

ルビと縦書きの Web ブラウザへの実装とその背景

塩澤 元^{†1} 松原 俊一^{†2} Martin J. Dürst^{†2}

近年の Web は様々な印刷物に代わる役割を担う場面が増えており、Web 技術を基盤とする電子書籍規格 EPUB も注目を集めている。そのため、日本語と関連が深いルビと縦書きについて、活発な議論が行われている。本研究では、従来の印刷物と異なる Web の自由度の高さを考慮し、ルビと縦書きを Mozilla Firefox へ実装し、Web 上の組版の向上を目指す。ルビに関しては HTML5 や XHTML、また妥当でないマークアップへ対応し、他ブラウザより詳細なスタイルの指定を可能にした。縦書きに関しては左から右の縦書き、横組みと縦組みの組み合わせ、縦書きの多段組を実装し、その活用についても述べる。加えて、ルビと書字方向選択の W3C 仕様についても考察する。

Ruby and Vertical Layout on the Web – Implementation and Background –

HAJIME SHIOZAWA,^{†1} SHUN-ICHI MATSUBARA^{†2}
and MARTIN J. DÜRST^{†2}

The variety of Web pages has greatly increased in recent years, and the Web is beginning to take over more and more of the roles of printed paper. In particular, the EPUB eBook format, based on HTML and CSS, has attracted much attention recently. Ruby and vertical layout, both used widely in Japanese documents, have been discussed extensively. In this research, we have investigated and implemented ruby and vertical layout in Mozilla Firefox, with the goal of improving typography on the Web. For ruby, our implementation supports HTML5, XHTML, and invalid markup as well as a wide range of styling properties not implemented on other browsers. For vertical layout, we support not only lines ordered from right to left but also from left to right, the combination of horizontal and vertical layouts, as well as multicolumn vertical layout. We also discuss the W3C specifications on ruby and writing-mode.

^{†1} 青山学院大学大学院 理工学研究科 (Graduate School of Science and Engineering, Aoyama Gakuin University)

^{†2} 青山学院大学 理工学部 (College of Science and Engineering, Aoyama Gakuin University)

1. はじめに

近年、様々な種類の Web ページが登場し、紙の印刷物の役割を担う場面が増えている。特に 2010 年は電子書籍元年と言われ、HTML と CSS を用いた電子書籍形式である EPUB が注目を集めた。しかし、現状の Web は世界の様々な表記体系を表現できない。また、高品位な組版を提供できず、書籍や論文のためには専用の文書形式が用いられている。HTML で文書を作成することは、データの構造化、ハイパーリンクの利用、画像や動画の埋め込みなどの点で非常に優位である。また最近では、デバイスの発展によってタブレット PC やスマートフォンなどの環境でも、高解像度で Web を閲覧できるようになっている。

そこで、本研究ではこのような利点を生かすため、『ルビ』と『書字方向選択』(縦書き)の二つに注目して研究と実装を行った。オープンソース Web ブラウザ Mozilla Firefox に実装し、それらの活用例などを考え、現在の Web 仕様について考察する。ルビと縦書きは日本語だけでなく、東アジアの言語と関連が深い。中国や台湾では発音を記述するためにルビを用いる。また、縦書きはモンゴル文字の表記には必須であり、縦書きによって Web のレイアウトの自由度を更に向上させることができる。そのため、単なる日本語組版の向上にとどまらず、Web ページの国際化にとって非常に意義のあるものだと考えられる。

2. ルビ

2.1 ルビの概要

2.1.1 役割

ルビとは文章中の任意の文字に対して振り仮名、説明、異なる読み方などを、文章中の文字より小さな文字で付随させるものである。昭和初期まで、書籍や新聞などではすべての漢字に対してルビを振る『総ルビ』が主流であったが、コストの問題やルビを振ることへの否定的な意見があり、現在は漢字の一部のみにルビを振る『パラルビ』が主流である。しかし、ルビ自体がなくなることはなく、現在でも多くの場面でルビが用いられている。

日本語におけるルビは発音の記述という役割以上の、美術的、意味的性質を保持している¹⁾。James Joyce によって書かれた *Finnegans Wake* の柳瀬尚紀による日本語訳『フィンゲンズ・ウェイク』は非常にルビを効果的に用いており、その文学的表現を高めている。ま

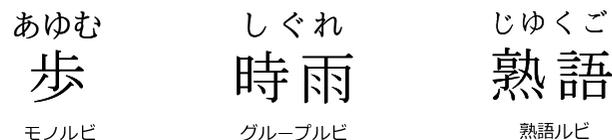


図 1 モノルビ, グループルビ, 熟語ルビ
Fig. 1 Mono-ruby, Group-ruby, Jukugo-ruby.

た, 現代のコンピュータ時代において, 漢字語圏のみに用いる発音記号としての役割だけでなくインラインの注釈 (inline annotation) として再評価する動きもある²⁾。

また, 中国語においても発音を記述するためルビが用いられる。中国語では, 発音を表すための文字体系として, 注音符號 (Bopomofo) と拼音^{ピンイン}があり, 拼音は中国やシンガポールで用いられ, 注音符號は台湾で用いられている。中国語のルビは発音を記述するという目的のため, 小学生向けまたは中国語を学ぶ人のための教科書で用いられることが多い。

2.1.2 組 版

日本語の文書の組版は, 日本工業規格である JIS X 4051 によって一定の基準が定められており³⁾, その一部としてルビの種類や組版が規定されている。ルビは, ルビをつける対象となる文字である『親文字』, ルビを表す文字の『ルビ文字』の二つの要素によって構成される。

ルビは組み方によって, モノルビ, グループルビ, 熟語ルビという 3 種類に分けることができる (図 1)。モノルビは親文字列 1 文字ごとに対応させたルビである。モノルビでは基本的にルビ文字を親文字の中心に配置する。しかし, 縦書きの場合に限って, 親文字の先頭とルビ文字の先頭を揃える肩つきルビと呼ばれるものがある。グループルビは 2 文字以上の親文字列全体にまとめてつけたルビである。当て字など, 熟語で漢字ごとに読みが分けられない場合にグループルビを用いることが多い。熟語ルビとは, モノルビがつく親文字群が熟語を構成するルビのことであり, 親文字列長とルビ文字列長の関係によって組み方が異なる。熟語ルビに関しては, W3C によって公開されている日本語組版処理の要件⁴⁾ の付録に, 詳細な説明が記載されている。

この他に, 親文字列の長さがルビ文字列未満の場合には, ルビ文字を親文字群の前後の文

字にかけて配置することができる。これを『ルビかけ』と呼び, ルビ文字の種類, そのサイズによって親文字群の前後の文字にかける量が異なる。また, この量は行頭や行末にルビが来た場合でも異なるため注意が必要である。

2.2 Web におけるルビ

本節では, Web におけるルビを用いるための仕様について解説する。現在, ルビに関して 3 種類の仕様が存在する。XHTML Ruby Annotation -W3C Recommendation⁵⁾, HTML5 Ruby Markup -W3C Editor's Draft⁶⁾, CSS3 Ruby Module -W3C Editor's Draft⁷⁾ である。本稿において, それぞれを順に XHTML ルビ, HTML5 ルビ, CSS ルビと呼ぶ。XHTML ルビや HTML5 ルビはルビのマークアップのために用い, CSS ルビはルビの表示と詳細なスタイルの指定のために用いる。ここで, W3C Recommendation となっている XHTML ルビ以外は, 今後変更される可能性が非常に高い。

2.2.1 マークアップ (XHTML と HTML5)

XHTML ルビでは図 2 に示すように, 簡単ルビ (Simple Ruby) と複雑ルビ (Complex Ruby) という 2 種類のマークアップがある。この分類はマークアップ手法の分類であり, 日本語組版上のモノルビやグループルビなどの分類とは異なる。簡単ルビは, <ruby> 内に一つの <rb> と一つの <rt> を持ち, 1 文字のモノルビやグループルビを表現することができる。複雑ルビは複数の <rb> と <rt> がある場合や, 上下にルビを付けたい場合 (縦書き時は右と左に対応) に用い, <rtc> を二つ用いることもできる。その場合には, 二つ目の <rtc> で指定されるルビは下側に表示される。

HTML5 ルビでは, 図 3 のようにマークアップする。XHTML ルビとの違いは, <rb> がないことである。ルビ文字列を示す <rt> を複数個用いて, 複数文字のモノルビを表現することができる。HTML5 ルビでは XHTML のように複雑ルビは定義されていない。そのため, <rtc> や <rbc> がなく, 両側にルビを同時に配置することができない。

2.2.2 スタイルの指定

CSS ルビではルビの詳細なスタイルを指定することができる。現在, 以下のようなプロパティが提案されている。

- (1) ruby-position
- (2) ruby-align

```

    簡単ルビ: <ruby><rb>気質</rb><rt>かたぎ</rt></ruby>
    複雑ルビ: <ruby>
      <rb><rb>気</rb><rb>質</rb></rb></ruby>
      <rt><rt>き</rt><rt>しつ</rt></rt></ruby>
    </ruby>
  
```

図 2 XHTML ルビのマークアップ例
 Fig. 2 An Example of XHTML ruby markup.

```

    グループルビ: <ruby>気質<rt>かたぎ</rt></ruby>
    モノルビ: <ruby>気<rt>き</rt>質<rt>しつ</rt></ruby>
  
```

図 3 HTML5 ルビのマークアップ例
 Fig. 3 An example of HTML5 ruby markup.

- (3) ruby-overhang
- (4) ruby-span

ruby-position はルビ文字を、親文字に対してどの位置に配置するかを指定するプロパティである。例えば、上側に配置する before や、下側に配置する afterなどを指定できる。ruby-align はルビ文字列や親文字列の寄せ方を指定するプロパティである。ruby-overhang はルビかけを指定するためのプロパティである。ルビをかける量は、日本語組版の要件⁴⁾で規定された方法で求める。ruby-span は隣接するルビ文字列、または親文字列との結合数を指定するプロパティである。ruby-span はテーブルの colspan 属性と同様の働きをし、右に隣接するルビ文字列、または親文字列を結合するために用いる。ruby-span を用いると片側が対字ルビ、片側がグループルビという両側ルビを表示することが可能になる。

2.3 ルビ表示の実装

2.3.1 Mozilla Firefox でのレイアウト処理

ルビ表示の実装について述べる前に Mozilla Firefox におけるレイアウト決定処理について述べる。Firefox では内部に HTML の DOM ツリーに対応する Content ツリーと、ページレイアウトに対応する Frame ツリーがある。Content ツリーには <div> や <p> などのマークアップに関する要素名の情報が含まれている。しかし、Frame ツリーではページのレイ

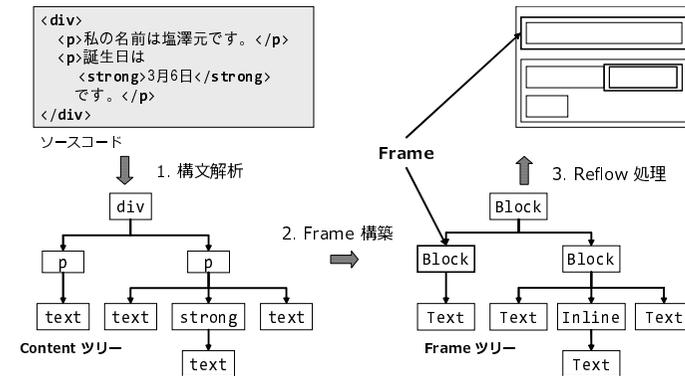


図 4 Mozilla Firefox でのレイアウト処理

Fig. 4 Process of computing layout on Mozilla Firefox.

アウトに関する情報のみを取り扱うため、要素名などの情報は保持されない。これら二つは、文書の論理構造を示す (X) HTML とスタイルを表現する CSS のような関係である。

レイアウト決定処理は図 4 に示すように行う。ソースコードを構文解析して Content ツリーを構築し、Content ツリーをもとに Frame ツリーを構築する。Frame ツリー構築後、個々の Frame の幅や座標の計算を行う Reflow 処理をする。これらの処理によって最終的なレイアウトを決定し、レンダリングを行う。

2.3.2 実装の詳細

本実装ではルビ表示のために nsRubyFrame, nsRubyContainerFrame, nsRubyCellFrame という三種類の Frame を提案し、新たなクラスとして実装した。Frame 構築処理では Frame 補完処理を用いて、HTML5 ルビや XHTML ルビ、その他妥当でないマークアップからでも適切な Frame を構築する。妥当でないマークアップからの構築処理では、妥当でない Table タグから Table の Frame を構築する処理を参考に、20 種類ほどの妥当でないマークアップを考慮した。最後に、構築された Frame ツリーをもとに座標や大きさを計算するルビ用の Reflow 処理を実装した。Frame 構築処理によってマークアップの差異を吸収することで、Reflow 処理はシンプルな実装が可能である。

表 1 ルビのマークアップのブラウザの対応
Table 1 Compatibility of ruby markup on our firefox and other browsers.

ブラウザ	シェア	HTML5	XHTML (簡単)	XHTML (両側)	XHTML (複雑)
Internet Explorer 8.0	60%			-	-
Firefox 3.6	23%	-	-	-	-
Chrome 8.0, Safari 5.0	13%			-	-
Opera 11.0	2%	-	-	-	-
Amaya 11.3	-	-			-
Our Firefox	-				

2.4 考察

2.4.1 実装の比較

本項では他ブラウザとの対応の比較を行う。表 1 にルビに関するマークアップの対応を示す。Firefox 以外の主要なブラウザでは、基本的なルビの表示が可能になっている。そのため、本実装が公式の Firefox に組み込まれることで、ほとんどすべてのブラウザでルビの表示が可能になる。また、本実装では他のブラウザが対応していない XHTML の両側ルビや複雑ルビのマークアップに対応している。

次に、各種スタイル指定への対応について比較する。既存のブラウザでは、スタイルの指定に関してほぼ未実装である。一方、本実装では ruby-align や ruby-position, ruby-overhang に対応している（一部プロパティ値未対応）。加えて、複雑ルビと ruby-span プロパティを組み合わせて指定できる機能は、Web ブラウザで初めての実装である。

2.4.2 Web におけるルビの活用

本項では Web でのルビの活用について考察する。現在の Web では、ルビを用いることの多い書籍やニュース記事などが一般化している。さらに、電子書籍形式 EPUB⁸⁾ やデジタル録音図書 DAISY⁹⁾ は Web 技術を基礎としている。このように、Web 上でのルビの必要性は以前よりも増していると考えられる。

加えて、Web の性質を考えた場合に紙面以上に便利なルビを提供することができる。まず、ルビのマークアップ自体は、コンピュータの力を借りることが可能である。形態素解析器を用いて漢字の読みを解析し、DOM 操作によって自動的に Web ページにルビを振るこ

```
<ruby>
  <rb>氣</rb><rb>質</rb>
  <rp>( </rp><rt>き</rt><rt>しつ</rt><rp>)</rp>
</ruby>
```

図 5 <rb>と<rt>をまとめたルビの例

Fig. 5 An example of ruby markup with collecting <rb> and <rt>.

とができる。また、Yahoo! JAPAN が提供するルビ振り API^{*1} では小学校の学年を指定することで、その学年のレベルにあった振り仮名を返してくれる。その結果を用いれば、同一の Web ページを小学生の学年に合わせてルビを振るということも可能である。

2.4.3 マークアップに関する考察

本項ではルビのマークアップについて考察する。現在、HTML5 ルビでは <rb> を用いることができないため、任意で <rb> の利用を可能にする提案が行われている。既存の多くのルビマークアップは <rb> が用いられている点、親文字にスタイルを与える場合に不便である点などが理由として挙げられている。また、現在のマークアップでは <rp> を用いたフォールバックの際に、『気(き)質(しつ)』のように読みが分かれてしまう。加えて、『きしつ』や『気質』という単語が検索できないという問題点も指摘されている。しかし、<rb> を可能とし、図 5 のようなマークアップが可能になれば、『気質(きしつ)』のように表示できる。この場合には、対応していないブラウザでも読みやすい表示が可能であり、また検索も可能である。

一方で、これらの問題点は XHTML のルビを用いればすべて解決することができ、マークアップの簡素さという点以外では、XHTML ルビは非常に有益であると考えられる。この点を考慮し、本実装では XHTML ルビに加え、<rb> を用いたルビ、<rt> と <rb> をまとめたルビにも対応している。

2.4.4 スタイル指定に関する考察

本実装では日本語組版を考慮し、必要のないプロパティ値には対応していない。まず一つ目は、ruby-align の distribute-letter である。ルビ文字に対して均等割りを行うことはないので必要はないと考えた。また、CSS ルビの仕様書内でも指摘されているように、

*1 <http://developer.yahoo.co.jp/webapi/jlp/furigana/v1/furigana.html>

line-edge に関しては異なるプロパティにするべきであると考えた。ruby-overhang では start と end は必要ないと判断した。これは、ルビかけをする場合、どちらか片方だけルビかけをするということは考えられないからである。

3. 縦書き

3.1 書字方向の分析

グリフを配置する方向のことを『書字方向』という。グリフを垂直に配置するレイアウトを『縦書き』、水平に配置するレイアウトを『横書き』と呼ぶ。しかし、このような『横書き』や『縦書き』という分類では、様々な種類の書字方向を適切に分類することはできない。左から右に書き進める横書き（英語など）と、右から左に書き進める横書き（アラビア語など）があり、右から左に行が進む縦書き（日本語など）と、左から右に行が進む縦書き（モンゴル語）があるからである。また、同じ言語でも一つの書字方向だけを持つとは限らない。もともと日本語では縦書きのみであったが、明治より横書きを取り入れ、どちらの書字方向も可能になっている。

Etemad は、書字方向の分析とその指定方法について述べている¹⁰⁾。それによれば、書字方向は以下の三つのプロパティによって構成される。

- (1) ブロック方向 (block progression)
- (2) 行内方向 (inline progression)
- (3) グリフ方向 (glyph orientation)

それぞれのプロパティの概要を図 6 に示す。ブロック方向は行の進む方向を決定する。行内方向は、個々のグリフを進める方向を決定する。グリフ方向は、グリフの回転方向を決定する。ここで、グリフは一般的に直立状態であり、その値は 12 時方向となるが、主言語中に他言語の文字列が含まれている場合に、このグリフ方向を変更する。この三つのプロパティを定めることによって、書字方向を決定することが可能である。

3.2 CSS による書字方向指定

Web で書字方向を指定するため、CSS3 Writing Mode Module -W3C Working Draft¹¹⁾ の策定が進められている。この CSS3 Writing Mode は、Etemad による書字方向の指定の手法を参考にしており、書字方向は以下の三つのプロパティで指定する。

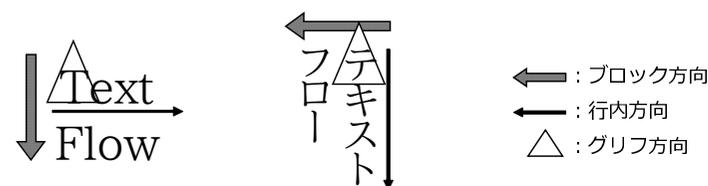


図 6 書字方向を指定するための三つのプロパティ
Fig. 6 Three properties for specifying writing-direction.

- (1) writing-mode
- (2) direction
- (3) text-orientation

writing-mode は、ブロック方向に対応するプロパティである。このプロパティ値として、上から下への方向を指定する horizontal-tb、右から左への方向を指定する vertical-rl、左から右への方向を指定する vertical-lr が提案されている。direction は、行内方向を指定するためのプロパティである。プロパティ値として、左から右へと進める lr、右から左へと進める rl が提案されている。加えて、Bidi 問題を解決するアルゴリズム¹²⁾を指定する unicode-bidi というプロパティと組み合わせて利用される。これらは縦書きの場合には、左を上、右を下と読み変えて考える。text-orientation は、グリフ方向に対応し、主言語中に他言語の文字列が来た場合のグリフ回転を指定するプロパティである。

3.3 縦書きの実装

書字方向選択実装には、既存のレイアウトエンジンの出力から、各書字方向へ変換する方式を用いる。具体的な処理は図 7 に示すように、三つの処理に分けられる。図 7 の 1 を『インライン変換』、2 を『グリフ処理』、3 を『ブロック方向処理』と呼び、それぞれについて以下で詳細を説明する。

3.3.1 インライン変換

本実装は横書きレイアウトからの変換方式を用いるため、縦書きの行を直接生成しない。そこで図 8 のように『個々のグリフの回転』と『行全体の回転』という二つの処理を行って

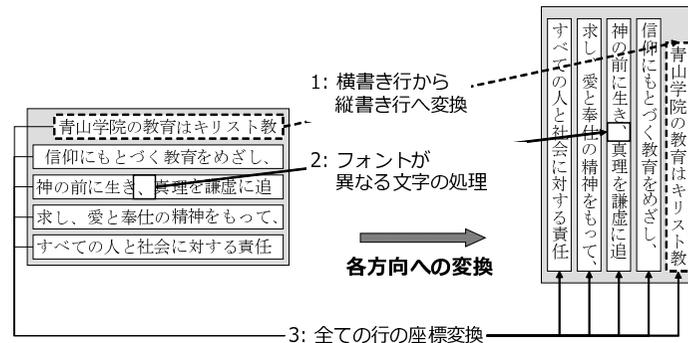


図7 異なるブロック方向への変換
 Fig. 7 Process of translation to other writing-mode.

横書き行から縦書き行へと変換する。すべての個々のグリフを左に90度回転させた状態にし、その行全体を右に90度回転させることで縦書き行を実現する。グリフの回転は行に対応するFrameの構築時に行い、行長の計算などは既存のレイアウトエンジンで行う。行の回転はテキストのレンダリング処理の際に行う。

3.3.2 グリフ処理

日本語に関してほとんどの文字は横書きと縦書きで同じグリフを使うことができる。しかし、『』や『。』などのいわゆる約物は、横書きと縦書きで違うグリフを使う必要がある。またラテン系の文字は縦書きの際には左に90度回転させることが多い。

フォントデータは内部に縦書き置換用のテーブルを保持しており、それらを用いて置換処理を行うことができる。しかし、本実装では直接それを用いる実装は行っていない。Windowsではフォント名の頭に@マークを用いると縦書きフォントが利用可能になっており、これを用いてグリフ処理を行う。また、text-orientationプロパティが指定された場合には、回転処理を別に行う必要があるが、本実装ではこれに対応していない。

3.3.3 ブロック方向処理

ブロック方向処理では行の進む方向を、指定された方向へと変換する処理を行う。本実装では、右から左に行が進む縦書き以外にも、左から右に行が進む縦書きや下から上へ行が進む

横書きへの変換も可能とした。まず、既存のレイアウトエンジン部分が上から下への横書きレイアウトを計算する。その結果に対してフレームの回転処理と座標変換を行い、行の進む方向を変換する処理を実装した。

3.4 考察

3.4.1 縦書きの活用

本項ではWebにおける書字方向の指定の活用について考察する。横書きが幅広く受け入れられている現在でも、新聞や書籍などは今でも縦書きが定着している。また、モンゴル文字によるモンゴル語では縦書きのみ可能である。このように、縦書きが定着している、また縦書きのみ可能であるという文書をWebページにできることは非常に有益である。これに加えて、書字方向の選択が可能になることで、様々な書字方向が混在した新しいWebのレイアウトが可能になり、レイアウトの自由度を上げることができる。従来の書籍のように、すべてを縦書きとして表示することがなくても、Webページの一部などに縦書きのレイアウトが可能になることは、デザインの点で非常に有益であると考えられる。

また、最近Webでは多段組が注目されており、我々は縦組みの多段組が有用であると考えている。Braganzaらはディスプレイ上における、多段組の文書のスクロール方法について調査を行っている¹³⁾。その結果、横書きの多段組は左右スクロールが必要であり、現在のWebでのスクローリングに慣れているユーザが多いため、あまり好まれないという結果を報告している。しかし、多段組のほう行幅が適切で読みやすいという意見もあると述べており、慣れの問題ではないかと結論付けている。一方、縦書きの多段組では、上下方向のスクロールとなる。そのため、行幅が適切でありスクロールも慣れたものであるため、ユーザビリティに優れていることが推測できる。本実装においては縦書きの多段組も可能となって

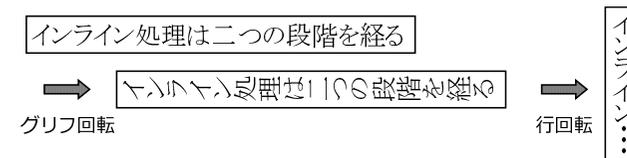


図8 縦書き行を生成する二段階の処理
 Fig. 8 Two process of generating a vertical line.



図9 縦書きの多段組の例

Fig.9 An example of multicolumn vertical layout.

おり、図9は縦書きの多段組を用いた表示の例である。

3.4.2 論理プロパティと物理プロパティ

論理プロパティは before や after などを用いて、ブロック方向の指定に依存するような位置指定を行う。それに対して、物理プロパティは left や right などを用いて、ブロック方向の指定に依存しないで位置を指定する。論理プロパティでは、logical-width は行の長さを表し、logical-height は行の進む方向の長さを表している。start は行頭

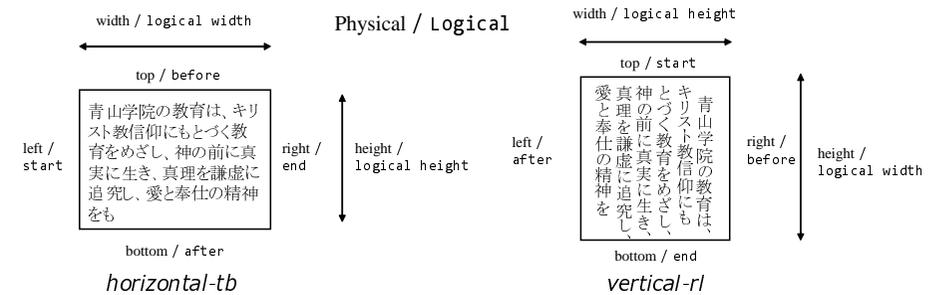


図10 物理プロパティと論理プロパティの関係

Fig.10 Relation of physical properties and logical properties.

方向を表し、end は行末方向を表している。before は先頭行の方向を表し、after は末尾行の方向を表している。図10はこれらの論理プロパティの概要図である。図10において、/の左側は物理プロパティでの指定値、右側は論理プロパティでの指定を示している。

論理プロパティは、もともと横書きと縦書きをでスタイルシートを変更する手間などは省くために用いると考えられる。しかし、今回のような座標変換を用いた実装方式では、実装する際の問題点としても考えられる。図11のように余白が指定された場合、本実装のブロック方向の座標変換を行うと指定とは異なる方向に余白が現れてしまう。そこで、前処理として、最終的に適切な位置に余白がくるよう、変換先のブロック方向に応じて事前に入力値を入れ替えている。この例であれば、上側余白に右側余白の値を入れることで、最終的に右側に余白をとることができる。この入れ替えは論理プロパティと物理プロパティの概念と一致している。座標変換を用いた場合に必要となる各方向の入れ替えは、論理プロパティの実装を行うことで除去できると考えている。

3.4.3 縦組みと横組みの組み合わせ

最後に、縦組みと横組みを組み合わせる際に問題になる幅や高さの決定について考察する。横組みの中に縦書きを入れた場合には、縦組みのボックスの高さをどのように決定するかを考える必要がある。横書きの場合、CSS ではボックスの高さが規定されていないため上から下に無限に伸びることができる。そのため、height が指定されていないボックス

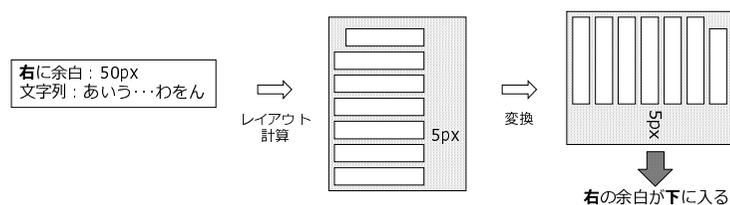


図 11 ブロック方向処理における問題点
Fig. 11 Problem of block progression process.

は、内部のボックスに応じて伸び縮みするようになっている。しかし、この規定をそのまま採用すると、横組みの中に縦組みが含まれた場合には、行がまったく折り返されない。

これを解決するため、本実装では以下のような規則を用いた。まず、それまでに height の値が指定されていた場合には、それを用いる。それが指定されていない場合には、ウインドウの幅、正確には 100vh (viewport height) という値を与える。これは横組みの場合の行幅と同じ挙動であり、きわめて直感的なものであると考えられる。

4. おわりに

本稿ではルビと縦書きの Web ブラウザへの実装について延べ、その活用や W3C 仕様について考察した。ルビの実装では、HTML5 と XHTML、多くの不整合なマークアップに対応した。そして、ブラウザでの実装例がない XHTML の複雑ルビに関して、両側ルビや ruby-span 指定も含めて実装した。書字方向選択の実装では、縦書きのレイアウトを可能とし、右から左の縦書きだけでなく左から右の縦書きなどにも対応した。また、縦書きの多段組やページの一部のみの縦書きも可能である。仕様に関する考察では、ルビに関してはマークアップと CSS プロパティについて延べ、縦書きに関しては横組みと縦組みの組み合わせに関する規定について述べた。今後、これらの考察は仕様への意見としてメーリングリストなどに投稿する予定である。これからの課題として、ルビに関してはモノルビや熟語ルビの場合の改行処理や、縦書きのキャレット選択の実装などが挙げられる。

参考文献

- 1) 由良君美. ルビの美学 (上). 言語, Vol.2, No.7, pp. 556-562, 1973.
- 2) 小林龍生. ネットワーク社会でのルビの再評価 HTML, Unicode に即して. 情報処理学会研究報告, 1998-IM-34, pp. 17-22, 1998.
- 3) 日本規格協会. JIS X 4051: 日本語文書の組版方法, 2004.
- 4) 阿南康宏, 千葉弘幸, 枝本順三郎, Richard Ishida, 石野恵一郎, 小林龍生, 小林敏, 小野澤賢三, Felix 佐々木. 日本語組版処理の要件 (日本語版) - W3C 技術ノート. <http://www.w3.org/TR/jlreq/ja/>, 2009.
- 5) Michel Suignard, Masayasu Ishikawa, Martin Dürst, and Tex Texin. *Ruby Annotation - W3C Recommendation*. <http://www.w3.org/TR/ruby/>, 2001.
- 6) Ian Hickson. *HTML5 - A vocabulary and associated APIs for HTML and XHTML - W3C Working Draft*. <http://www.w3.org/TR/html5/>, 2010.
- 7) Richard Ishida. *CSS3 Ruby Module - W3C Working Draft*. <http://dev.w3.org/csswg/css3-ruby/>, 2010.
- 8) Garth Conboy, Matt Garrish, Markus Gylling, William McCoy, Makoto MURATA, and Daniel Weck. *EPUB 3 Overview - Working Draft*. <http://idpf.org/epub/30/spec/epub30-overview.html>, 2011.
- 9) National Information Standards Organization. *ANSI/NISO Z39.86: Specifications for the Digital Talking Book*, 2005.
- 10) Erika Etemad. Robust Vertical Text Layout. In *Proceedings of 27th International Unicode Conference*, 2005.
- 11) Erika Etemad, Koji Ishii, and Shinyu Murakami. *CSS Writing Modes Module Level 3 - W3C Working Draft*. <http://www.w3.org/TR/css3-writing-modes/>, 2010.
- 12) Mark Davis. *USA#9: Unicode Bidirectional Alogrithm*. <http://www.unicode.org/reports/tr9/>, 2010.
- 13) Cameron Braganza, Kim Marriott, Peter Moulder, Michael Wybrow, and Tim Dwyer. Scrolling Behaviour with Single- and Multi-Column Layout. In *Proceedings of the 18th International World Wide Web Conference*, 2009.