

オーサリングツール Amaya に対する HTML5 編集機能の実装

繪面 友香 松原 俊一 Martin J. Dürst

青山学院大学理工学部情報テクノロジー学科

1 はじめに

HTML5 の策定が進んでおり、主要なブラウザが対応に取り組んでいる。既に HTML5 で記述された Web ページも存在し、今後の HTML 規格の主流となることが予測される。

本研究では Web オーサリングツール Amaya に HTML5 編集機能を実装した。実装の方針として、HTML 4 との違いに着目した。既存の HTML 4 編集機能を再利用しつつ、HTML5 専用の処理を加え、編集対象の文書のタイプによって HTML 4 用と HTML5 用の処理を切り替えるようにした。追加機能として、文書構造の直感的な把握を補助するため、構造を示す要素の可視化機能を作成した。

2 Amaya

Amaya は W3C と INRIA が共同で開発した Web オーサリングツールである。W3C が策定した規格や新しい機能のデモンストレーションを目的として作成された。HTML 4.01, XHTML 1.0, 1.1, CSS 2.0, HTTP 1.1 などに対応しており、ブラウザの機能も含まれる。

図 1 は Amaya の起動画面である。WYSIWYG のように編集ができ、メニューやボタンから要素・属性の挿入や変更ができる。段落やリストなどの構造を考慮して要素を挿入する機能を持ち、文書構造を重視した編集に特長がある。

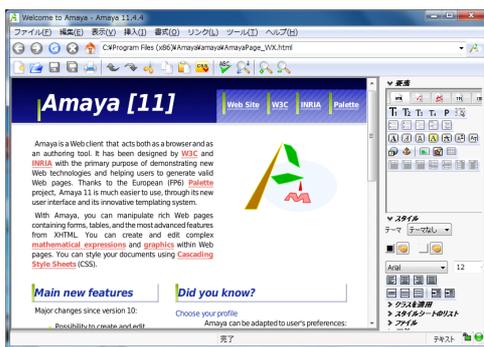


図 1: Amaya の起動画面

Implementation of HTML5 Editing for Amaya
 Yuka Ezura, Shunichi Matsubara, and Martin J. Dürst
 Department of Integrated Information Technology, College of Science and Engineering, Aoyama Gakuin University, 5-10-1 Fuchinobe, Chuo-ku, Sagamiyama, Kanagawa 252-5258, Japan
 ezura@sw.it.aoyama.ac.jp, {matsubara, duerst}@it.aoyama.ac.jp

3 実装

3.1 処理の概要

Amaya は Thot[1] という文書編集ツール開発システムを基盤に作成されている。要素が持ち得る属性、属性値、子要素、表示、外部出力する際の形式、イベント発生時の処理の情報を Thot 独自の言語で指定し、この情報に沿うように Thot がエディタ全体を制御する。

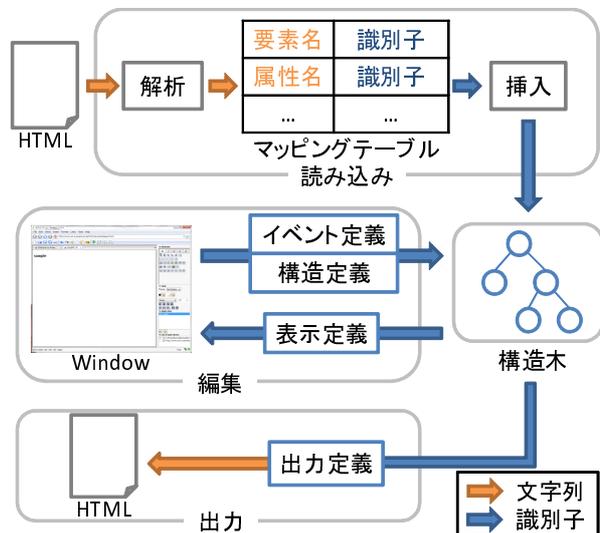


図 2: Amaya の処理の流れ

図 2 に Amaya の処理の流れを示す。Amaya の内部では要素・属性・属性値を内部識別子で管理している。要素名や属性名、属性値を解析し、マッピングテーブルを参照して内部の識別子に変換する。

3.2 処理構造の変更

HTML 4 では指定する DTD によって使用可能な要素や属性が異なる。これに対応するために、Amaya は DOCTYPE 宣言から DTD の種類を判別し、その結果とマッピングテーブルの情報を用いて、文書中の要素・属性が使用可能か判断する。使用可能なものは構造木に挿入され、Thot が定義ファイルの情報に従って管理する。構造木に挿入されたものは許可された要素・属性であることが保証されるので、HTML 4 の異なるドキュメントタイプ同士でも共通の定義ファイルで管理できる。

HTML5 編集機能の実装にあたり、既存のマッピングテーブルと定義ファイルに HTML5 の要素などを加え、DTD の一種のように扱うことを試みた。しかし、

グローバル属性やコンテンツモデルなどの、構文規則にかかわる大きな変更があるため、既存の定義ファイルを使用できなかった。そのため、HTML5 専用のマッピングテーブル、イベント定義、構造定義、表示定義、出力定義、解析機の一部を作成し、HTML5 を読み込む際にこれらを使用する。

3.3 定義の変更

HTML5 の定義は HTML 4 のものを基にした。HTML 4 からの変更点 [2] を参考にしながら、廃止または継続して使用する要素・属性の定義を書き換えた。廃止要素の表示定義を残し、UI と子要素の設定を変更することで、表示はするが、新たに挿入されないようにした。新規要素を追加するために、情報をマッピングテーブルや定義ファイルに記述した。表示定義については [3] に従い、ブラウザの一般的な表示に近づけた。もっとも変更があったのは各要素が子要素として持ち得る要素の規則である。HTML 4 は DTD でその規則を定義していたが、HTML5 では DTD が廃止された。代わりに、カテゴリーと呼ばれる似た特徴を持つ要素のグループを用いて、HTML5 の仕様書内で定義している。そこで、カテゴリーを表す内部要素を作成し、Thot 定義ファイルの中でカテゴリーの概念を再現した。

3.4 UI

廃止要素の挿入を防ぐために、HTML5 編集時には廃止要素の挿入メニューとボタンを無効にした。同様に、HTML 4 編集時に HTML5 の新規要素の挿入メニューを無効にした。

4 構造の可視化機能

HTML5 で新たに構造を表す要素が追加された。構造を示す要素の多くは表示に影響しないため、ブラウザではその存在が確認し難い。本ツールにソースや構造木の表示機能があるが、全体の文書構造を直感的に把握することに向かない。解決策として構造を表す要素の可視化機能を作成した。

HTML5 の仕様書 [4] の Sections 項に定義されている要素である `article`, `section`, `nav`, `aside`, `hgroup`, `header`, `footer`, `address` と `div` を可視化の対象にした。Sections 項で定義されている要素は他に `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `body` があるが、`body` は表示上の全部分にあたることや、`h1`, `h2` などは表示上で確認できることから対象から除外した。`div` は Section 項にはないが、表示に影響せず、一つの固まりを表す要素のため対象に加えた。メニューから可視化の有無の切り替えができる。

図 3 は同じ HTML 文書を読み込み、可視化の有無のみを変更した画面である。この HTML 文書は `article`, `section`, `nav`, `header`, `footer` を使用しているが、可視化していない左の表示ではその存在や範囲が確認できない。一方、可視化機能を有効にした右の表示では、要素の位置や内容が直感的に把握できる。

それぞれの要素の表示をデフォルトで、図 3 の右の画面のように要素の周りを四角で囲むように設定してあるが、ユーザーが変更することも可能である。表示は



図 3: 構造可視化機能

Amaya のインストール時に作成されるフォルダ中にある設定ファイルで指定され、内容を通常の CSS と同じ要領で書き換えることで表示を変更できる。このファイルを削除した場合は、次の起動時にデフォルトの設定ファイルが再度作成される。

5 おわりに

本研究では HTML 4 の編集機能を利用して HTML5 の編集機能を実装した。構造を表す要素の追加にあわせて、それらの要素の可視化機能を実装し、文書構造を意識した編集に対する利便性を向上させた。

謝辞

研究を進めるにあたり、INRIA の Vincent Quint 氏と Laurent Carcone 氏に多くの助言を頂きました。貴重な時間を割いてご協力頂き、深く感謝いたします。

参考文献

- [1] THOT.
<http://opera.inrialpes.fr/Thot.en.html>, 10 January 1997.
- [2] Anne van Kesteren and Simon Pieters. HTML5 differences from HTML4, W3C Working Draft.
<http://www.w3.org/TR/html5-diff/>, 25 October 2012.
- [3] Michael Smith. HTML: The Markup Language (an HTML language reference), W3C Working Draft.
<http://www.w3.org/TR/html-markup/>, 25 October 2012.
- [4] Robin Berjon, Travis Leithead, Erika Doyle Navara, Edward O'Connor, Silvia Pfeiffer, and Ian Hickson. HTML5, W3C Candidate Recommendation.
<http://www.w3.org/TR/html5/>, 17 December 2012.