

# XSLFO2.0へ向けての日本語組版機能提案

大野邦夫

株式会社ジャストシステム

今日の日本における紙の文書において、縦書きの文書が一般的である。しかし現状のWeb文書は、縦組みの日本語には対応していない。今後の日本語の情報の電子化を考えると、WebやXML文書が使いやすい縦組みの日本語をサポートすることが望まれる。さらに、近年急成長を遂げる中国や台湾、韓国などの漢字文化圏のWebやXML文書の動向を考えると、漢字組版機能を、W3Cの正式な規定として提供することは、W3Cにとっても、会員企業にとっても望ましいことである。そのような背景に基づき、2006年10月18日に開催されたXSL-WGのワークショップにおいて、日本語組版機能をXSLFO2.0に加えることを提案したので、その内容について紹介する。

## Japanese Document Layout Proposal to XSLFO2.0

Kunio Ohno

Justsystems Corporation

Though vertical layout is very popular in Japanese printed document, current web document seldom uses vertical layout. Considering the future Japanese document, to support vertical layout in web or XML document will be expected. Besides, recent quick development of eastern Asia countries as China and Korea web based Kanji layout standard will be required for W3C and its corporate members. Then Japanese layout requirement studied through JAGAT study team was proposed to the workshop of XSLFO2.0 which was held on October 18 2006. This paper describes the outline of the proposal.

### 1 はじめに

昨年の2月末に開催されたW3C Technical Plenary Meetingにおいて、日本語組版機能をW3Cの規定として実装することを関連WGに提案した[1]。その後、4月に提案内容を検討するための組織（縦組スタイルシート作業部会）がJAGAT（印刷技術協会）を中心に結成され、既存の標準であるJIS-X-4051をベースに、今後の電子化日本語文書、すなわちWeb、ケータイ画面やデジタルTV、電子ブックなどを包含する日本語コンテンツにとって要求される組版機能について検討した。毎月2回のミーティングを開催し、9月までに、用紙と版面関連、行組版関連、ページ組版関連、図・写真などの処理、表組版関連、漢文関連に大別してA4で70ページ程度の素案としてまとめた。その作業はJAGATの小野沢さんが素案を作り、それを参加者全員でレビューする形式を取った。なお漢文については日本エディタースクール小林さんが担当した。

この内容について、現状のXSLFO[2]とCSS[3]で実現不能と思われる内容をアンテナハウスの石野さんに検討していただいた。その概要をW3CのXSLFO2.0ワークショップ[4]で紹介したが、この内容が欧米の関係者に関心と呼んだ。本報告ではその概要を紹介する。

### 2 提案内容

#### 2.1 提案の背景

日本を中心とする東アジア圏における漢字レイアウトの要求について説明した（図1）。今のところ、日本での課題しか検討していないので、東アジア圏というのは憚れるが、W3C関係者に関心を持ってもらうためには日本に限定すべきではないと考えた。そのような背景の下に、すでに述べたとおりJAGATを中心として、今後の日本語組版を検討するためのタスクフォースが作られたことを説明した（図2）。日本に

### Requirements for text layout in East Asia

- Based on Han-Character (East Asian Ideograph)
  - Difference of font design framework
  - Square type face vs. ascendant / descendent line based design
- Strong market needs for vertical text layout exist
  - Publishing industries and news paper industries stick on vertical layout
  - Some educational area requires vertical text layout

図1 東アジア圏における漢字レイアウトの要求

### JAGAT Taskforce

- Currently voluntary research project conducted by JAGAT
  - Antenna House and Justsystems are involved
- Planned to input to W3C's incubator group
- Requirement: Based on JIS X 4051
  - JIS X 4051:1993 – Horizontal layout
  - JIS X 4051:1995 – Vertical layout
  - JIS X 4051:2004 – Updated
- Project editors are the core contributor for this taskforce
- Plan: align with CSS3 draft in general and add some new functionality from Japanese market needs
- Plan: harmonization with XSL 2.0

図2 JAGATタスクフォース

における状況を明確にしておくために、既存規格として制定されているJIS-X-4051について説明を加えた。さらにタスクフォース・メンバーについても具体的に紹介した(図3)。

### Taskforce Members

- Hiroyuki Chiba Japan Association of Graphic Arts Technology
- Junsaburo Edamoto Morisawa & Company, Ltd.
- Hiriji Emoto Digital Media System Co.,Ltd
- Kenichiro Ishino Antenna House, Inc.
- Tatsu Kobayashi Justsystem Corporation
- Tokushige Kobayashi Antenna House, Inc.
- Toshi Kobayashi Japan Editors School
- Osamu Ogasawara Japan Association of Graphic Arts Technology
- Kunio Ohno Justsystem Corporation
- Kenzo Onozawa Japan Association of Graphic Arts Technology
- Tetsuya Takahashi Canon System Solutions Inc.

図3 タスクフォース・メンバー

## 2.2 用紙と版面関係

### 2.2.1 日欧におけるレイアウトの相違

先ず、欧文と和文の文字配置の相違を説明した。欧文は、小文字・大文字などに応じた基線上の2種類の

高さで基線下のはみ出してポディーサイズが決まる。中学の英作文の授業で、最初にアルファベットの筆記の練習をさせられた時の筆記ノートを思い起こしていただければ良い。それに対して、和文の記述は基本的にグリッドであり、ポディーサイズもグリッドの大きさで決まる。(図4)。

### Japanese document layout

- Japanese character size is square.
- Japanese document layout is based on the grid type manuscript paper layout.

図4 日欧におけるレイアウトの相違

### 2.2.2 原稿用紙レイアウト

日本語組版は、基本的には「原稿用紙」のような組版である。版面は、文字数と行数を指定することにより定められる。XSLにはそのような指定機能はない。また、日本語組版では、各行を行の高さの整数倍の位置に配置するのが原則であるが、XSLでは、単純に矩形を積み上げる方法によって文字位置などをレイアウトする。それはXSLが行進行方向とブロック進行方向という2方向への積み上げで実現していることに起因する。積み木をベースラインに沿って横に並べ、行がいっぱいになったら次の行の端から並べ始めると考えれば分かりやすいであろう。XSLの組版において、行は逐次必要な高さ分だけずらされる。すべて同じ文字サイズで組んでいれば行の位置は揃うが、大きな文字や、ベースラインを移動して組む添え字などが出現すると、占める高さは変化し、行の位置は揃わなくなる。

CSS3にはgridという概念が存在している。しかし、CSSの世界で行進行方向に文字を升目に入れることは、和文と欧文との混在では詰め処理などを必要とすることになるためそれほど重要ではない。ブロック進行方向に行を揃える機能は必要であり、それはXSLのline-stacking-strategyを拡張すれば可能であろう。

XSLのline-stacking-strategyは、line-height、font-height、またはmax-height(以下、このような項目選択については、”line-height | font-height

|max-height”と標記する) という値を採る。CSS3のLine Stackingでは、inline-line-height | block-line-height | max-height | grid-height である。line-height が inline-line-height に、font-height が block-line-height に対応する。grid-height は、ここでの目的に合う。そのとき、行取りをした別行見出しは、行の配置をvertical-alignなどで調整すればよい。

## 2.3 行組版関係

### 2.3.1 単位

単位として「級」(Q)が必要である。1級 = 1/4mm = 0.25mm。また、XSLでは 1pt = 1/72in = 0.3528mm(PostScript Point/DTP Point) であるが、JIS Z 8305でのポイントは = 0.3514mm(American Point)である。これは、83pica = 35cm という定義から、1/12pica = 0.3514mm となる(なお、XSLでは 1pc = 12pt である)。これも指定できるようにする必要がある。ちなみに、TeXでは、1pt = 1/72.27in = 0.3514mm であり、1bp = 1/72in = 0.3528mm である。

### 2.3.2 詰め

日本語のグリフは、原則的に正方形であり、句読点や括弧なども正方形である。そこで、句読点と開き括弧の間などが間延びするのを防ぐため、詰めて組む(図5)。

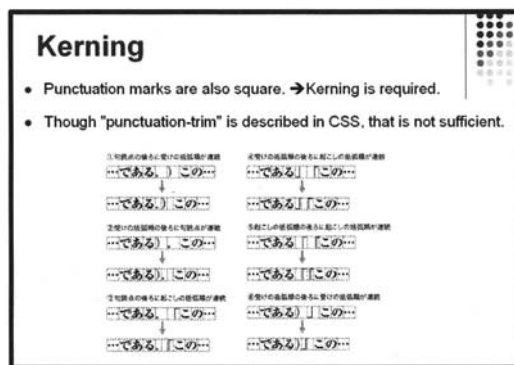


図5 詰め

### 2.3.3 ぶら下げ

句読点を、版面外にはみ出させることがあり、ぶら下げと呼ぶ(図6)。XSLにぶら下げはない。CSS3では、hanging-punctuation で、none | start | end | both を指定できる。ここでは、行末だけでなく、行頭への引用符のはみ出しのようなぶら下げも考慮している。

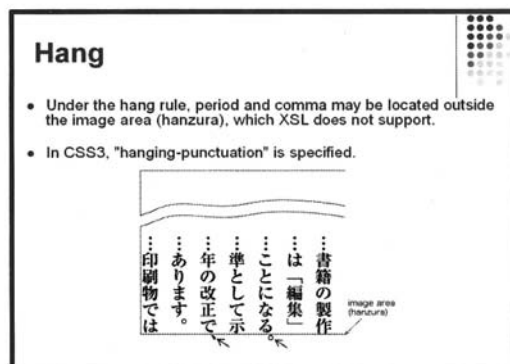


図6 ぶら下げ

### 2.3.4 和欧文間空白

JIS-X-4051では、和欧文間は原則四分アキ(全角の1/4の空き量)としている。XSLにこのような指定はない。CSS3では、text-autospace として考慮されており、JIS-X-4051にも引用されている。指定できるのは、下記の4種類である。表意文字とはUnicodeとしての分類であり、漢字やかななどが含まれる。(注: ideograph は表意文字であるが、そこにはかなのような表音文字も含まれている。)

- 表意文字と欧文数字の間
- 表意文字と欧文英字の間
- 表意文字と欧文括弧の間
- 表意文字で囲まれた欧文空白

現実には、ここで区別されるべき欧文をまず定義しなければならない。CSS3には、ラテン文字、キリル文字、ギリシャ文字、アラビア文字、ヘブライ文字が「例示」されている。JIS-X-4051は、欧文に対して「欧文用文字及び欧文間隔で構成される文章であって、横書きとし、欧文ピッチ処理の対象となるもの」とされているが、具体的な記述はなく、区別の仕方は実装依存となっている。

CSS3では、括弧以外の欧文約物に対する指定がない。括弧以外の欧文約物に対する制御も必要なのがわかっている。また、CSS3では、括弧の向きに対して区別がない。開き括弧-和文間、和文-閉じ括弧間に対しては、空きを挿入すべきではない。

### 2.3.5 行分割と禁則

行分割制御は、UAx#14: Line Breaking Properties があり、JIS-X-4051にも反映されているため、JIS-X-4051で行頭禁則和字はすべて禁則処理される。この行頭禁則和字には、長音や小書きのかななど

が含まれるが、それらは、禁則されずに組まれることが多い。XSLやCSS3には禁則を制御する機能はない。

また、半角カナは、Unicode的にはLatin文字と同じような扱いになってしまい、全角かなと同等の行分割効果が期待できない。漢字やかななどは表意文字として、(禁則を除いて)各文字間での行分割が可能であるが、Latin文字などは単語を構成するとして、ハイフン位置以外に単語途中で行分割することがない。つまり、Unicode的には連続する半角カナを途中で行分割しない。

### 2.3.6 ルビ

XSLでは、ルビは、<fo:inline-container> を使って模倣するのがせいぜいである。しかし、これでは日本語組版が要求する仕様を満足しない。なにより、行分割に対応できない。<fo:inline-container> は外側のテキストと隔離されるので、justifyもされない。ルビは、CSS3のRubyにある。この仕様で、JIS X 4051 はおおむね満足されると思われる(図7)。

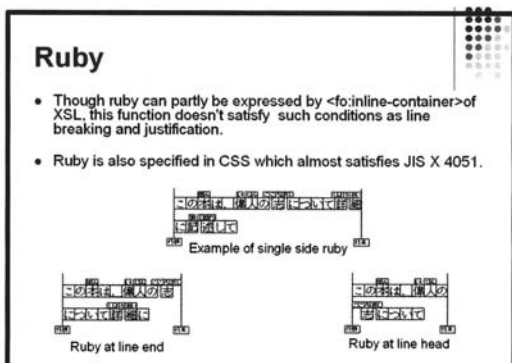


図7 ルビ

### 2.3.7 割注

割り注は、1行の文字幅に複数行の文字列を挿入する機能である(図8)。XSLでは、割注も、<fo:inline-container> を使って模倣できるかも知れないが、これも行分割に対応できず、無理がある。

CSS3には、text-combine として出てくる。none | letters | lines を指定できる。XSLにはない。

letters は、全角1文字幅に文字列を2行に詰め込む指定で、5文字まで組めると書いてある。なぜ5文字なのかは不明だが、「菫」のような組み文字に利用する。これは割注ではない。組み文字は、専用グリフの用意されていることがあるが、この指定により、任意の文字列を組み文字にできる。

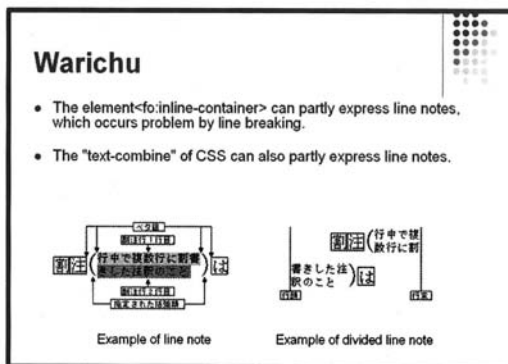


図8 割注

lines は、割注を実現する。この仕様で十分だと思われる。

### 2.3.8 縦中横

縦中横は、縦書きの行において横書きを指定する機能で、縦書き中に2桁の数字を記述するような場合に用いられる(図9)。XSLで縦中横を行なうためには、

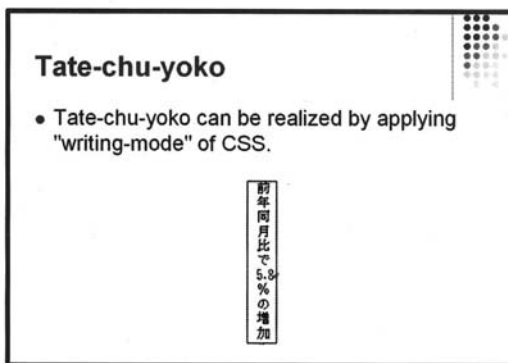


図9 縦中横

そうしたい文字列に対して回転などを指示しなければならない。自動的に縦中横に組むということはできない。CSS3では、writing-mode のところに出てくるが、縦中横にしたい部分にマークアップして明示する例であり、抽象化されているわけではない。縦中横は、writing-modeが縦書きのときだけの機能である。glyph-orientation-verticalを拡張し、ある文字数または指定幅までの文字列を横に組むような指定を用意すればよさそうである。

### 2.3.9 振分け

XSLにもCSS3にも振分けとしての構造表現はない。振分けは、割注などと違って複数行に同一振分け単位が分割されることはない。その行末までが最大の長さである。もし、行末近くで振分けが始まった場

合、振分け内での行数が相当多くなる。最大の振分け高さを指定したり、最小の振分け幅を指定したりできる必要があるであろう。(拡張方法未検討) (図10)

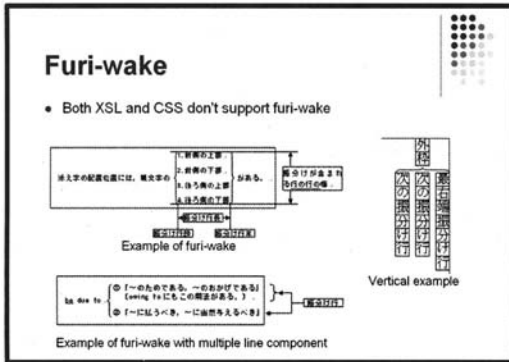


図10 振分け

### 2.3.10 圏点

XSLに圏点はない。CSS3には、Fontsにあり、font-emphasize-styleとして、accent | dot | circle | discが用意されている。現実には、◎など不足すると思われる。font-emphasize-styleの値に任意の文字を指定できればよいであろう。どこまでやるかの線引きは必要だろう。圏点とルビを同時に振る場合の規則は、JIS X 4051にはない。どう出るのかは実装依存でいいのかも知れない。(図11)

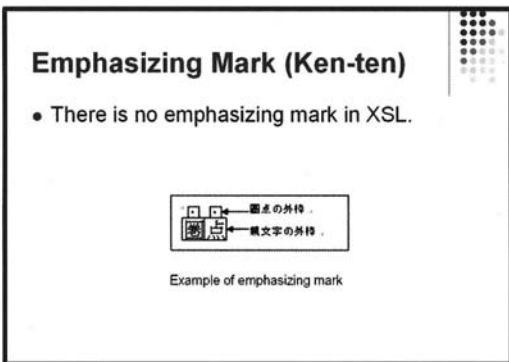


図11 圏点

### 2.3.11 下線

CSS3ではテキスト修飾における下線の線種の指定や位置の調整が可能である(図12)。しかしXSLでは下線や傍線の線種の指定や位置の調整はできない。テキスト領域にボーダーを引き、下線を真似れば、実現可能ではある。日本語組版とはおそらく関係ないが、下線は、グリフにかかってしまうことがある。そのような場合はグリフをよけて線を引くことが望まれる

が、XSLではできない。(CSS3では可能) 縦組みでの下線をどちら側に引くのか、という指定も、XSLは不可能であるがCSS3では可能である。

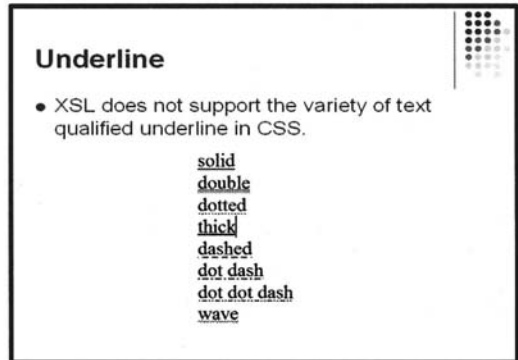


図12 下線

### 2.3.12 添え字

XSLでの添え字は、ベースラインをずらす指定をする。baseline-shift="super"、baseline-shift="sub"という指定が可能であり、どのくらいずらすかを意識する必要はない。ただし、スーパーSCRIPTとサブSCRIPTを上下に並べようと思ったら、どちらかを進行方向から戻してやる必要がある。(図13)

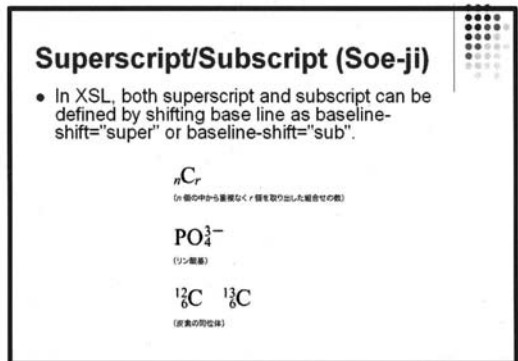


図13 添え字

### 2.3.13 タブ

XSLにもCSS3にもタブストップの概念はない。XSLでのタブ文字は、一般の空白文字と区別されていないので、まずそれを区別するためのtab-treatmentのようなプロパティを用意し、タブストップ位置を指定するtab-stopsプロパティを用意することになるであろう。(図14)

## 2.4 ページ組版関係

### 2.4.1 段抜き



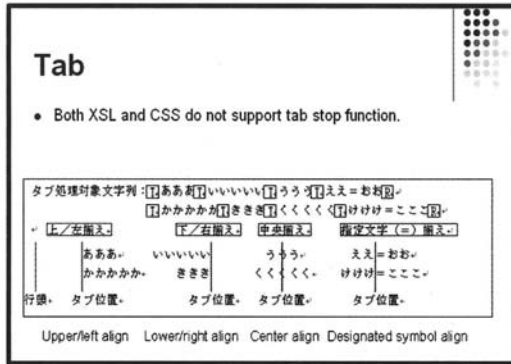


図14 タブ

XSLでは、全段抜きしかできない。図15に示すような段構成が必要である。そのためは、`span="all"`だけでなく、`span="2"`とかが指定できればよいのであろう。

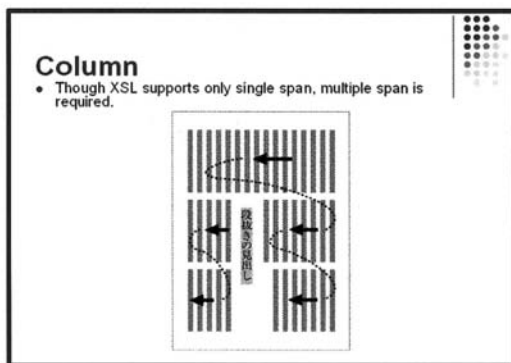


図15 段抜き

#### 2.4.2 脚注

ページレイアウトとしては、脚注だけでなく、頭注、傍注なども欲しいところである (図16)。

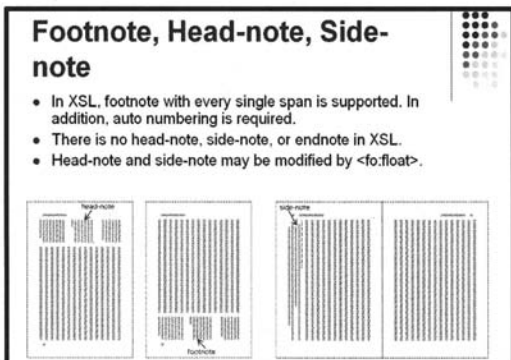


図16 脚注、頭注、傍注

XSLは、「全段抜き脚注」しかない (図17)。注番号の

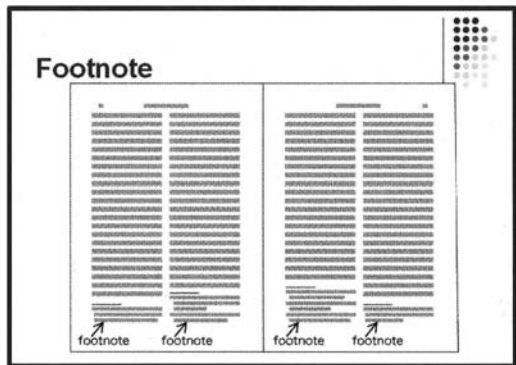


図17 脚注

自動採番もなければ、頭注、傍注、後注などもない。もともと、頭注、傍注は、`<fo:float>`によって模倣できる。傍注は、ページの外側 (小口) に置かれる場合と、右側に置かれる場合がある (図18)

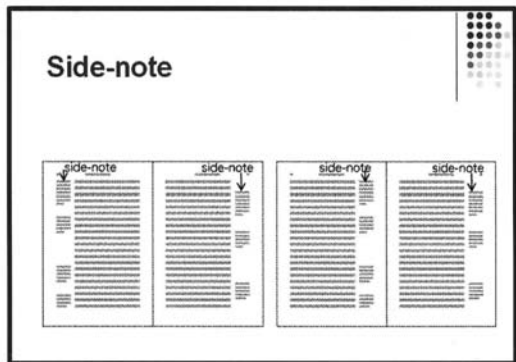


図18 傍注

縦組みでの脚注の位置は、地の側であり、天の側の頭注と相対する。XSLでの脚注は、`after`側である。つまり、縦組みではページの左側 (ノドまたは小口側) に脚注が配置される。

#### 2.4.3 ブロック進行方向の均等割り

テーブルセルに限らず、ブロック進行方向に行を均等割りする機能はない。XSLには、`display-align` と `vertical-align` というプロパティが存在している。`display-align` は、ブロック進行方向の配置を指定するもので、`before | center | after` が指定できる。`justify` はない。これは、参照領域の子領域に対して効果があり、すべての子に適用される。フローの途中にある図をページ下部に配置するということはできない。`vertical-align` は、行内の配置を指定するものであり、ブロック進行方向に関する指定ではない。CSS3では、テーブルセル内の各行のブロック進行方向への配置にも効果があるが、XSLでは効果がない。

display-align="justify" は、拡張すればよい。(図19)

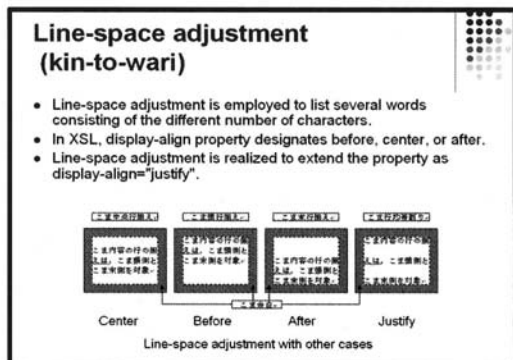


図19 ブロック進行方向の均等割り

#### 2.4.4 段間罫

段間罫はコラム間の罫線であるがXSLにはない。CSS3では、Multi-column layoutのColumn gaps and rulesにある。罫の長さを指定することはできないので、常に領域の端から端まで引かれる。

#### 2.4.5 ガイドマーク(とんぼ)

XSLにガイドマーク(とんぼ)はない。XSL Formatterでは拡張済みであり、とんぼが可能である。(図20)

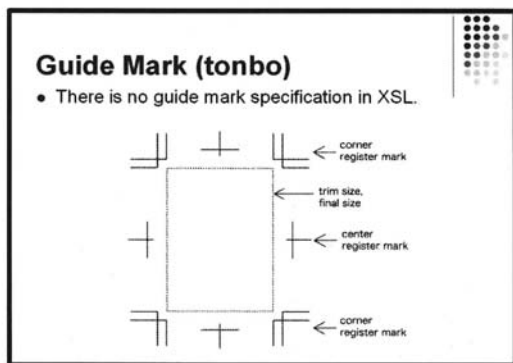


図20 ガイドマーク(とんぼ)

### 2.5 図・写真の配置

図や写真をページ上の絶対位置に置き、テキストを回り込ませることはできない。図や写真がページや段の下部にかかったとき、その量によって、それを追い出すのか追い込むのかしなければならぬ。そのような指定は、XSLにもCSS3にもない。

回り込みは<fo:float>によって片側だけ可能である。XSLでは、多段組で段に跨る図を置いて、回り

込みを行なうことはできない。CSS3では、古い Multi-column layoutのFloating in and between columnsにあるが、仕様としては不完全である。

なお、JIS-X-4051では、図や写真の配置に関して、図21、図22のような配置規則を定めている。

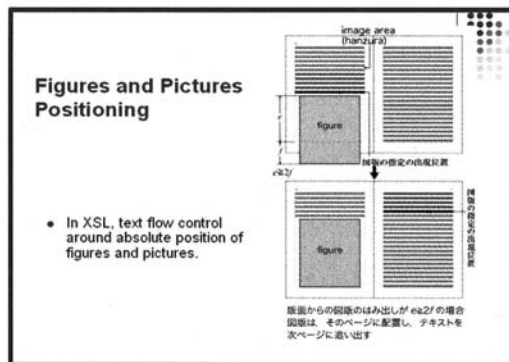


図21 図、写真の配置(1)

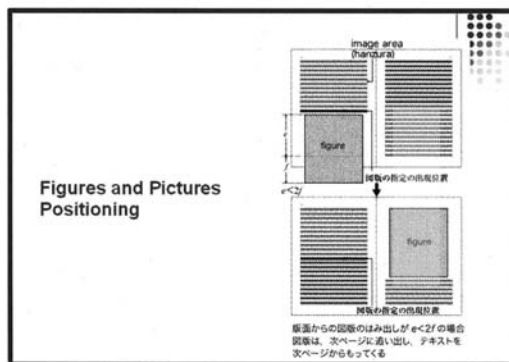


図22 図、写真の配置(2)

### 2.6 漢文

漢文の用途は、歴史文書や教科書が主であろうが、過去の引用を含む一般的な記事や随筆などで、今後広く使われる可能性がある。漢文は基本的に縦書きであり、ルビや圏点だけでなくレ点や返り点などの情報を付加する必要がある。XSL、CSSでは、漢文に関しては何も考慮されていない。(図23)

## 3 まとめ

以上の個別機能の説明に基づき、図24のように総括した。要するに、東アジアでは、グリッドベースのレイアウトと縦書きを必要とする漢字組版のニーズがある。既存のXSLFOやCSSはその機能をサポートしていないが、それらを仕様化することにより、W3Cの技

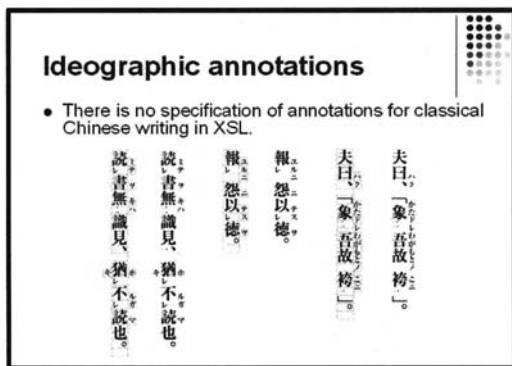


図23 漢文

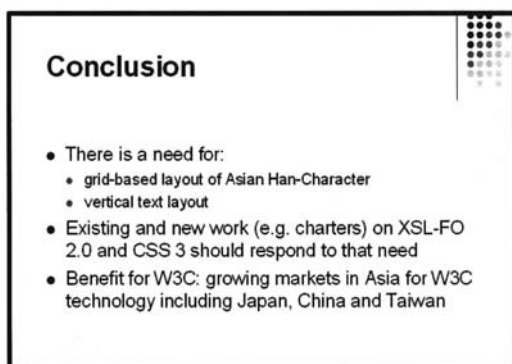


図24 まとめ

術は東アジアの漢字文化圏に大きな影響力を持つことができる。

## 4 考察

以上、日本語組版を行う上で、XSLFOとCSSが未サポートな機能をピックアップしてスライドで説明し、その機能を仕様化することがW3Cにとって大きなメリットがあることを述べた。このメリットはW3Cだけでなく、漢字をWeb上に表示する東アジアの文化圏の人々も共有するものであり、その中の一員である日本のユーザやビジネスを行う人々にとっても同様である。

説明する側としては、最初は欧米の関係者に果たしてどの程度理解して頂けるのか不安な気持ちであったが、意外にも活発な質問や意見が多数寄せられた。質問は主に説明内容の不十分さに関するものであったが、振分けの改行やルビと圏点の同時表示のような、内容についてかなり理解しないと発することができな

いような質問も見受けられた。さらにこの説明の元となったJIS-X-4051に関しての質問も相次ぎ、ついにはJIS-X-4051の英訳を要求する決議がなされてしまった。

他方、XSLFOの主な機能を実現しているアンテナハウス社のFormatterについても質問が出され、在席していた小林社長が対応する場面もあった。さらに小林社長から、XSLFOとCSSとの整合性に関する要望が提案され、総論としては賛意を得たが、個別的には多くの困難があることが指摘され、W3C内部でも考え方の多様性、WGごとの方針の差異を感じさせられた。

## 5 謝辞

本報告は、図3に示すJAGATのタスクフォース（縦組スタイルシート作業部会）の成果をとりまとめたものであり、関係各位に謝意を表するものである。さらに、W3Cの立場で、数多くの助言を頂いたフェリックス佐々木さんに御礼申し上げます。

## 6 あとがき

XSLFO2.0に日本語組版機能を加えて欲しいという我々の要求は歓迎されたのであったが、次の段階として日本側がどのように対応すべきかが今後の課題である。

取りまとめた文書の英訳が当面の課題であろう。これだけでも訳語などで苦勞しそうであるが、職人芸的な日本語組版の世界を、世界的に知ってもらうためには、それが有益であることを具体的に示す努力をせねばならない。さらに東アジアとして取り組むというのであれば、中国や韓国とも連携して行かねばならない。残された課題の大きさを痛感するとともに、関係各位のご努力を期待する次第である。

### 参照情報・文献

- [1] 大野邦夫; “W3C Technical Plenary Meeting 参加報告”, 情報処理学会デジタルドキュメント研究会研究報告, DD55-4, (2006.5)
- [2] <http://www.w3.org/TR/xsl/>
- [3] <http://www.w3.org/Style/CSS/>
- [4] <http://www.w3.org/Style/XSL/2006-Workshop/>