

**Webや電子書籍にフォーカスした
“日本語組版の要件”
JLReq-d**

JLReqの仕掛け人、小林さんから

なぜJLreqだったか

2005年当時

- ・ 非日本語話者が、日本語の組版の要件を知る手段はなかった
- ・ 一方、Web Browserやワードプロセッサなどのドキュメント閲覧環境の開発体制は、急激にグローバルでのワンソース管理に移行していた
- ・ このままでは、日本語の環境は取り残されてしまう。
- ・ 英語で読める日本語組版の要件をまとめよう！

JLreqの功罪

良かったこと：

- ・ CSSがWebKitがJLreqを参照するようになった
- ・ W3Cに、*Lreq開発の動きが多数起こった

よくなかったこと：

- ・ JLreqの金科玉条化
- ・ 要件のプライオリティ付けがなかったため、実装コストと効果のトレードオフが計れず、要件の実現が停滞するようになった

JLreqは結局何だったのか

明治20年以来の活版技術における現実的な組版規則の明文化

- ・ 活版技術の物理的な制約に起因する規則が混在
- ・ 電算写植時代の過渡的な規則の混在
- ・ 整版（木版印刷）技術や卷子本の蓄積が生かせていない
- ・ 紙への印刷を前提とした版面設計ありきの体系となっている
- ・ 欧文組版と和文組版のフォント設計思想の相違が反映されていない

**でさあ、木田さん
どうする？**

(話者スイッチ)

たいへんな丸投げをいただいた、JLReqタスクフォースの議長、木田でございます。

JLReqは結局何だったのか

広い範囲のデジタルデバイス上で
ちゃんとした日本語組版を可能にした
巨大なインパクトをもたらした技術文書

本題に入ってしまう前に、小林さんが触れられた「JLReqは結局何だったのか」について、私なりに捉えてみたいと思います。

一言で言うと、JLReq、日本語組版処理の要件は、世界に日本語の組版というものを説明し、現在見るように広い範囲のデジタルデバイスで日本語組版を可能にするという、巨大なインパクトをもたらした、そしてもたらし続けている技術文書であると言えます。

これがwebドキュメントであって常にアクセスできるというインパクトも大きく、日本でもJIS X4051ではなくこちらを参照される方も多かったのではと想像します。

小林さんも触れられたように、このアプローチは世界に影響を与え、たくさんのフォロワーを生みました。かなりの開発労力がかかるのにも関わらず、今やW3Cサイト上には16のスキriptに対して同じ形式の文章が存在します。(Arabic, Chinese, Ethiopic, Latin, Hebrew, Bengali, Devanagari, Tamil, Hangul, Mongolian, Lao, Thai, Khmer, Javanese, Tibetan & Japanese)

規格化/実装に際して 問題点も見えてきた

同時に、規格化、実装に際して問題点も見えてきました。

例えば、Appleが電子書籍の日本語対応を行なった際、「英語のドキュメントがあるからよろしく」、とは行かず、webkitなどの実装チームに対してかなりのサポートが私のチームから必要となりました。結局日本語組版をよく理解し、日本にいる組版や出版のエキスパートにアクセスのあるエンジニアが必要だったのです。他社でも同じような状況だったと想像します。これは日本語のエキスパートにアクセスのない会社にとって実装が困難になることを意味します。

またJLReqが登場して10年以上になりますが、業界全体を見ると、規格化、実装がスムーズに進んでいるとは言えない状況です。

「問題はどこにあるのでしょうか？」

国際化ソフトウェア 多言語環境との整合性

問題点の一つは国際化ソフトウェア、多言語環境との整合性です

例えばJLReqはJISの文字セットに対して文字クラスを定義しています。が、国際化されたソフトウェアはUnicodeで動いているので、文字クラスを拡張する必要があります。そのためにはJLReqの文字クラスを深く理解し、Unicodeのそれぞれの文字がどのクラスに属するかを決めてゆく必要があります。

またJLReqは日本語の組版要求だけでなく、日本市場という小宇宙における組版の要求に応えるためのさまざまな要素を含んでいます。英文や数式の組み方などです。規格化・実装の際にはここは読み飛ばす必要があります。これらはJIS文字セットがギリシャやロシアの文字を全角で含んでいるのと似ています。これらが、国際化ソフトウェアとコンフリクトを起こします。

また、OSやWebブラウザのテキストは欧文のベースラインをもとに組み立てられています。これと、JLReqの定義する版面をもとにしたグリッドをどのように対応させれば良いでしょう。

プライオリティがわからない デフォルト値の指定がない

問題点の二つ目は、**プライオリティが示されていない、デフォルト値の指定がない**という問題。これはソフトウェア開発にとっては本当に困ります。

というのは、これらが示されていないと自ら判断を行なう必要があるのですが、組版のプライオリティには、理屈以外に歴史や文化や市場の実情が深く関わってくるので、門外漢に判断は不可能です。結局、ここでも日本語と日本市場に精通したエンジニアが必要となります。

まあ、私のような日本人エンジニアの飯の種になるといえばそうなのですが、書いておいてくれれば解決するようなレベルの話を組織ごとに再発明する必要があるわけで、より高度な日本語組版に話を進めるための障害となります。

これが、規格化と、実装の遅れを生み出しています。仕様というのは実装されないと絵に描いた餅なので、なんとか解決したいところです。

Webと印刷は基礎となる仕組みが根本的に異なる

最後に、もっと根本的な、つまり面白い問題も見えてきました。印刷とWebでは目的は同じながら基礎となる仕組みが根本的に異なる、という根の深い問題です。

根本的に、どのように、異なるのでしょうか：

Webと印刷は基礎となる仕 組みが根本的に異なる

組版が最後の瞬間になるまで決定されない

最も大きな違いは、組版が固定されるタイミングです。

印刷では送り手のもとで組版が固定され、単一のイメージが多数の受け手の元に配布されます。

対して、Webのモデルでは、変化できるデータが送り出され、組版が固定されるのは受けての手元で表示される瞬間です。ユーザーの必要性やデバイスによって、無数に異なるレイアウトが生成されます。リフローするというとわかりやすいでしょう。

印刷では、組版の結果がイメージとして最終的に固定される前に、通常さまざまなレベルで人の目でのチェックが入ります。このため、組版結果が破綻していても、人の手での発見、調整が可能です。結果、組版規則は暗黙のうちに人の目によるチェックを期待とするようなものになっています。

例えば長いルビが行末にかかるると行の調整が大きくなってスカスカの行が出てきてしまいます。これ、印刷では人の目を経ますので組版の技術者や校正の段階で、例えばルビを分割したり、例えば言葉を変えたり、などして対処します。

対してWebのモデルでは、どこで行の折り返しが起こるのかが予測つきませんし、組版の結果を人の目でチェックする段階はありません。よって、完全に自動的に組版を行なっても破綻しないような規則にすることが重要です。

しかしこの特徴のために、表示をデバイスに合わせたり、ユーザーのニーズに合わせたり、印刷にできない芸当が可能です。

Webと印刷は基礎となる仕組みが根本的に異なる

組版が最後の瞬間になるまで決定されない
動的挙動がある

もう一つ、印刷にない要素として動的挙動があります。

同じくルビの例ですが、ルビを含んだテキストを選択したら、ルビはハイライトされるべきか？ コピーして、プレーンテキストの環境にペーストしたら、何がペーストされるべきでしょう？ 読み上げはどうなるべきでしょう？

これも、印刷にはない特徴です。

Webと印刷は基礎となる仕組みが根本的に異なる

組版が最後の瞬間になるまで決定されない
動的挙動がある

印刷とwebでは目的は同じながら基礎となる仕組みが根本的に異なる、と言うと、そりゃそうだ、とおそらく多くの方が同意してくれると思います。しかしその違いの意味、重要性には気が付きにくい。なぜかと言うと、従来の努力の多くは、紙の印刷をコンピューター上で再現するための努力だったわけです。例えばPDF、電子書籍、ページレイアウトソフトなどは重要な成功例です。なので暗黙のうちに紙の印刷が最終目標だと皆さん思ってる。

しかしこの両者は別物だ、と考えると世界は違って見えます

馬車と自動車 書写本と印刷本

ここで少し状況を俯瞰してみましょう。歴史の中には、今お話ししているのと似た現象、つまり目的は同じだけれど基礎となる技術が変わった、というイベントがたくさんあります。

馬車から自動車、そしてテキストの世界では、書写本から印刷本への移行がその例でしょう。

みなさん世界で最初の交通違反切符の話をご存知ですか？ 時は1896年、明治29年の英国、なんと制限時速の4倍の猛スピード、時速13キロで走っていた車を警官が自転車で8キロにわたって追いかけて、1シリング、現代にしてどれくらいなんだろう、の罰金を取りました。もしかするとこれは世界で最初のカーチェイスかもしれません。追いかけたのは自転車ですが。数年後の1899年、New Yorkの街をさらに速い時速19 kmの神をも恐れぬスピードで走っていた自動車がこれも自転車で追いかけてきた警官に捕まり、今度はかわいそうに牢屋送りになりました。これがUSでの最初の例です。

みなさんお気づきかもしれませんが、これらの交通規則はもちろん自動車のためではなく、その当時の主流であった馬車に対する規則です。その当時自動車は馬車と同一の規則の下にあったのです。

もう一つ、書写本から印刷本へ。印刷が発明された当初の技術開発は、印刷ならではの発展というよりは、手書きの書写本を模倣することに大きな努力が払われました。どこかで聞いたことのある話ですね。

これらの例のように、技術の移行の最初は新しい技術で前のモノを模倣するところから始まっています。これは新しい技術が世に受け入れられるために必須のステップのようです。

「しかし、車はいつまでも馬車の速度で走るわけには行かないし、印刷はいつまでも手書きの書籍を真似るわけにはゆきません。」

(脱線：書写から印刷へ：受注生産による工芸品から、資本集約的なビジネスに（生産設備、在庫）、そして。面白いのは印刷のもたらしたビジネスへのインパクトです。書写本の時代は本は当然のことながら非常に高価で、手工業的に受注生産で作られていました。これが印刷になると、印刷機械、そして一度にすることでコストが下がりますので、そうすると印刷された紙の在庫を抱える、という設備と在庫のビジネス。つまり資本がものを言うビジネスに変貌しました。Webになるとどうでしょう？ 生産設備や在庫は必要がなくなります)

(脱線2：書籍の広がり→メガネの広がり→レンズ職人→顕微鏡(1590)、望遠鏡(1608)→ケプラーによる天体の運動の解明 (1609-1619)→ニュートンの万有引力(1687)。)

(jlreq-d はとっかかり。転がし始める役目)

Webにおける組版が印刷組版を目標にしている現状は歴史の過渡期

このような例を見てゆくと、**Webの組版は印刷とは別物であって、印刷組版を目標にしている現状は歴史の過渡期ではないか**そう捉えることで新しい可能性が見えてくるかもしれない、という視点を得ることができます。

Webのための 日本語組版規則が必要

つまりはWebのための日本語組版規則が必要、というわけです。

これが今日の一番重要な点です。ここさえ、なるほど、と置いていただけたら今日のプレゼンは成功です。

Webのための日本語組版規則、一体どんなものでしょう。

その前に、一つ私たちが使っている言葉を説明させてください。

デジタルテキスト

デジタルテキスト。今までWebと説明してきましたが、同様の性質を持っているテキストはWebだけではありません。例えばリフロー型の電子書籍、もっと身近なメール、それからデバイスのUI文字列もそうです。ですからこれらをまとめる言葉をして、とりあえずデジタルテキスト、と呼びたいと思います。

デジタルテキストのための 日本語組版規則が必要

この言葉で言い換えると、このような性質を持ったデジタルテキストのための組版規則が必要、ということになります。

デジタルテキストのための 日本語組版規則が必要

組版が最後の瞬間になるまで決定されない

動的挙動がある

国際化ソフトウェア、多言語環境との整合性

プライオリティ・デフォルト値

デジタルテキストのための日本語組版規則

ここまで、JLReqの問題点から出てきた**三つの課題**についてご説明しました。

- ・国際化ソフトウェア、多言語環境との整合性
- ・プライオリティ・デフォルト値
- ・デジタルテキストのための日本語組版規則

もう一点、最後の課題をご説明します。それはアクセシビリティです。

国際化ソフトウェア、多言語環境との整合性

プライオリティ・デフォルト値

デジタルテキストのための日本語組版規則

アクセシビリティ

生活のさまざまな局面に対してアクセシビリティを確保すること、テキストにおいては読字障害や弱視へ対応すること、は社会的にその重要性が高まっています。

従来、アクセシビリティに対応した書籍、例えば教科書は通常版とは全く異なるモノとして提供されてきました。しかしデジタルテキストにおいては、組版が最後の瞬間になるまで決定されないという特徴を活かして、ユーザーに合わせた表示を一つのソースから行うことが可能です。例えば iPhone では文字の大きさや太さを変えることができます。

同様に、読字障害や弱視に対応するためにどのような考慮が必要か、その知識はもはや一般の組版要件文書に書かれているべきであろう、と JLReq タスクフォースでは考えています。どのような議論を行なっているか、JLReq タスクフォースのメンバーであり、日本 DAISY コンソーシアム 技術委員会委員長、村田真さんに、話していただきます。



読字障害や弱視への対応: どんな日本語レイアウトが読みにくい？

- ・区切りが分からないので、分かち書きされていないと読みにくい？
- ・縦書きが苦手で、横書きでないと読みにくい（またはその逆）？
- ・総ルビでないと読みにくい？
逆にルビがあると読みにくい？
- ・ルビが大きめでないと読みにくい？
- ・ひらがなが読みにくい？
逆に漢字が読みにくい？
- ・行間が広めでないと読みにくい？
- ・文字と文字の間が広めでないと読みにくい？
- ・文字や背景の色によっては読みにくい？
- ・明朝体は尖って見えて読みにくい？
- ・「き
や」という改行は読みにくい？

研究

科研費基盤研究B(2019-2022)による実証的な研究

- ・ 大阪医科薬科大学LDセンターに通う児童を対象
- ・ 課題文として光村図書出版小学校国語教科書3年から抜粋した文などを使用
- ・ 視線追跡装置Tobii pro スペクトラムを用いて眼球の動きを測定
- ・ 測定データをプログラムで分析
- ・ 最初は分かち書きと階段状ベースラインについて実験

実験中の動画



現場からの声

普通の教科書が読めない人にボランティアがデイジー教科書を作成し、提供している。

- ・ 令和3年度の利用者実績は15,059人

現場での合意に基づき明文化されている規則

- ・ 行間260%

そこまでは至っていない知見

- ・ 総ルビ（アンケートで上位）
- ・ 分かち書き
- ・ 広めの字間
- ・ ラグ組

デジタルテキストの組版

最後に、デジタルテキストのための組版の方向性について、現在の議論をお伝えしようと思います。

プロジェクトはまだ始まったばかりでして、今からお話する部分は固まった物ではなく、現時点ではこう考えているという内容です。皆様のさまざまな視点からのフィードバックをいただくと大変ありがたい。

デジタルテキストの組版

課題の克服

- ・ Unicode化・多言語環境の中の日本語組版を定義
- ・ プライオリティ・レベル分け。シンプルな規則から複雑な規則へ
- ・ 日本語の知識を前提とせず基礎から記述
- ・ ベースラインシステム、フォントの組み合わせ
- ・ 完全な自動化に耐える頑強な規則に
- ・ アクセシビリティの知識、一般的な読みやすさへのガイドライン

デジタルテキストの組版

見直し

- ・ 複雑な機能
- ・ 実際と乖離している規則
- ・ 活字の物理的/経済的な制約から来るとされる規則
- ・ デジタルな規則、二分三分四分

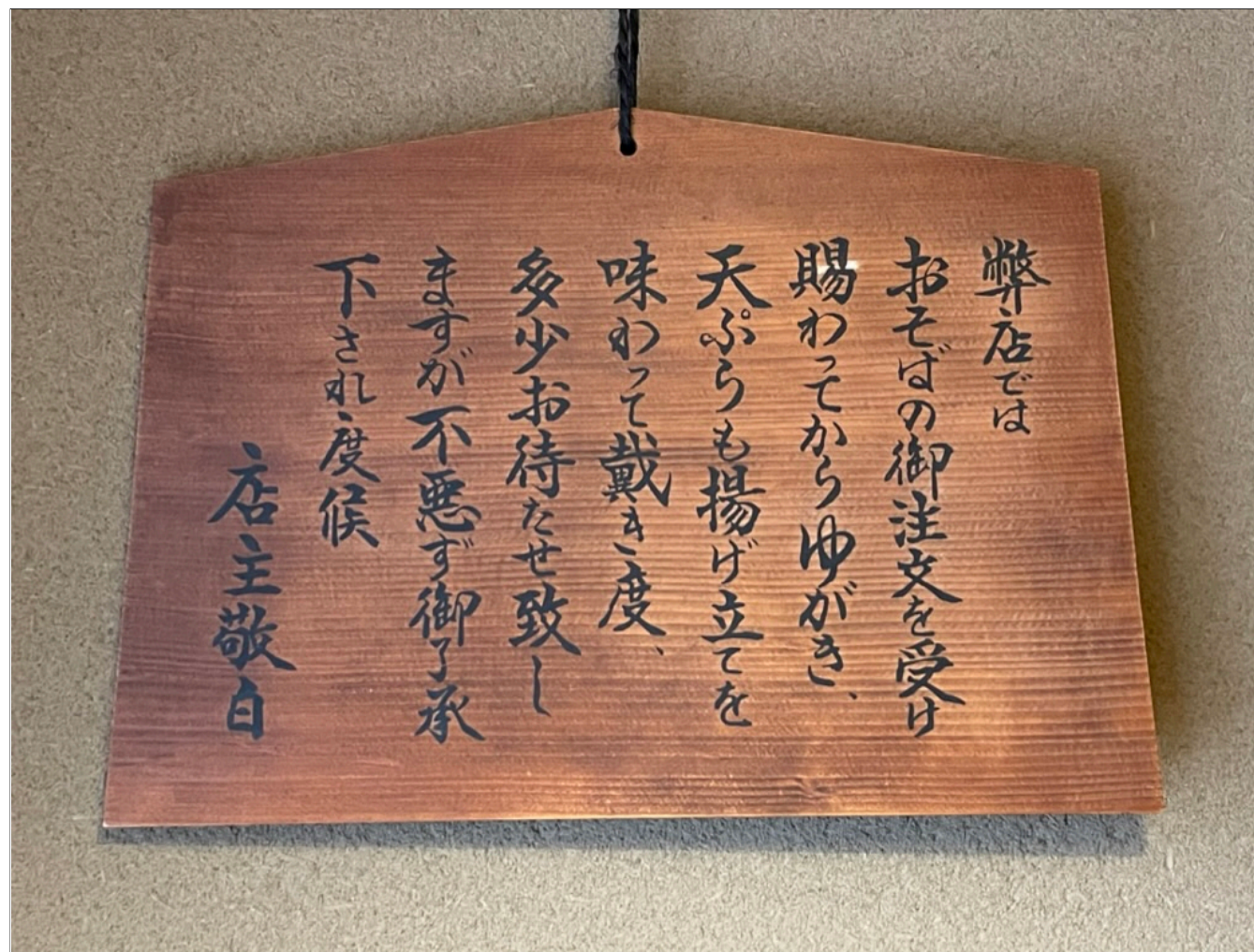
見極めは難しい。振り返って結論めいたことを言うのは簡単なのですが、走りながら、どれは捨てることができ、何が引き継がれてゆくべき本質か、見分けるのが難しいのです。古いと思われた特徴が実は重要だったり、リバイバルしたりします（自動車の座席、テンキー、巻物）。その時その時で考え続けるしかない。

デジタルテキストの組版

古くて新しい流儀

- ・ ラグ組み、空白行によるパラグラフ区切り
- ・ 文節や単語での行折り返し
- ・ 巻物の復活：冊子の優位性の喪失
- ・ 全角でない和字

パラグラフは最初は行を分けずピルクロウなどの文字を挿入することによって示されていました。モダンなタイポグラフィでは今のいわゆる日本語組版のように字下げ方式。さらにコンピューターでは今みなさんwebやメールで見る、空行によってパラグラフを分ける形に進化してきています。



弊店では

おそびの御注文を受け

賜わってからゆがき

天ぷらも揚げ立てを

味わって戴き度

多少お待たせ致し

ますが不悪ず御了承

下され度候

店主敬白

ラゲ組、単語や文節での行の折り返し

デジタルテキストの組版

新しい要素・要因

- ・ 動的挙動
- ・ 紙面のコストがないことの影響
- ・ 画面サイズの制約
- ・ 縦書き横書きを行き来できるテキストに
- ・ データとしての正しさ、書記技術

デジタルテキストにフォーカスした “日本語組版の要件” JLReq-d

- ・印刷組版をWebに適用しようとして突き当たっている問題を解決する
- ・捉え方を世に示すことで未来への一歩にする
- ・おそらく世界で初めて