文脈を考慮したHTML4からHTML5への書換え支援に関する研究

2008MI102 河野 仁美　　2008MI178 大石 嗣也　　2008MI258 内村 健太
指導教員　蜂巣 吉成

1　はじめに

HTML5 は W3C などによって策定が行われ，普及が進んでいる [3]．HTML5 では文書構造を明確に表現する要素や input 要素の属性などが追加されている．HTML5 を使うことで，新しいユーザインタフェースが利用でき，Web アクセシビリティが向上するなどの利点がある．HTML5 の記述支援に Dreamweaver などの Web オーサリングツールが存在する [2]．ツールを使用して HTML5 の文書を新規に作成することができるが，既に作成した HTML4 の文書を書き換えた方が手間が少ない．ツールでは DOCTYPE 宣言を HTML5 に書き換えることはできるが，タグの書換えなどはできず，手作業で HTML4 から HTML5 に書き換えることになる．文書全体から書換え箇所を目視で確認し，編集するので時間と労力がかかり，誤りが混入する可能性も高くなる．これは書換え作業の自動化で回避が可能である．

書換えの自動化には，書換え箇所の識別が必要だが，単純にタグのみを識別する方法では不可能である．HTML4 の文書から書換え可能と識別するための情報を発見し，組み合わせる必要がある．書換えも書換え箇所の要素の単純な書換えだけでなく，他の要素も書き換える場合がある．フォームの複数の入力欄を 1 つの入力欄に書き換える場合は，要素の書換えと振舞いを同じにするために JavaScript による処理の追加も必要となる．

本研究では，書換え作業の自動化による HTML4 から HTML5 への書換え支援を行う．HTML4 の Web ページを調査した結果，書換え箇所の文脈が書換えに使用できると考え，文脈により識別方法を 4 通りに分類した．あるタグ A の文脈とは，タグ A が現れる箇所のタグ A の属性値およびタグ A の前後の要素やテキスト，コメントの組み合わせのことである．また，書換え方法を書換え後の記述方法によって，6 通りに分類した．さらに書換え方法と識別方法を組み合わせて細分化し，自動的に書換えを行う方法を提案する．

提案方法に基づいて，HTML4 から HTML5 への書換えツールを設計・実現し，評価を行う．評価には調査に使用した Web ページと調査に使用しなかった Web ページを用い，ツールにより適合率 90.8%，再現率 84.4% の高い精度で書換えを行うことができた．適切に書換えが出来なかった箇所については考察を行う．

2　HTML5 の要素の変更点と書換え対象

2.1　HTML5 での変更点

HTML5 での変更点を表 1 に示す [4]．HTML5 では，構造的に見やすい HTML 文書を書くことやブラウザでの新たなインタフェースの利用など，新規の機能が追加された．逆に一部の要素は使用例が少ない，CSS で扱うべきであるなどとして廃止された．また，ユーザがより使用しやすいように意味が変化した要素もある．

2.2　本研究での書換え対象

本研究では，HTML5 でアクセシビリティが向上した input 要素の属性や，文書構造を明確に表現できる header，footer 要素，HTML5 で廃止された要素を書換え対象とする (表 1)．これらの要素は id 属性や class 属性，タグが出現する前後のテキストなどの文脈を考慮することで，書換え箇所を識別可能である．

対象外とした要素は，そもそも HTML4 の文書中に書換え元となる記述が存在しないもの (audio など)，文書の意味を理解しなければ書換え箇所を識別できないものや (aside など)，意味が変わった要素である．意味が変わった要素は，HTML4 の場合と使用箇所は変わらないので，書き換える必要はない．

表 1　HTML4 から HTML5 の変更点と書換え対象

追加された要素	文書構造に関する要素	対象	header, footer
		対象外	article, aside, hgroup, nav, section
	テキストに関する要素	対象外	datalist, mark, time, rp, rt, ruby, wbr
	フォームに関する要素	対象外	keygen, meter, output, progress
	コンテンツに関する要素	対象外	audio, canvas, embed, vide, figure, source, figcaption
	ユーザの操作に関する要素	対象外	summary, command, details, menu
input 要素に追加された属性	タイプ属性値	対象	search, tel, url, email, date, month, time, number
		対象外	datetime, datetime-local, week, range, color
	属性	対象	required, min max, step
		対象外	multiple, placeholder, pattern, autocomplete, autofocus
廃止された要素		対象	acronym, basefont, big, center, dir, font, strike,tt
		対象外	frame, frameset, noframes, isindex, applet
意味が変わった要素		対象外	b, i, em, strong, small, hr

3　書換え支援方法の提案

本研究では文脈を考慮した書換え支援方法の提案を行う．実際に使われている Web ページについて，どのような文脈で書換え対象のタグが出現するかを調査して，識別方法を 4 通りに分類し，文脈として探索するノードの範囲を定めた．書換え方法は，書換え元の要素や CSS，JavaScript の追加記述などから 6 通りに分類した．

3.1　識別方法

HTML4 で書かれた Web ページ 50 件に対して，どのような文脈で書換え対象のタグが出現しているか調査し，次の 4 通りに分類した．

1. HTML4 のタグのみ
2. 属性値とテキスト，コメントの組み合わせ
3. 属性値とコメントの組み合わせ
4. それ以外の組み合わせ

1 に該当する要素は HTML5 で廃止された要素が該当する．廃止された要素は，そのタグが出現した文脈によらずに書き換えることができる．

2 は多くの要素が該当し，属性値，テキスト，コメントに特定の文字列が出現している文脈において書換えが可能である．各要素について HTML 文書を調査した結果から，識別に用いる文字列を書換え箇所として確定できる文字列と，書換え箇所の候補となる文字列に分類した．確定できる文字列は，その要素以外に出現する可能性が低い文字列であり，候補となる文字列は他の箇所に出現する可能性がある文字列である．例えば，input の type 属性値を email に書き換える場合は，属性値やテキストに「mail」や「メール」という文字列が出現している場合は書き換えを行い，「add」(address の略) は書換えの候補とする．文字列はまず，属性値を優先して調べ，その後テキストとコメントを調べる．これは属性値の方がタグ自身の情報をもっている可能性が高く，識別に適しているからである．

3 は header，footer 要素が該当し，HTML4 の文書で div 要素であった箇所から書き換える．テキストを用いない理由は，テキスト中に header や footer に関する文字列が出現する例がなかったからである．

4 は特殊な例であり，複数の要素を組み合わせる場合や，img 要素の alt 属性を用いる場合などがある．

2, 4 に該当する要素は，書換えの基準となるタグとして input 要素を用いる．年月日などは select 要素によるセレクトボックスで表現している場合があるので, input 要素の type 属性値を”date”, ”month”, ”time”, ”number”のどれかに書き換える場合は，select 要素も書換え箇所とする．

表 2, 表 3, 表 4 に識別に用いる文字列を示す．完全一致と書かれていない文字列は部分一致とする．※が付いている要素は，書換えを確定する文字列が複数の組み合わせで使用されている場合のみ書換えを行う．アルファベットで書かれている文字列はすべて大文字，小文字関係なく識別を行う．空白の欄は，識別に使用できる文字列がなかったものである．

表 2　識別に用いる属性値

識別に用いる属性値 / HTML5 の要素	書換えを確定する属性値	書換え候補に用いる属性値
header	header，hedda，ヘッダ	
footer	footer，futta，フッタ	
input type = ”search”	search，query，kensaku，situmon，検索，質問，ワード	key，word，kw，q(完全一致)
input type = ”tel”	tel，phone，denwa，keitai，fax，電話，携帯，ファックス	
input type = ”url”	url，http://，site，homepage，link，saito，ho-mupe-ji，rinku，サイト，ホームページ，リンク	
input type = ”email”	mail，me-ru，メール	add
input type = ”date”※	年，月，日	yy，mm，dd，year，month，day，y(完全一致)，m(完全一致)，d(完全一致)
input type = ”month”※	年，月	yy，mm，year，month，y(完全一致)，m(完全一致)
input type = ”time”※	time，hour，minute，時，分	hh，mm，h(完全一致)，m(完全一致)
input type = ”number”	count，nin，kazu，people，個 (完全一致)，数 (完全一致)，人 (完全一致)	yy，mm，dd，year，month，day，pass，y(完全一致)，m(完全一致)，d(完全一致)
input required		

表 3　識別に用いるテキスト

HTML5 の要素	書換えを確定するテキスト
header	
footer	
input type = ”search”	検索，ワード
input type = ”tel”	tel，phone，電話，携帯
input type = ”url”	url，サイト，リンク，ホームページ
input type = ”email”	mail，メール
input type = ”date”※	年，月，日
input type = ”month”※	年，月
input type = ”time”※	時，分
input type = ”number”	個，数，人，生年月日 (完全一致)，年，月，日
input required	必須, ∗, ※

表 4　識別に用いるコメント

HTML5 の要素	書換えを確定するコメント
header	header，ヘッダ
footer	footer，フッタ
input type = ”search”	検索，ワード
input type = ”tel”	tel，phone，電話，携帯
input type = ”url”	url，サイト，リンク，ホームページ
input type = ”email”	mail，メール
input type = ”date”※	年，月，日
input type = ”month”※	年，月
input type = ”time”※	時，分
input type = ”number”	個，数，人，生年月日 (完全一致)，年，月，日
input required	必須, ∗, ※

3.2　識別に用いるノードの探索範囲

識別に使用する文字列を抽出するためのノードの探索範囲を広くすると誤った書換えを行うことがあり，狭くすると文字列を抽出できないことがあるので，探索範囲を適切に定める必要がある．また，より正確な識別を行うために，探索するノードの優先順位も定める必要もある．そこで，識別に必要な情報があるノードの箇所を，要素ごとに HTML4 で書かれた Web ページを 50 件ずつ調査した．その結果, 基準となるタグの兄弟, 親, 親の兄弟，親の親までに識別に使用する要素があるのは，平均で 45 件あった．親の親より広い範囲を指定しても書換えに関する文字列の位置にばらつきがあったので，上記の範囲を探索範囲とする．ノードの優先順位は，書き換えるそれぞれの要素で異なり，兄弟, 親, 親の兄弟, 親の親の中で件数が多かった順とする．なお，input type = ”date”, ”month”, ”time”に関しては識別に用いる文字列を複数取得する必要がある．1 つ目の識別に用いる文字列を取得してから次の文字列を取得するまでのノードの探索範囲を弟の弟とする．

3.3　書換え方法

HTML4 で記述された Web ページを要素ごとに 50 件調査し，書換え方法を 6 通りに分類した．

1. タグのみの書換え
2. タグと属性値の書換えと CSS の追加
3. input 要素の type 属性値の書換え
4. input 要素の type 属性値の書換えと JavaScript の追加
5. select 要素から input 要素への書換え
6. select 要素から input 要素への書換えと JavaScript の追加

1 に該当する要素は，HTML5 で廃止された要素の中でタグのみを書き換えれば良いものと header，footer 要素である．書換えを行っても，HTML4 で使用していた要素を使ったときと同じ表示がブラウザで可能である．

2 は，廃止された要素の中で，CSS を使用して表現できるものである．書換えを行うと書換え前と同じ表示ができなくなるが，外部ファイルの中の CSS の設定をユーザが直接入力することで，同じ表示はできなくてもユーザが望む表示にすることができる．

3 は，input 要素で書かれており，type 属性値を書き換えるものである．書換えを行うことで HTML4 よりも視覚的にわかりやすい入力フォームを作成することができる．

4 は，input 要素で書かれている複数の入力欄を 1 つに書き換えるものである．例えば，HTML4 で年と月と日で入力が分かれている箇所を input type = ”date”を使って 1 つの入力欄に置き換えて，JavaScript による処理を追加する．

5 は，select 要素で書かれているセレクトボックスを input 要素の type 属性値で書き換えるものである．

6 は，4 と同様だが，複数の select 要素で書かれている入力欄を 1 つの input 要素に書き換えるものである．

3.4 入力欄が複数ある場合の書換え方法

3.3 節で分類した 4 と 6 の書換えはタグや属性を単純に 1 対 1 で書き換えるものではなく，複数の箇所の記述を 1 つに書き換え，さらに JavaScript による処理を追加する書換えである．HTML5 では日付や時間の入力欄として，input 要素の type 属性値に date や time などが新たに追加され，これらを使用することで，アクセシビリティの向上したユーザインタフェースを利用できる．HTML4 での日付や時間の入力欄の記述は，input 要素や select 要素を複数用いているものが大半である．よって，書換えを行うには複数の要素の記述を 1 つに書き換える必要がある．しかし，これだけではサーバへ送信するフォームデータが書換え前とは異なるので，JavaScript による処理を追加して書換えを行う．入力された値を JavaScript を呼び出して複数に分割し，それを input 要素の type 属性値の hidden に追加することで値を受け渡す．例えば input type = ”date”は，yyyy-mm-dd の形なので，yyyy と mm と dd の 3 つに分割を行い，値を受け渡す．

4 書換え支援ツールの設計と実現

本研究ではツールの実現にあたり，オープンソースの HTML パーサである jsoup[1] を用いた．jsoup は，タグなどを抽出する際，セレクタ構文を用いて容易に HTML を操作できることから，本研究で採用した．言語は Java を使用し，ツールは 3044 行になった．CSS の外部ファイルは 23 行，JavaScript の外部ファイルは 70 行になった．HTML4 から HTML5 への変換ツールの処理手順を図 1 に示す．

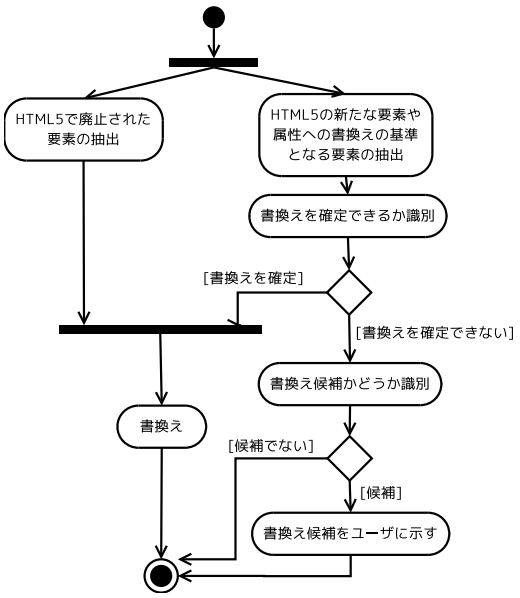

図 1　変換ツールの処理手順

5 評価と考察

変換ツールが正しく書換えを行えるかを評価した．

5.1 評価方法

評価に用いる対象は，本研究で書換え対象とする要素ごとに対し各 50 件ずつとする．実際に使用されている件数が少なかった要素は，各 20 件とした．これらが記述された HTML4 の文書に対してツールを適用し，正しく書き換えられたかを，適合率と再現率を用いて評価する．適合率は結果の正確性を表し，再現率は結果の網羅性を表す．適合率と再現率の算出式を次に示す．

$$\text{適合率} = \frac{\text{ツールが書き換えた適切な件数}}{\text{ツールが書き換えた件数}}$$

$$\text{再現率} = \frac{\text{ツールが書き換えた適切な件数}}{\text{書換え可能箇所の総数}}$$

書換え可能箇所とは，HTML4 から HTML5 の要素や属性に書き換えられるところを HTML4 の文書のソースコードから目視で確認した箇所を表す．ツールが書き換えた適切な件数とは，書換え可能箇所のうちのツールが書き換えた件数を表す．評価対象となる HTML 文書は，3 章の調査に使用した HTML 文書と，それ以外の HTML 文書の 2 種類とする．ツールは 3 章での調査に基づいて設計・実現しているので，調査に使用した HTML 文書に対しては高い適合率，再現率になることが予想される．調査に使用しなかった HTML 文書に対しても実験を行うことで提案方法の妥当性を検証する．

5.2 評価結果

調査に使用した HTML 文書と，それ以外の HTML 文書に対してツールを使用した結果の適合率と再現率を

表 5，表 6 に示す．なお，廃止された要素は一部の要素を表に載せているが，廃止された要素は，調査に使用した HTML 文書とそれ以外の HTML 文書共に，すべて書き換えることができた．header，footer に関しては識別に使用する属性値やコメントがある文書を評価に使用したので，再現率が 100% となっている．全体の適合率は 90.8%，再現率は 84.4% となった．

表 5　調査に使用した HTML 文書

HTML5 の要素	適合率		再現率	
header	50/60	83.3%	50/50	100.0%
footer	49/53	92.5%	50/50	100.0%
input type = ”search”	44/49	89.8%	44/50	88.0%
input type = ”tel”	48/52	92.3%	48/50	96.0%
input type = ”url”	47/52	90.4%	47/50	94.0%
input type = ”email”	45/50	90.0%	45/50	90.0%
input type = ”date”	34/35	97.1%	34/50	68.0%
input type = ”month”	10/10	100.0%	10/20	50.0%
input type = ”time”	27/27	100.0%	27/50	54.0%
input type = ”number”	45/56	80.4%	45/50	90.0%
input required	46/57	80.7%	46/50	92.0%
廃止された要素	適合率		再現率	
acronym	20/20	100.0%	20/20	100.0%
dir	20/20	100.0%	20/20	100.0%
strike	20/20	100.0%	20/20	100.0%

表 6　調査に使用していない HTML 文書

HTML5 の要素	適合率		再現率	
header	50/62	80.6%	50/50	100.0%
footer	49/53	92.5%	49/50	98.0%
input type = ”search”	31/33	93.9%	31/50	62.0%
input type = ”tel”	43/45	96.6%	43/50	86.0%
input type = ”url”	48/65	73.8%	48/50	96.0%
input type = ”email”	45/49	91.8%	45/50	90.0%
input type = ”date”	31/33	93.9%	31/50	62.0%
input type = ”month”	8/8	100.0%	8/20	40.0%
input type = ”time”	24/24	100.0%	24/50	48.0%
input type = ”number”	25/39	64.1%	25/50	50.0%
input required	28/30	93.3%	28/50	56.0%
廃止された要素	適合率		再現率	
acronym	20/20	100.0%	20/20	100.0%
dir	20/20	100.0%	20/20	100.0%
strike	20/20	100.0%	20/20	100.0%

5.3　考察

5.3.1　評価結果の考察

input type = ”month”，input type = ”time”はいずれも再現率が低い結果となった．これらは，識別方法や書換え方法が他の要素と違い特殊であり，入力欄とテキストが組み合わさった複数の箇所すべてのノードを正確に識別する必要がある．しかし，3.2 節で指定したノードの探索範囲では識別できなかったので，書換えを行えなかった．

評価実験で書換えが行えなかった原因として，ノードの探索範囲が狭かったことと識別に用いる文字列とは別の文字列を使用していたことが挙げられる．ノードの探索範囲が狭かった例として，input type = ”tel”に書き換わる箇所の入力欄が 3 つで構成されている場合に，2 つ目の入力欄までしか書き換えられないという例があった．別の文字列を使用していた例として，input type = ”number”で「台」や「サイズ」といったテキストを使用している例があった．これら以外にも，スペルミスにより文字列を取得できない事例が存在した．

書換えを行わない箇所を書換えてしまう原因は，書換え箇所でないノードの探索範囲内に偶然識別に用いる文字列が存在することが挙げられる．例として，input type = ”number”で「人」や「個」といった文字列を誤って識別した．

5.3.2　ツールの再評価

ノードの範囲を広くし，識別できなかった文字列を新たに追加することで，適合率と再現率がどのようになるかを再び評価した．ノードの探索範囲は，これまでのツールが親の親まで探索するのに対し，更にその親まで追加して探索するよう定めた．評価の対象とする HTML 文書は，5.1 節の評価実験と同様のものを対象とした．その結果全体の適合率は 88.0%，再現率は 85.3% となった．ツールの変更前と比較した結果，適合率が 2.8% 減少し，再現率が 0.9% 増加した．これらの結果から，識別に用いる文字列とノードの探索範囲を変えても最適な文字列や範囲は一意に定まらないと考えられる．そこで，今後はユーザが識別に用いる文字列とノードの探索範囲をカスタマイズして書換えを行えるようなツールの作成を検討する必要がある．

6　おわりに

本研究では，HTML4 から HTML5 への書換えを行うために文脈を考慮した書換え支援の方法を提案した．また，HTML5 への書換えツールを設計，実現した．ツールで書き換えた HTML 文書の評価を行った結果，適合率は 90.8%，再現率は 84.4% となった．これらの評価をもとにノードの探索範囲や識別に使用する文字列を変更し，さらに評価を行った．その結果，適合率は 88.0%，再現率は 85.3% となり，提案方法の有効性を確認した．

今後の課題は，識別に用いる文字列やノードの探索範囲をユーザが要素ごとに指定可能にすることと，作成したツールを利用して，HTML4 から HTML5 への書換えを試せるような Web サイトを提供することなどが挙げられる．

参考文献

[1] J. Hedley, “*jsoup Java HTML Parser*, ” http://jsoup.org/, 2011.

[2] J. Varese, “*HTML5 Pack for Dreamweaver CS5* の使い方，”http://www.adobe.com/jp/devnet/dreamweaver/articles/html5_update_for_dwcs5.html, 2010.

[3] World Wide Web Consortium (W3C), “*W3C Confirms May 2011 for HTML5 Last Call, Targets 2014 for HTML5 Standard*, ” http://www.w3.org/2011/02/htmlwg-pr.html, 2011.

[4] 白石 俊平, HTML5&API 入門　キャンバス、Video から Web Sockets まで–次世代 Web 標準の全容，日経 BP 社，2010.