HTML 校正ソフトウェアの開発

06H008 上森 史利

目次

1 はじめに 1

2 専門用語とソフトウェアの説明 2
2.1 HTML と HTML 文書 . . . 2
2.2 W3C . . . 2
2.3 要素 . . . 2
2.4 HTML 4.01 . . . 3
2.5 DTD . . . 3
2.6 Another HTML-lint との違い . . . 4
2.7 システムの概要 . . . 5
2.8 開発環境 . . . 5
2.9 処理の流れ . . . 6
2.10 各処理の説明 . . . 7

3 結果と考察 11
3.1 各要素の実際の処理結果 . . . 11
3.2 考察 . . . 18

4 まとめ 19
4.1 今後の課題 . . . 19

付録 A 付録 1 21
A.1 付録 1.1 . . . 21

概要

HTML 4.01 が策定されて 10 年が経つ。HTML 4x は以前のバージョンとは違い、デザインは CSS を用いて、HTML は文書の論理構造のみを記述する。これにより、メンテナンス性の向上、アクセシビリティの向上、検索エンジン最適化などのウェブサイトを制作する側、利用する側双方に対してメリットがある。しかし、多くのウェブサイトは HTML 4.01 の仕様に沿っていない。その理由として既にコンテンツとして完成した HTML を HTML 4.01 の仕様に変更するための時間や知識が、ユーザーに無いからではないかと私は考えた。この問題を解決するためには、知識の無いユーザーでも簡単に仕様に沿った HTML 文書に変更できるソフトウェアが必要である。そこで私は変更を加えたい HTML ファイルを指定するだけで、仕様に沿った HTML に校正してくれるソフトウェアを開発した。動作の流れとして、ユーザー側でソフトウェアに校正するファイルを指定する。次に保存したいディレクトリの選択とファイル名を入力し、校正を開始するボタンを押すだけである。なお、本ソフトウェアは保存ファイル名の入力以外の操作は全てマウスから行うことができる。ソフトウェアは受け取ったファイル内を捜査する。本ソフトウェアで処理する項目は大きく分けて 3 つある。1 つ目に、HTML 4.01 で非推奨とされている要素の削除。2 つ目に、ブロック要素に内包できない要素の削除。3 つ目に、テーブル要素の文法の比較と削除。最後に、校正した HTML ファイルをユーザーが指定したファイル名で保存する。本ソフトウェアをテストした結果として、本研究で開発したソフトウェアにより一部の HTML 文法の校正に成功した。また時間や知識の無いユーザーの手間を減らすことにも成功した。現状の問題点として仕様に沿わない要素を削除するのみで追加はしないため、元の HTML ファイルに要素が足りない場合に対処ができない。また、属性の理解と判断ができない。今後の課題として、現状の問題と併せてソフトウェアのユーザーインタフェイスの改良を行う。

1　はじめに

HTML 4.01 が 1999 年に策定されて今年で 10 年となる。HTML 4x は以前のバージョンとは違い、HTML には文書の論理構造のみを記入し、デザインは記述しない。これにより、メンテナンス性の向上。アクセシビリティの向上。検索エンジン最適化などのウェブページを制作する側、利用する側双方に対してインターネットがより有用なものになる。しかし、未だに W3C[*1][1][2] の定めた仕様に沿っていないウェブサイトが多数存在している。その一因として既に仕様に沿っていない HTML 文法で書かれたウェブページを持っている製作者が、自分のページを仕様に沿った HTML に書き換えていないことが問題ではないかと考える。または、製作者本人に既存のウェブページを書き換える意識があったとしても、HTML に対する知識が無い場合やまたはその逆で HTML に関する知識を持ちながら、処理に掛かる手間や時間等の理由から、改善する意識が意識が無い場合もありえる。しかし、HTML を学ぶことや既存のウェブページを書き換えることは、いずれにしても製作者の時間を大きく奪うものである。そのような問題を根本的に回避するには、製作者がウェブページの変更に時間を割かれずにウェブページを正しく書き直せるソフトウェアが必要である。そこで今回、既存のウェブページを HTML 4.01 の仕様に沿ったものに変換するソフトウェアを開発し、その効果を検証した。論文の構成として、論文の骨組みでこの論文内で使われる用語や HTML の基礎的な説明とシステムの概要。各処理のイメージ。結果と考察で実際の処理結果をまとめる。まとめで今回のソフトウェアで有用性と今後の課題を示す。

[*1] WWW で利用される技術の標準化をすすめる団体。WWW 技術に関わりの深い企業、大学・研究所、個人などが集まって、1994 年 10 月に発足した。

2　専門用語とソフトウェアの説明

この章では HTML と、それに関わる用語の定義と基礎知識を記述した後、ソフトウェアの各処理を解説する。

2.1　HTML と HTML 文書

HTML とは、HyperText Markup Language を略した言葉で、W3C が策定したマークアップ[*2]言語である。最新版は 1999 年 12 月に策定された HTML 4.01。その仕様に沿って書かれた文書を HTML 文書と呼ぶ。文書中にタグと呼ばれる文字で文を指定することでブラウザ上で文書を閲覧や画像や音声、動画、特殊な効果や他の HTML 文書へのリンクを埋め込むことができる。HTML 文章とは仕様書 (DTD) に定められた記述方法に従った文書。タグ (TAG) によって要素 (ELEMENT) を宣言し、マークアップすることでテキストなどの情報に構造上の意味を与えた文書。

2.2　W3C

W3C とは、World Wide Web Consortium の略で、WWW 技術に関わりの深い企業、大学・研究所、個人が参加している国際的な産学官共同コンソーシアム。発足は 1994 年 10 月。活動内容は Web で利用される技術仕様の標準化。技術仕様は草案、勧告候補、勧告案を経て、最終的に勧告が行われるが、各段階で仕様を公開し、一般のユーザーから広く意見を求める。その意見の取り纏めと次回仕様への反映を行う。Web の持つ機能を最大限に引き出すと同時に、Web のユニバーサルアクセス[*3]を実現するのが目的。

2.3　要素

製作者が意図的に他と区別しようとした文書などの情報。HTML ではタグによって宣言される。全ての要素には親子関係がある。

2.3.1　ブロック要素

ブロック要素とは HTML 文書構造を構成する基本要素。HTML 4.01 では (例外を除き) ブロックレベル内でのみ文書を記入してよい。要素によって内包できる要素が違う。

2.3.2　インライン要素

インライン要素とは文章に特定の機能や役割を持たせる要素。いずれの要素もインライン要素を内包することはできるが、ブロック要素を内包することは認められていない。

[*2] マークアップとは HTML でのマークアップとは、文書に効果を与えるために (多くの場合) タグで文書の前後を囲い目的の要素を宣言すること。

[*3] 「言語、文化、あらゆる機器からのアクセス、障害の有無に関わらず」と言う意味。

2.4 HTML 4.01

HTML 4.01 は、HTML 4.0 を全般的に改訂したもの。HTML とはそもそも、文章の論理構造を記述する言語であった。しかし以前の規格である HTML3.2 から、見た目に関する要素拡張の結果、見栄えを記述するタグが取り込まれた。そこで、HTML 4.01 では文書の論理構造を記述するという本来の目的に立ち返り、フォント情報、配列、色のような文書のレイアウトに関する見栄えの部分は CSS[*4]で設定する。現在の標準的な HTML の仕様である。以前のバージョンの HTML との互換性を保つために 3 種類の DTD を持っている。

2.5 DTD

Document Type Definition の略で「文書型定義」の意味。文書を記述する際、その文書中でどのような要素や属性が使われているかを定義したもの。Strict,Transitional,Frameset の 3 種類があり、それぞれ使用が許されている要素と属性に違いがある。今回の研究で制作するソフトウェアでは 3 種類の DTD の中で、もっとも厳密な定義されている DTD である Strict を基準とする。フレーム、見た目に関する要素と属性は原則非推奨または禁止。以下の記述は Strict の DTD 宣言である。

!<**DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN""http://www.w3.org/TR/html4/strict.dtd"**>

[*4] Cascading Style Sheet の略で Web ページのレイアウトを定義する規格。

2.6 Another HTML-lint との違い

Another HTML-lint [3] と今回開発したソフトウェアとの違いを表 **1** に示す。

	Another HTML lint	開発した HTML 校正ソフトウェア
チェック方法	DTD に則した文法 アクセシビリティガイドラインに準拠した項目 作者が望ましくないとした項目	DTD(Strict) に則した文法
チェック後の処理	警告文 警告数に応じた点数表示	違反要素の削除
利用方法	URL を指定する 入力欄に直接ソースコードを記述する ローカルファイルをアップロードする	ソフトウェアで HTML ファイルを指定する

表 1　Another HTML-lint との比較

今回開発したソフトウェアと Another HTML-lint とのもっとも大きな違いは、文法的ミスを発見した場合の処理方法にある。Another HTML-lint は文法的ミスを発見した場合の処理として警告を行うが、実際の HTML 文書に変更を加えるのはユーザーであった。しかし、本ソフトウェアはユーザーの手間、時間等の観点から、文法的ミスがあった場合、自動で HTML 文書に変更を加える。

2.7　システムの概要

システムの概略図を図 1 に示す。

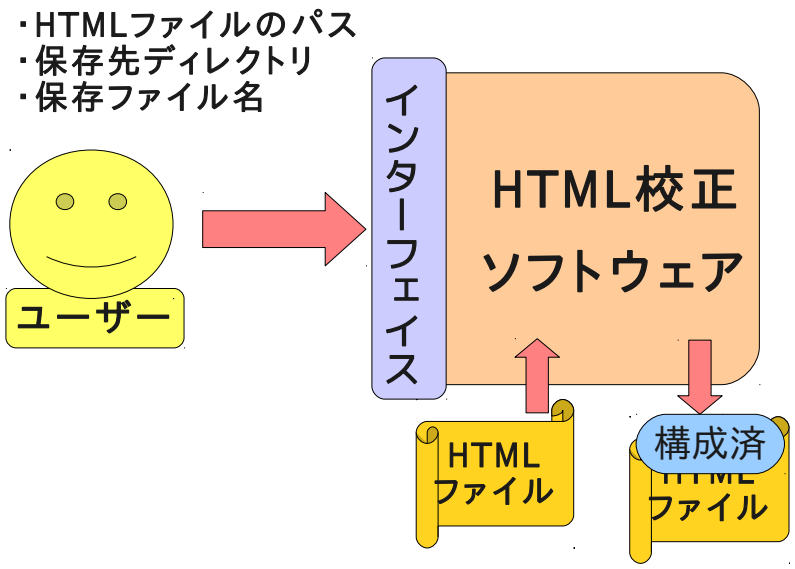
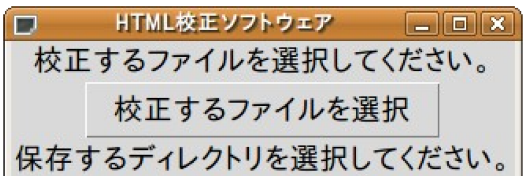


図 1　システムの概略図

2.8　開発環境

- ハードウェア (AT 互換機)
  - CPU:Intel(R) Celeron(R) CPU 2.80GHz
  - Cache Memory:256KB
- OS：Vine Linux 4.2　/ Ubuntu Linux 9.04　[研究室内マシン/ 個人所有マシン]

- 開発言語：python[4] 2.5 / 2.6 [研究室内マシン/ 個人所有マシン]

2.9　処理の流れ

本ソフトウェア起動後の操作は全て、GUI 画面から入力が可能である。実際のインタフェイスを図 **2** に示す。