

ブラウザに依存しない WWW ページの記述を目的とした HTML エディタの作成

伊藤 充男 杉村 貴士 島 和之 松本 健一 鳥居 宏次
奈良先端科学技術大学院大学情報科学研究科

WWW ページの記述が HTML 規格に準拠していない場合、その解釈は WWW ブラウザに依存して異なる。このため、その WWW ページの作者が使用しているブラウザと異なるブラウザを使用した場合、作者が意図している表示と異なることが多い。WWW ページを作成するときは、HTML 規格に準拠して、特定のブラウザに依存しないように作成することが重要である。しかし、現実にはほとんどの WWW ページはブラウザに依存している。本研究では、HTML 規格に準拠した WWW ページの記述を支援する HTML エディタを作成した。このエディタを用いることにより、WWW ページの作者が HTML 規格を習得しなくても、ブラウザに依存しない WWW ページを記述することができる。

An HTML Editor for Creating Browser-Independent WWW Pages

Atsuo Ito, Takashi Sugimura,
Kazuyuki Shima, Ken-ichi Matsumoto and Koji Torii

Graduate School of Information Science
Nara Institute of Science and Technology

When a WWW page does not conform to the HTML specification, its interpretation may differ between browsers. So, when a user uses a browser that is different from the one that the WWW page creator used, the shown page may be different from the one that the creator originally intended. It is important WWW pages conform to the HTML specification and are independent of browsers. But in reality most of the WWW pages are created with only a few of the browsers in mind. In this paper, we propose an HTML editor for creating WWW pages that conform to HTML specification. With this editor, it becomes possible to create browser-independent WWW pages without complete knowledge of HTML specification.

1 はじめに

インターネットの発展に伴い大学などの教育機関だけでなく、個人や企業などでも WWW(World Wide Web) ページを所有することが多く見られるようになった。WWW ページは HTML(Hyper-Text Markup Language) と呼ばれるページ記述言語に従って記述する。HTML の規格は W3C(World Wide Web Consortium) [1] によって定められているが、現在出版されている書籍や WWW ページで述べられている HTML の説明は必ずしも HTML 規格に準拠していないことが多い。したがって、それらを手本として記述された規格外の WWW ページが WWW に氾濫している。これは、HTML の学習を目的として既存の WWW ページのソースを参照する人にも悪影響を及ぼす。

WWW ブラウザは HTML 規格に準拠して開発されているので、表示しようとする WWW ページが規格に準拠している場合は、どのブラウザでもほぼ同じ表示を行う。しかし、WWW ページが HTML 規格に準拠していない場合、その解釈はブラウザに依存するため、閲覧に利用するブラウザによっては、WWW ページの作者の意図したものと異なることが多くなり、問題となる。また、WWW ブラウザによっては HTML 規格に拡張機能を取り入れて独自の HTML を解釈するものもある。そのような拡張機能を利用した WWW ページは、他の WWW ブラウザでは作者の意図通りには表示されない。

このような問題に対して、現状では3つのアプローチが挙げられる。1つ目は、WWW ページの作者が表示の確認に利用した WWW ブラウザを WWW ページ中で言明する方法である。2つ目は、WWW ページの作者が複数の WWW ブラウザでの表示を確認する方法である。3つ目は、WWW ページが HTML 規格に準拠しているかどうかをチェックするツールを用いる方法である。

1つ目の方法では、WWW ページを閲覧する人が作者が利用したものと同一 WWW ブラウザを用いることにより、同じ表示を得ることができる。しかし、これは WWW ページを閲覧する人が様々な WWW ブラウザを持っていることを前提とするので、ほとんどの場合問題を解決できない。

2つ目の方法では、WWW ページの作者が複数の WWW ブラウザで表示を確認することにより、WWW ブラウザの独自の拡張に依存している部分を見つける。また、現在最も普及している WWW ブラウザは少数に限られるので、それらの WWW ブラウザでの表示を確認すれば、ほとんどの場合は正しく表示されると考えられる。しかし、WWW ページを変更する度に、複数の WWW ブラウザで表示を確認する作業が必要となり、望ましくない。

3つ目の方法では、既存の WWW ページが HTML 規格に準拠しているかどうかをツールを用いて調べ、準拠していない部分を準拠するように変更する。こ

のようなチェックツールは、WWW ページのどの部分が HTML 規格に従っていないかを指摘する。ただし、それをどのように修正すれば良いのかは使用者が考えなければならない。このため、修正のための労力は無視できない。

そこで本研究では、WWW ブラウザに依存しない WWW ページを記述することを目的として、HTML 規格に準拠した WWW ページの作成を支援する HTML エディタを提案する。これまでも WWW ページ作成作業を支援する HTML エディタはいくつか提案されている。しかし、これらの HTML エディタは使い易さや WWW ブラウザ独自の拡張への対応に重点を置いたものが多く、HTML 規格に準拠しているものはほとんどない。これらの従来の HTML エディタに対し、本研究では HTML 規格に準拠した WWW ページを容易に作成できることを目指す。

2 HTML 規格と WWW ブラウザの解釈

2.1 HTML

WWW ページは HTML で記述される。HTML は SGML(Standard Generalized Markup Language) を応用したものであり、HTML で用いられる文字集合と構文は SGML 宣言と DTD(Document Type Definition) により定義されている。

- SGML 宣言：HTML で記述される WWW ページで用いられる文字コードやタグの長さの制限などが明記される。
- DTD：HTML で記述される文書はパラグラフ・リンク・イメージなどの型を持つ要素から構成されている。DTD では HTML で用いるこれらの要素型などが宣言され、HTML 文書の構造が記述される。

2.2 HTML 規格と WWW ブラウザ

これまでに規定されてきた HTML には以下のようなものがある。

- HTML 2.0 (RFC 1866)
- Internationalization of the HTML (RFC 2070)
- HTML 3.2 (HTML 3.2 Reference Specification) [2]
- HTML 4.0 (HTML 4.0 Specification) [3]

現在では、W3C (World Wide Web Consortium) により HTML 3.2, HTML 4.0 が推奨されており、広く利用されている WWW ブラウザは HTML 3.2 に対応している。

Microsoft 社の Internet Explorer や Netscape Communications 社の Netscape Communicator などの WWW ブラウザでは独自の拡張を加えた文法を定めており、Microsoft 社は Internet Explorer の DTD を公開している。

2.3 タグと要素

HTML 文書は HTML 規格で定められている階層化された要素から成り、例えば BODY 要素には P 要素や TABLE 要素が含まれる。また、タグは要素の開始位置または終了位置を表すものであり、例えば <P>P 要素のコンテンツ</P> のように書かれる。

2.4 WWW ブラウザによるタグの相違

現在、Netscape Communications 社の Netscape Communicator と Microsoft 社の Internet Explorer は最も広く利用されている。これらの WWW ブラウザは、基本的には W3C の HTML 規格に準拠しながら、HTML 規格にないタグを導入することにより、独自の拡張機能を持つ。しかし、特定の WWW ブラウザ独自の拡張機能を利用して記述した WWW ページを、他の WWW ブラウザで表示すると正しく表示されないという問題が起こる。

ここで HTML 3.2, HTML 4.0, Internet Explorer, Netscape Communicator の文法でそれぞれ定義されたタグの違いを示し、WWW ブラウザに依存するタグを整理する [1]。

表 [1] 中の数は、A(縦)にあって B(横)にないタグ(正確には要素)の数を表す。例えば、HTML 3.2 にあって HTML 4.0 にないタグの数、つまり HTML 4.0 で削られたタグの数は 3 つである。この表より、HTML 4.0 や Netscape Communicator, Internet Explorer は HTML 3.2 を基礎にして拡張が加えられたものであり、HTML 3.2 に含まれないタグを多く含んでいることが分かる。また、HTML 4.0 では Netscape Communicator と Internet Explorer の共通のタグが取り入れられていることも分かる。

表 1: DTD によるタグの違い

A \ B	HTML3.2	HTML4.0	NCom4.0	MSIE3.0
HTML3.2	0	3	1	0
HTML4.0	23	0	17	10
NCom4.0	19	15	0	9
MSIE3.0	22	12	13	0

Ncom: Netscape Communicator
MSIE: Internet Explorer

3 HTML エディタのユーザインタフェース

HTML の様々なタグを利用することにより、文章中にハイパーリンクを設けたり、画像を表示させたりして、WWW ページに多彩な表現を与えることが容易に実現できる。しかも、HTML はコンピュータ言語の一種であるが、その構文はプログラミング言語のようなコンピュータ言語に比べると、非常に簡潔である。また、WWW ブラウザには、その時表示している WWW ページのソースファイルを参照する機能が提供されていることが多いため、他人の作成した WWW ページのソースファイルを参考にして HTML を学習することが可能である。このため、WWW ページの作者は利用したいと考えるタグとその機能を断片的に学習する傾向にあり、W3C により推奨されている HTML 規格の構文はあまり知られていない。また、WWW ページの中に構文上正しくないタグの使用があっても、WWW ブラウザはそれを無視して解釈し、エラーメッセージを出力しないため、WWW ページの作者は誤りに気づかないことが多い。しかし、構文上正しくないタグの使用があると、WWW ブラウザによって表示が異なるという問題がある。本研究で提案する HTML エディタでは、HTML 規格の構文を知らなくても、規格に準拠した WWW ページを作成できるように、WWW ページ作者を支援する。この章では、提案する HTML エディタのユーザインタフェースの設計とその理由について述べる。

3.1 HTML エディタを用いた WWW ページの作成手順

本エディタを用いて WWW ページを作成する手順を図 1 に示す。

本エディタでは、HTML ファイルとして省略できないタグや省略すべきでないタグが初期設定される。例えば、<!DOCTYPE> や <TITLE> タグは記述されなければならないと DTD で明記されているため、初期設定される。

本エディタを用いた WWW ページの作成では、[タグの編集]・[文書作成]・[ブラウザでの表示確認]の作業が任意の順で繰り返される。ここで、各作業の内容は次のとおりである。

[タグの編集] タグの挿入・削除・移動・変更、および、タグの属性値の追加・変更・削除。

[文書作成] 「タグの編集」作業以外の文書記述。

[ブラウザでの表示確認] 作成した WWW ページのブラウザによる表示確認。

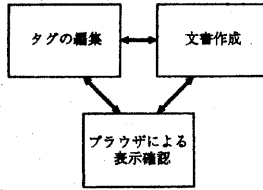


図 1: 作成手順

3.2 タグの編集

WWW ページの構造は HTML のタグで示されるので、HTML 規格に準拠した WWW ページを作成するためには、WWW ページにタグを正しく挿入することが必要である。また、WWW ページの作成中には、一度挿入したタグを削除したり、移動したりすることが必要な場合がある。よって、ここではタグの挿入・削除・移動に注目して、提案する HTML エディタのユーザインタフェースについて述べる。

正しくないタグの使い方の例としては、以下のようものが挙げられる。

- 任意の WWW ページで必ず使用しなければならないタグを挿入していない。
- 文章中のある範囲を示すようなタグにおいて、その範囲の開始を示すタグと終了を示すタグとが対応していない。
- 別のタグで指定される範囲の中でしか有効でないタグを、その範囲外で使用する。

提案する HTML エディタでは、タグの挿入・削除・移動において制約を設け、WWW ページ作成中の任意の時点において HTML 規格に準拠した構文上正しい状態を維持する。

HTML 4.0 で定められているタグの中には、現在広く用いられている Netscape Communicator や Internet Explorer では使用できないタグが含まれているため、そのタグを用いた HTML 文書はこれらの WWW ブラウザでは適切に表示されないと考えられる。また、HTML 4.0 では HTML 3.2 で規定されていたタグや属性の一部が削除されており、その中には HTML 4.0 で新しく導入されたタグに置き換わるものがある。このため、新しいタグに対応していない WWW ブラウザでは HTML 4.0 では使用が勧められていないタグが必要である場合がある。HTML 3.2 から HTML 4.0 への過渡期におけるこの問題を解決するために、W3C は、今後段階的に廃止されるタグを一切含まない HTML 4.0 Strict DTD だけでなく、今後は廃止されるが現在使われているタグも含む HTML 4.0 Transitional DTD を公開している。本研究では WWW ブラウザの現状を考慮して HTML 4.0 Transitional DTD を用いることとし、HTML 4.0 では定められているが WWW ブラウザ側では対応でき

てないタグの使用をオプション設定で禁止できるようにする。

3.3 WWW ブラウザによる表示確認

本エディタにより編集される WWW ページは編集中の任意の時点において DTD で定められた構文に関する誤りを含まない。つまり、作成される WWW ページは任意の時点において WWW ブラウザに依存しない状態が保持されているため、WWW ページの作成者は編集中の任意の時点においてブラウザで WWW ページの表示を確認することが可能となる。

この利点を活かし、作成される WWW ページの表示のされ方を確認する作業に伴う手間を軽減するために、本エディタは編集されている WWW ページの WWW ブラウザによる表示を容易に確認できる機能を持つ。

4 HTML エディタ

4.1 HTML 文書の構文解析

HTML エディタの作成にはオブジェクト指向言語の Java を使用する。以下、データ構造を主眼においた本エディタの設計について述べる。

HTML の要素の階層構造は HTML の構文により定められている。例えば、HTML 要素に含まれる要素は HEAD 要素と BODY 要素に限られ、HEAD 要素に含まれる要素は TITLE 要素や BASE 要素などに限られるため、要素の階層構造は木構造で表される (図 2)。

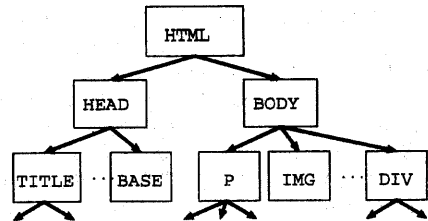


図 2: HTML 要素の階層構造

このため、HTML 規格に準拠した WWW ページに記述されている要素の階層構造も木構造で表される。ここで、HTML クラス、HEAD クラスのように各要素をクラスとしたデータ構造を採用する。また、これらのクラスに共通な特徴をもったスーパークラスを次のように定める。ただし、String は文字列、Vector は配列を表すクラスである。

elementName は要素名を表し、HTML 文書中の該当する要素に含まれる要素が child に記憶される。配列 properElement にはその要素のコンテンツとして

構文上許される要素名を保持しておき、保持された値は `getMenuItem()` メソッドにより参照される。ここで、要素の階層関係において要素 A の直接上位階層にある要素 B を要素 A の親要素、要素 A を要素 B の子要素と呼ぶことにする。

表 2:

クラス名	Element
変数	String elementName Vector child
定数	Vector properElement
メソッド	Vector getMenuItem()

本エディタでタグが挿入されると、そのタグが表す要素の親要素のインスタンス変数 `child` に挿入された要素のインスタンスが加わる。また、HTML 文書からタグが消去されると、そのタグが表す要素の親要素のインスタンス変数 `child` から消去されたタグが表す要素のインスタンスが削除される。この処理を HTML 文書作成開始時から順次続けてしていくことで、任意の時点での HTML 文書の構造が得られる。

4.2 HTML エディタの構成

本エディタは、作成中の HTML 文書を構造的に表示する HTML 文書表示部、テキストを編集するテキストエディタ部、WWW ブラウザでの表示を確認する WWW ブラウザ部の 3 つの部分から構成される。ここで、テキストエディタ部と WWW ブラウザ部は HTML 文書表示部から呼び出されたときのみ現れるものであり、HTML 文書作成開始時に常に必要な項目が初期設定された HTML 文書表示部のみ現れる。

● HTML 文書表示部

HTML 文書表示部では HTML 文書の構文解析に基づいた文書構造の情報を表示し、階層関係のある箇所ではタグとそのコンテンツをインデントして表示する。HTML 文書表示部では、文書の編集はタグの追加・削除のみ許され、文字サイズ・文字の色などフォント関連のタグが挿入されるとその効果が表示に反映される。また、表の行数・列数はタグの属性値として扱うこととし、タグの挿入・削除操作では指定しない。

● テキストエディタ部

HTML 文書表示ブラウザ上でタグが指定されると、テキストエディタ部である文書編集ウィンドウが開いて文書の編集が可能となる。ただし、そのタグの表す要素のコンテンツに他のタグが存在する場合は文書編集ウィンドウは開かれない。テキストエディタ部では本文の一部を記述するための文字入力が可能であり、その編集部分を含む要素のタグの属性値を編集できる。また、テキストエディタ部では「<」・「&」などの

特殊文字はそれぞれ「<」・「&」のように実体参照されるように変換される。

● WWW ブラウザ部

WWW ブラウザ部は、HTML 文書の WWW ブラウザでの表示を確認するとき用いられ、作成された HTML 文書を解析した結果を表示する。

4.3 ソフトウェアコンポーネント

本エディタは、HTML 文書表示部・テキストエディタ部・WWW ブラウザ部というある 3 つの部品から構成されている。提案した HTML エディタ同様これらの部品もソフトウェアであり、ある程度の汎用性をもつことで他のアプリケーションの部品にもなりうる。このようにある一定の機能をもった再利用可能なソフトウェア部品はソフトウェアコンポーネントと呼ばれる。プラットフォーム非依存の Java を基礎とする Java Beans はソフトウェアコンポーネントの概念を採用した技術の 1 つであり、以下のように定義されている。

Java Bean は再利用可能なソフトウェアコンポーネントであり、ビルダーツールで視覚的に扱えるものである [4]。

WWW ブラウザの 1 つである Hot Java は Java Beans である HTML コンポーネントを含んでおり、実際に Java Beans を視覚的に扱うことのできるビルダーツールを用いて容易に作成できる。そこで、本エディタは Java Beans として利用できるソフトウェアコンポーネントを有効に用いて作成した。

5 おわりに

HTML 規格に準拠していない WWW ページの解釈は閲覧に利用する WWW ブラウザによって異なり、作者の意図通りに表示されない場合がある。そこで本研究では、WWW ブラウザに依存しない WWW ページの作成を目的として、HTML 規格に準拠した WWW ページを記述する HTML エディタを提案した。

WWW ページ作成中、WWW ブラウザでの表示を確認する作業が多いと考えられる。そこで、提案したエディタでは、エディタのメニュー操作から最低限の機能をもったシンプルな WWW ブラウザを起動できるようにした。また、本エディタで作成される HTML 文書は作成中の任意の時点において HTML 4.0 に準拠しているため、任意時に WWW ブラウザで表示確認できる。このため、WWW ページ作成中の作業の効率が高まると期待できる。

参考文献

- [1] "The World Wide Web Consortium,"
URL: <http://www.w3.org/>.

- [2] Dave Raggett, "HTML 3.2 Reference Specification, W3C Recommendation 14-Jan-1997," URL: <http://www.w3.org/TR/REC-html32>.
- [3] Dave Raggett, Arnaud Le Hors, Ian Jacobs, "HTML 4.0 Specification, W3C Recommendation 18-Dec-1997," URL: <http://www.w3.org/TR/REC-html40/>.
- [4] Sun Microsystems, "JavaBeans™ API Specification," July 1997.

付録

以降では、タグを空要素タグと非空要素タグとに分けて述べる。空要素タグとは、例えば
 のように開始タグのみが定義されているタグであり、</BR> のような終了タグの使用は禁じられている。非空要素タグは、例えば <H1> と </H1> のように開始タグ (<H1>) と終了タグ (</H1>) の両方が定義されているタグである。

A タグの挿入

ユーザはメニュー操作により、HTML 文書中にタグを挿入するものとする。ただし、その挿入によって構文上誤りが生じるようなタグはメニューから選択できない。次に、空要素タグのタグを挿入する場合と非空要素タグのタグを挿入する場合とに分けて、操作手順を示す。

[空要素タグの挿入]

1. ユーザはメニューから「タグの挿入」を選択する。
2. エディタは構文的に挿入が許されるタグのメニューを開く。
3. ユーザは挿入したいタグを選択する。
4. エディタはカーソル位置にタグを挿入する。

[非空要素タグの挿入]

提案するエディタでは、開始タグと終了タグを常に同時に挿入することにより、開始タグがあるのに終了タグがない場合や、開始タグがないのに終了タグがあるという、構文上の誤りを防ぐ。通常、開始タグと終了タグの間には文章が含まれるが、入力順序として次の2つのケースが考えられる。

ケース1: 開始タグと終了タグを挿入してから、文章を入力する。

ケース2: 文章を入力してから、開始タグと終了タグを挿入する。

以下に、それぞれの操作手順を示す。

ケース1

1. ユーザはメニューから「タグの挿入」を選択する。
2. エディタは構文的に挿入が許されるタグのメニューを開く。
3. ユーザは挿入したいタグを選択する。
4. エディタはカーソル位置に開始タグと終了タグを挿入する。

ケース2

1. ユーザは開始タグと終了タグで囲む文書の範囲をドラッグする。
2. エディタはユーザがドラッグした範囲の文書を反転表示する。
3. ユーザはメニューから「タグの挿入」を選択する。
4. エディタは構文的に挿入が許されるタグのメニューを開く。
5. ユーザは挿入したいタグを選択する。
6. エディタは指定された範囲の先頭に開始タグを、最後に終了タグを挿入する。

B タグの削除

ユーザはメニュー操作により、HTML 文書中のタグを削除するものとする。ただし、その削除によって構文上誤りが生じるようなタグは削除できない。次に、空要素タグのタグを削除する場合と container のタグを削除する場合とに分けて、操作手順を示す。**[空要素タグの削除]**

1. ユーザはエディタ上に表示されているタグをマウスでクリックする。
2. ユーザはメニューから「タグの消去」を選択する。
3. エディタは指定されたタグを消去しても構文誤りが生じないかどうかをチェックし、誤りが生じない場合にのみ確認メッセージを出力する。
4. ユーザは「消去」ボタンをクリックする。
5. エディタはタグを削除する。

[非空要素タグの削除]

1. ユーザはエディタ上に表示されているタグをマウスでクリックする。
2. ユーザはメニューから「タグの消去」を選択する。
3. エディタは指定されたタグを消去しても構文誤りが生じないかどうかをチェックし、誤りが生じない場合にのみ確認メッセージを出力する。
4. ユーザは「消去」ボタンをクリックする。
5. エディタは開始タグと終了タグを削除する。

C タグの移動

ユーザはメニュー操作により、HTML 文書中のタグを移動させるものとする。ただし、その移動によって構文上誤りが生じるような位置にはタグを移動できない。

[タグの移動]

1. ユーザはエディタ上に表示されているタグをマウスでクリックする。
2. ユーザはメニューから「タグの移動」を選択する。
3. ユーザは移動させたい箇所をマウスでクリックする。
4. エディタは指定されたタグを指定された位置に移動しても構文誤りが生じないかどうかをチェックし、誤りが生じない場合にのみタグを移動する。

D タグの変更

ユーザはメニュー操作により、HTML 文書中のタグを変更するものとする。ただし、変更するタグと変更されるタグとが構文上同じ使われ方をする場合のみタグを変更できる。

[タグの変更]

1. ユーザはエディタ上に表示されているタグをマウスでクリックする。
2. ユーザはメニューから「タグの変更」を選択する。
3. エディタはクリックされたタグに置き換えることのできるタグのメニューを開く。
4. ユーザはタグを選択する。
5. エディタはタグを変更する。

E タグの属性値の編集

タグの属性値はプロパティウィンドウで編集されるものとする。ただし、値を設定することが推奨されている属性には初期値が与えられている。

[タグの属性値の編集]

1. ユーザはエディタ上に表示されているタグをマウスでクリックする。
2. エディタはクリックされたタグに関する「プロパティウィンドウ」を表示する。
3. ユーザはプロパティウィンドウ内の属性値を入力あるいは変更する。
4. ユーザはプロパティウィンドウ内の「変更」ボタンを押す。
5. エディタはタグの属性値を変更する。
6. エディタはプロパティウィンドウを消す。