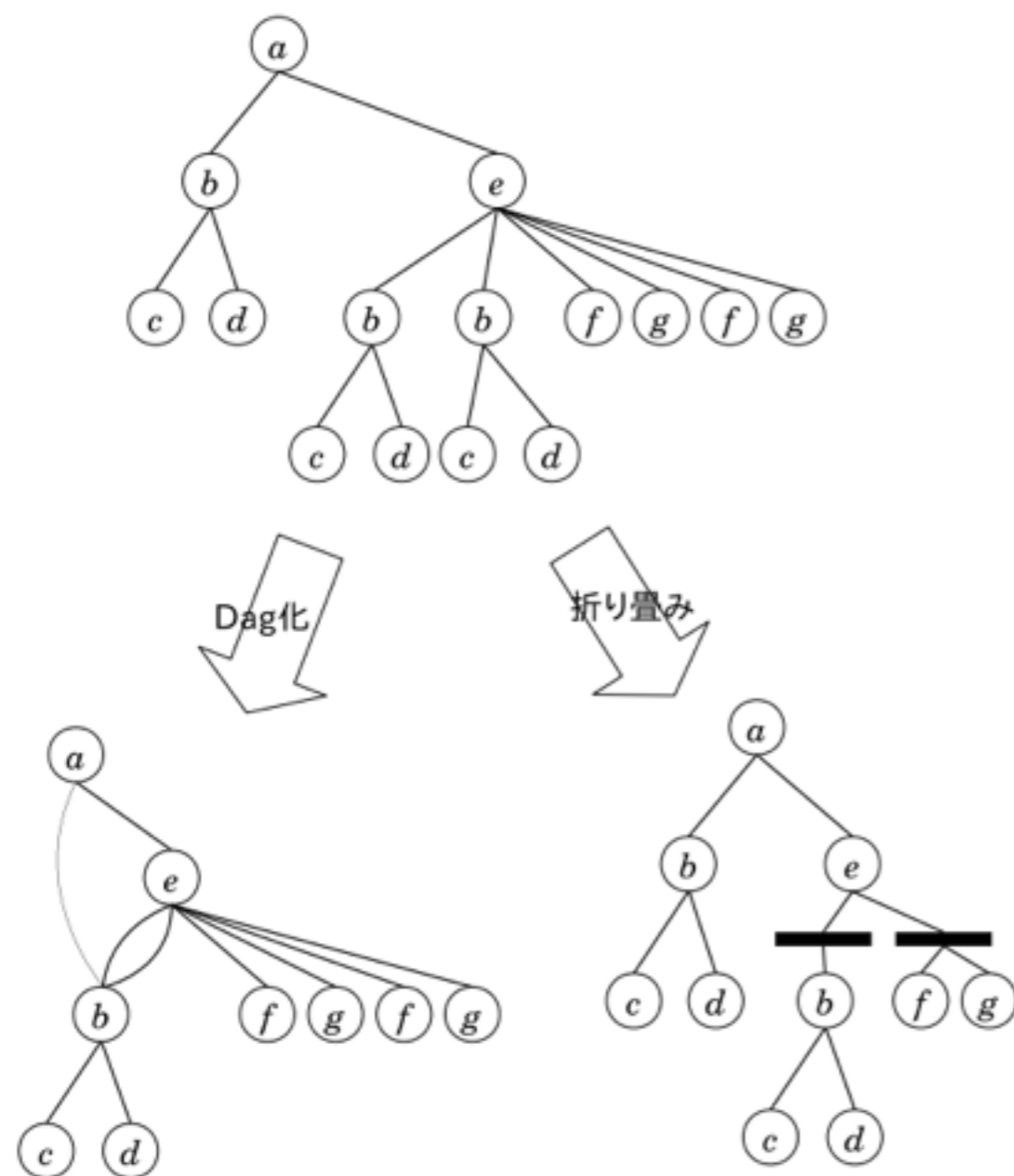


図 4 Dag 化と折り畳み

Fig. 4 Dagifying vs. folding

- 葉から根に向かって順に、すべての節点の子節点の繰り返しを、繰り返し節点に置き換える

Dag化との違い

- 子節点の並びの中から繰り返しを見つけて置き換える
- DOMツリーの異なる深さにある節点はまとめない

マージ可能性

- 繰り返し節点を子接点の並びに置き換えた並びの長さが同じ
- すべての対応する節点について、両節点等しい
- タグ名が等しい && 子節点の並びがマージ可能
- 両節点ともにテキスト節点

実験1: ページの分類

テストサイトはG, K, Rの3種類のページを持つ

1. G, K, Rのそれぞれについて複数のページのDOMツリーを取得
2. それらを2郡に分ける (seed, trial)
 - G-seed, G-trial, K-seed, K-trial, R-seed, R-trial
3. seed郡から、HTML構造パターンを生成する
 - G-pat, K-pat, R-pat
4. それぞれのpatをtrial郡に適用し、マッチするか試す
 - 表2

実験2: 無視される要素の割合

パターンがDOMツリーにマッチした際に

1. 元のDOMツリーのノード数と属性の値の数をカウントする
2. そのうち、いくつがワイルドカード節点やワイルドカード値にマッチしたのかを調べる

- 表3

考察

- ページの種類によって、利用されていない節点の割合のバラつきが大きい
- 生成されたパターンにワイルドカード節点が散見されたので、得手不得手がある可能性がある

評価結果

- 実験用Webアプリにおいて、ページの分類ができた
- 正解データからパターンを生成し、それとマッチするかどうか判定
- つまり、一貫性のテストができた

まとめ

- 文献[4]の手法（パターンのモデル）を拡張し、より多様なパターンに対してマッチングを行えるようにした
- WebアプリのUIテストを目的とした、パターンマッチングの評価方法について述べ、実験的評価を行った

提案手法により解決できること

- 一貫性のテストが可能になる
- ex) 異なる2つのキーワードで検索した結果画面が同じレイアウトに従っているか
- 壊れやすい問題は解決できないが、原因特定に役立つエラーメッセージを開発者に提供できる

関連研究

- DOM構造を解析して意味を抽出する
 - 文献[2], [3]
- 欠損を含む部分木のパターンの表現手法
 - 文献[1]
- 拡張元となった論文
 - 文献[4]