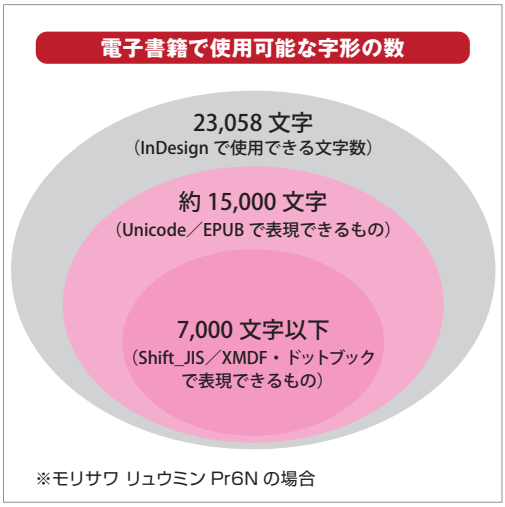
DTPデータから電子書籍を制作する際の「外字」問題

（株）三陽社　メディア開発室　田嶋　淳
ブログ「電書魂」運営中　　http://densyodamasii.com/

印刷会社の電子書籍制作の担当者で，電書魂というブログを運営しています。印刷用の DTP データから電子書籍を制作するにあたっての外字や文字化けの問題についてお話しします。「外字」というのは相対的なものですが，以下の資料で取り上げられている「外字」とは，印刷物では使用できるのに電子書籍では使用できない文字，より専門的には Adobe-Japan1 シリーズという印刷用の文字規格に含まれていて，Unicode では現状まだ使えなかったり，あるいは変換が必要になったりする文字のことを指します。

電子書籍で使用できる文字数は，印刷物のそれよりかなり少ない



現在日本語の印刷データの制作には，（通常に市販されているフォントを使用した場合）最大 23,058 文字を使用することができます。

それに対して，EPUB3 で日本語の表現のために使える文字数（UTF-8）は約 15,000 文字で，約 8,000 文字の差があります。また，XMDF やドットブックで使用できる文字数はさらに半分以下の約 7,000 文字です。この文字数の差が，外字の問題の根底にあります。

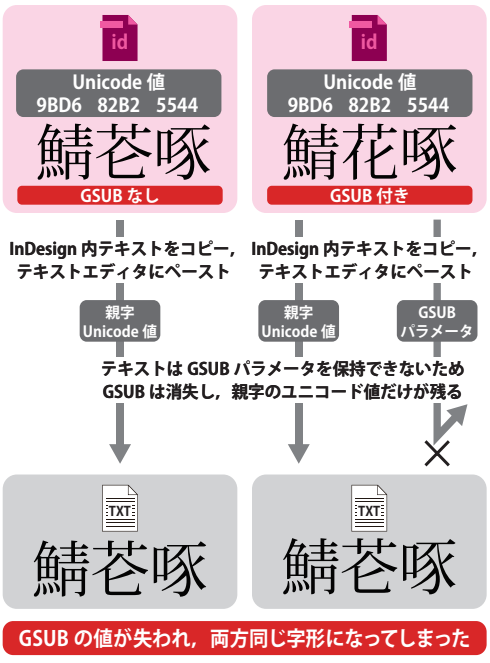
この文字数の差のうち，漢字の字形差（異体字表現）に基づく部分を吸収するために期待されているのが，今回のテーマである Unicode IVS です。

GSUB の情報をテキストが保持できないため字形が変化する

InDesign は，ドキュメント内部で Unicode の親字と異なる字形の文字を表示するために，Opentype フォントの GSUB（Glyph Substitution／字形置換）という仕組みを利用しています。

ただ，この Opentype フォントの GSUB の情報をテキストデータは保持することができません。従って，GSUB による字形置換が行われた InDesign の文字をコピーし，テキストエディタにペーストした場合，GSUB の情報は消え，親字の情報（Unicode 値）だけが残ります。この結果，字形が変わってしまいます（以降この資料では，InDesign が保持している Unicode 文字を「親字」と表記します）。

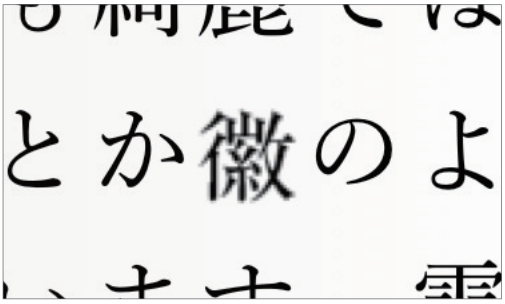
なおこれはコピー＆ペーストだけの話でなく，InDesign からの XML 書き出しや EPUB 書き出しでも同じです。



外字画像は必ずしもきれいではない

もとの印刷物の字形を電子書籍でもそのまま再現するためには，現状，文字を外字画像にすることが必要になります。ただ，外字画像は必ずしもきれいではありません。右はXMDF の外字の例ですが，あまりきれいとは言えません。

EPUB ではこれよりはかなり改善されますが，画面を黒バックにした際に外字部分が白く残ってしまったり，フォントを変えた際に外字画像の箇所だけ表示が変わらないといった問題は残ります。



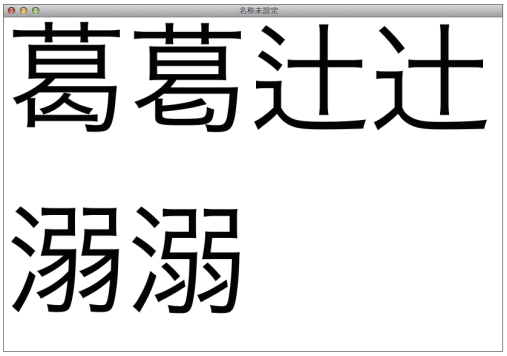
▲ XMDF の外字の例

このあたりの問題は，現状まだビューア側の対応やフォントライセンスの問題で現実的には使用できない，SVG フォントや WOFF フォントといった EPUB 内への埋め込みフォントが使用できるようになっても完全にクリアにはなりません。

結局根本的に漢字の字形差（異体字）の表現の問題を解決するためには，IVS の普及による文字数の拡張（外字の内字化）が必要になってきます。

▲外字部分が白く残ってしまう

InDesign は現時点ですでに IVS の表示そのものには対応している



▲テキストエディットで IVS 異体字を表示

InDesign は CS4 以降ですでに IVS の異体字の表示そのものには対応しています。IVS の情報を含んだテキストを InDesign にコピー＆ペーストして IVS 対応のフォントに切り替えれば，きちんと異体字が表示されます。ただし，IVS のテキストそのものを気軽に入力できる環境はまだ整っておらず，実際に IVS の字形変化を確認するためには Mac OS X 10.6 ／ Windows Vista 以上のオペレーティングシステム，対応フォント，アプリケーションが必要になります。Mac OS X 10.7 では，標準で付属する「テキストエディット」が IVS に対応しています。

InDesign には GSUB → IVS の異体字変換の機能がない

異体字を IVS で表現していれば，InDesign のデータを電子書籍にするような場合でも漢字の字形の違いは保てますが，過去に作られたデータは GSUB で異体字を表現しており，InDesign には GSUB で表現された異体字を IVS の異体字に変換する機能はありません。新書一冊で約 10 万文字あるため，これを手作業で置き換えるのは，例え IVS を簡単に入力できるインターフェースがあったとしても現実的な作業量ではありません。

そのため，もし本当に印刷物の字形をそのまま保って電子書籍にすることが必要なら，IVS そのものの普及と並行して，InDesign 内の GSUB の異体字を IVS の異体字に簡単に変換するためのツール・プラグインが必要になってきます。これはおそらく制作会社 1 社の努力で内部開発できるような規模のものではなく，専門の開発チームが時間をかけて開発しなければならないような大がかりなプロジェクトになることが予想されます。また，もしそういったものが必要になるのであれば，1 社が開発して開発した会社だけが使うという形ではなく，業界全体でコンセンサスを取って開発し，使用する形が望まれます。

従って，まずはできるだけ多くの方に外字の問題を知っていただき，こうしたツールが本当に必要なのかどうか，電子書籍の時代に「文字・字形」をどうするべきなのか，オープンな話し合いが必要と考えます。出版社をはじめとしたさまざまな立場の方の積極的な議論への参加を心から望みます。

GSUB フィーチャー消失による字形変化の実例

字形変化の実例①　合字

合字には，親字が複数の文字で構成されているものと，親字が 1 文字で Unicode に割り当てられているものの 2 パターンがあります。

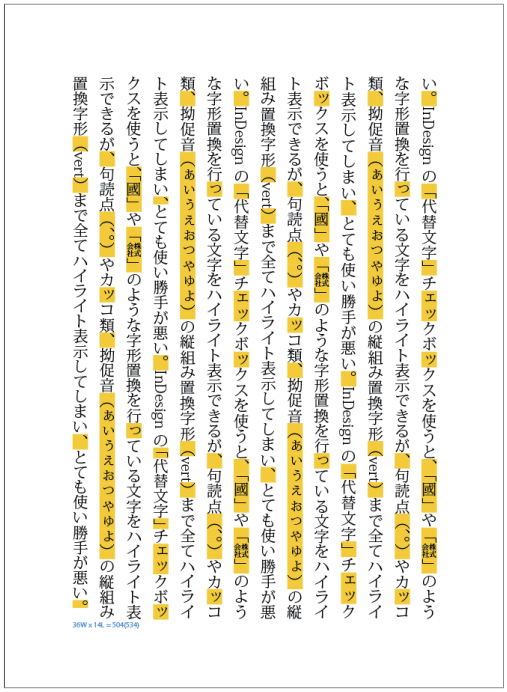
親字が複数の文字で構成されている「㍿」「㏍」のような文字の場合は，テキストにコピー＆ペーストした際，必ず複数の文字に変化してしまいます。

一方，親字が Unicode に割り当てがある「♨」「㈱」のような文字の場合，InDesign への入力方法によって，テキストにコピー＆ペーストした際に字形変化を起こすかどうかが変わります。

InDesign の字形パレットをダブルクリックして入力した場合や，ことえりのような日本語入力システムで変換して入力した場合には，その文字そのものが親字として入力されるため，テキストにコピー＆ペーストしても字形変化を起こしません。しかし，まず「温泉」「株式会社」と InDesign 内に入力してから文字パレットの「任意の合字」を用いて合字に変換した場合，親字は「温泉」「株式会社」のままなので字形変化を起こしてしまいます。

どちらの方法で入力された合字なのかは，InDesign 環境設定の「代替字形」チェックボックスをオンにすることで字形置換が行われている文字がハイライト表示され，見分けることができますが，この機能では縦書き時の句読点，括弧類，拗促音等の自動置換グリフ（vert）まですべて同一条件でハイライト表示されてしまうため，実際に対処が必要な文字が大量のハイライト文字の中に紛れてしまうことが厄介な問題です。

▲親字が Unicode に割り当てがある合字


▲句読点などの縦書き字形までハイライトしてしまう

字形変化の実例②　「旧字体」／「エキスパート字形」／「印刷標準字形」　他

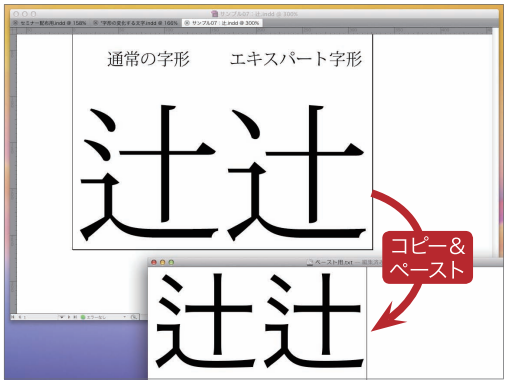
▲旧字体「傳」の字形変化の例

いわゆる「異体字」で，IVS の普及で外字ではなくなることを期待されている部分の文字です。

旧字体を中心に「傳」など，独自に Unicode のコードポイントを割り振られている文字もあり，こういった文字は通常コピー＆ペーストでも化けませんが，合字の「♨」のように新字体の「伝」を親字として入力し，InDesign の機能により旧字体の「傳」に字形だけを切り替えることができてしまうため，コピー＆ペースト時に字形トラブルになる可能性があります。

一方で，「辻」のように通常字形とエキスパート字形の双方に同一の Unicode 値が割り当てられ，GSUB の付加属性のみで字形を切り替えている文字も存在し，こうした文字はテキストにコピー＆ペーストすると GSUB の情報が失われ，同じ字形になってしまいます。

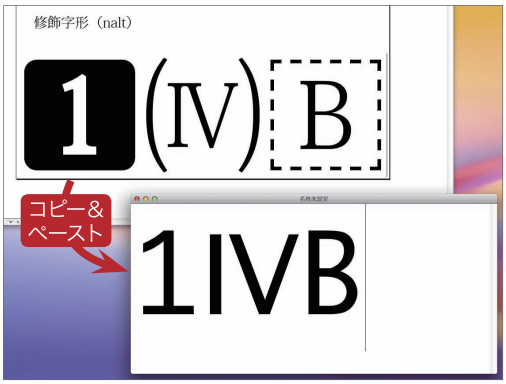
また，この文字群は InDesign の機能で全文に対して適用することができてしまい，段落スタイルや文字スタイルを用いて GSUB 情報を付加することもできるため，問題が広範囲に及びます。特に印刷標準字形などはこうした方法で全文に適用しているドキュメントが多数あるものと推測されます。



▲エキスパート字形「辻」の字形変化の例

字形変化の実例③　「すべての異体字（aalt）」／「修飾字形（nalt）」

▲すべての異体字の字形変化の例



▲修飾字形の字形変化の例

こちらも「異体字」ですが，InDesign のインターフェースで全文を選んで適用はできず，字形パレットからのみ入力できます。ひとつの親字に多数のすべての異体字／修飾字形が割り当てられていることがあり，枝番号で区別されています。

「修飾字形」は囲み文字や括弧付き文字がほぼ全てですが，「すべての異体字」は漢字の異体字や合字など多種多様です。IVS で表現できるのは，このすべての異体字のうち漢字の部分です。

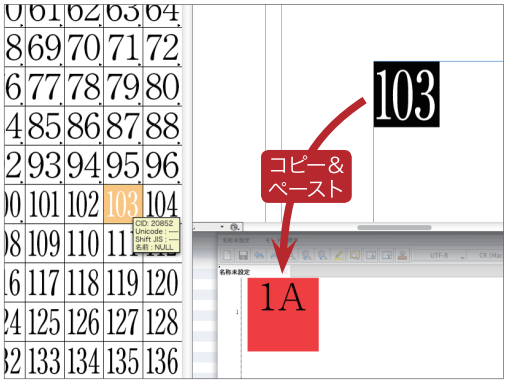
また，字形パレットから入力すると親字が変更され，それが必ずしも適切な文字になるわけではないことも問題です。代表的な例として，「〼」（CID12113）の親字が「!!」（U+203C）になってしまうパターンなどがあります。

同様に漢字の例としては「煙」（CID13657）の親字が「烟」（U+241C6）になるもの（「煙」の方がよい），「鉛」（CID13659）の親字が「鉱」（U+9206）になるもの（「鉛」の方がよい），「耕」（CID13770）の親字が「畊」（U+754A）になるもの（「耕」の方がよい），「飾」（CID13844）の親字が「𩰫」（U+991D）になるもの（「飾」の方がよい）などがあります。

字形変化の実例④　CID/GID しかない文字

Adobe-Japan1 には，GSUB でも表現できない文字（グリフ）が存在します。これらは，InDesign など対応アプリケーション内部からのみ呼び出すことができます。囲み数字や丸数字などを中心に，約 700 文字あるようです（Pr6N の場合）。

Unicode にも Shift_JIS にもコード割り当てが存在しない文字のため，確実に外字画像にする必要があります。InDesign 内でこれらの文字を選択し，テキストエディタ等にコピー＆ペーストすると「1A」という文字に化けます。これは InDesign が便宜的に親字としていた制御文字の「U+001A」が表示されたものです。制御文字ですので，テキストエディタの種類によってはこうした表示にならず，完全に消えたように見えることもあります。



▲ CID/GID 番号だけが割り振られている文字

その他の化ける文字

箇条書きリスト
スモールキャップス・オールキャップス
SING グリフレット
市販外字フォント（ビブロスなど）
独自作成外字フォント
インライン外字画像

この資料で詳述はしませんが，これらの文字類も字形変化を起こしてしまいます。
こちらに関して詳しくは私のブログ「電書魂」の記事
http://densyodamasii.com/ 電子書籍で外字を使うということについて /
をご参照ください。PDF/ 動画もあります。

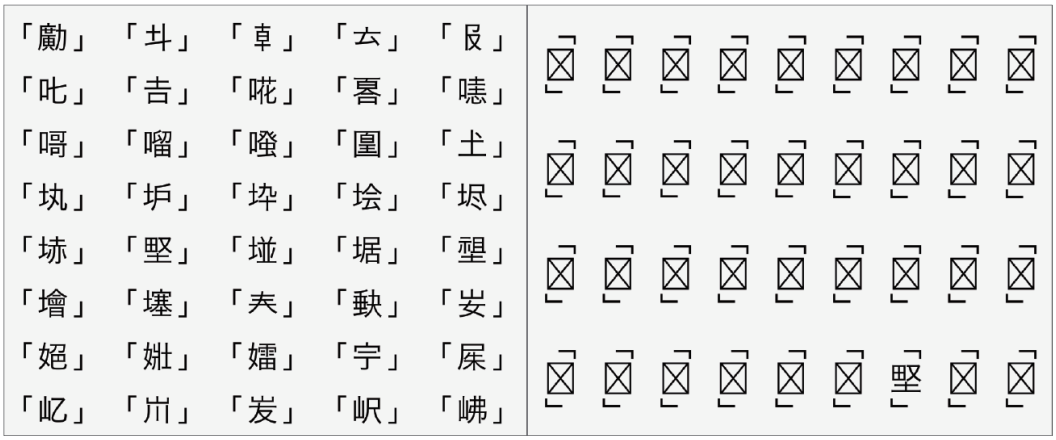
また，こちらにアップされているのと同じ PDF を出版デジタル機構ホームページ内
技術部だより：編集者が知っておきたい「電子書籍の文字化け」
http://www.pubridge.jp/info/20120611t/
からダウンロードしていただくこともできます。

環境によって字形が変化する可能性がある文字

環境によって化ける可能性がある文字①　サロゲートペア領域の文字

こちらは Adobe-Japan1 の GSUB フィーチャー消失に起因する字形変化の話ではなく，Unicode に含まれている文字が環境によって違って表示されることがあるという話になりますが，字形トラブルということでは同じですので少し触れておきます。

こちらは Unicode で「CJK 統合漢字拡張 B」「CJK 互換漢字補助」というような部分に分類されている文字群ですが，内部的に 2 つの文字コードの組み合わせでひとつの文字を表現しています。ATOK などで確認できる Unicode のコード表では，「2XXXX」と先頭に 2 がついて 5 桁になっている文字がこれにあたります。
「EPUB 日本語基準研究グループ」の Web ページ（http://www.epubjp.com/）からダウンロード可能な「EPUB3 日本語ベーシック基準 v1.0」の 28 ページにこれに関しての記述があります。実際に iBooks など多数のブックリーダの描画エンジンとして使用されている Webkit の縦書き時に，現状（原稿執筆時点の 2012/7/12 現在）まだ正常に文字が表示されません。また，おそらく Android の環境でも問題が出るものと思われます。
この文字群の大半は使用頻度のとても低い文字ですが，例外的に「叱」（U+20B9F ／新常用漢字）と「吉」（U+20BB7 ／つちよし）についてはそれなりに使われている文字のため，注意が必要です。



▲ iBooks でのサロゲートペア領域文字の縦書き時文字化け例（左：横書き／右：縦書き）

環境によって化ける可能性がある文字②　フォントのバージョンによる変化

過去に印刷データ制作時に発生していたのと同様の，フォントの字形差による字形変化の問題も存在します。具体的には，小塚ゴシック Pro やリュウミン Pr5/Pr6 など，JIS90 字形を基準としたフォントで作られた印刷データをもとに電子書籍を制作し，今後一般的に使用されると思われる JIS2004 字形を基準としたフォントを採用した電子書籍ビューアで読んだ場合，JIS 規格の例示字形の改正に伴って字形が変化する可能性がある文字が 168 文字あります。Windows Vista のメイリオで字形が変わって一時話題になった「葛飾区」と「葛城市」の「葛」の字などが有名な例です。なおこの変化は「字形が変わる可能性がある」であって「必ず字形が変わる」わけではありません。各フォントベンダーの方針によって変化する文字に幅があります。

電子書籍では読者の環境によってフォントや組版が変わる

読者の環境によってフォントや組版，コンテンツの表示色などさまざまな要因が変化し得るのが電子書籍です。最終出力が紙に固定されており，制作側でほぼ 100%コントロール可能な紙印刷物とはそこが決定的に異なります。

現状，電子書籍の閲覧環境は Web に比べても非常に多様であり，タブレットの画面サイズひとつにしても 5.5 インチから大きいもので 13.3 インチなどという幅があります。また，縦横比も 16：9 や 4：3 など統一されておらず，基準になる組版サイズといったようなものが存在していません。従って，特定のデバイスに配信先を限定しない限り，紙書籍と同様のすみずみまでの組版のコントロールはリフロー型の電子書籍では不可能です。

また，電子書籍には Web と同様にすべてを出版サイドでコントロールすることが嫌がられる部分があるようにも思います。これは，目の不自由な方，縦横の読字方向に障害をかかえる方など，多様な読者のさまざまなニーズにワンソースで応えられることを求められているのがリフロー型電子書籍であるからです。

フォントの話に戻りますと，現状日本語電子書籍の本文フォントはビューア環境に依存している形と思います。同じ電子書籍でも書店次第で本文のフォントは変わります。従って，本文が明朝かゴシックかというレベルのところまで最後はビューア依存になるのが現状で，大抵のビューアでは明朝ゴシックレベルの指定は可能なものの，印刷のように数百種類のフォントからよりどりみどりで選べるような状況ではありません。

おわりに

字形の非互換問題は 印刷データ→電子書籍 の制作時に出る問題の「氷山の一角」

字形の非互換問題は印刷データからの電子書籍制作の「氷山の一角」です。印刷データはその中身が非常に多種多様で，成果物がほとんど同じに見えたとしても，各制作会社，印刷会社ごとに制作方法が異なるのが実態であるように思います。それを許す形で機能拡張を繰り返してきたのが InDesign のような DTP 組版ソフトです。

それ自体は印刷物をより良い形で制作するための各社の努力のありようであり，何ら責められるべきものではありませんが，そうした各社各様の印刷データの作り込みが，電子書籍化などの目的でコンテンツを二次利用する際に大きな障害になってくるのもまた，事実です。

自動変換を妨げる外字以外の要素を簡単に列挙するだけでも
「特色」の問題，
「CMYK → RGB の色変換」の問題，
「変則レイアウトの書籍の変換」問題，
「インラインの表の変換問題」
など，さまざまなファクターが考えられます。

結局，印刷データからの電子書籍制作は近道ではない

このようなさまざまなファクターにより，印刷用 DTP データからの電子書籍制作には必然的に各所で手作業による修正が発生し，結果的に高コストなものにならざるを得ません。

各印刷会社，出版社に蓄積されている過去の印刷データからの電子書籍の制作および，今後しばらくの間発生してくると思われる紙書籍流用型の電子書籍のデータは印刷用データを元にして制作するしかありませんが，将来的には上流にマスターデータ（XML）を蓄積して印刷データと電子書籍データを自動変換出力する形でのワークフロー構築を目指すべきと考えています。

これは，将来的に出てくると考えられる印刷データに先行して電子書籍を作る動きや，印刷データと電子書籍の構成を大幅に変えて出すような動きに柔軟に対応するためにも必須の流れと思います。

印刷データをハブにした電子書籍制作は，特定メーカーの DTP 組版ソフトの仕様に制約されて，ゼロから電子書籍制作をしていれば特に苦にならなかったはずのそういった多局面の柔軟な展開を難しくしてしまいます。

長期にわたって販売される書籍の商品特性からも，マスターデータが必要になる

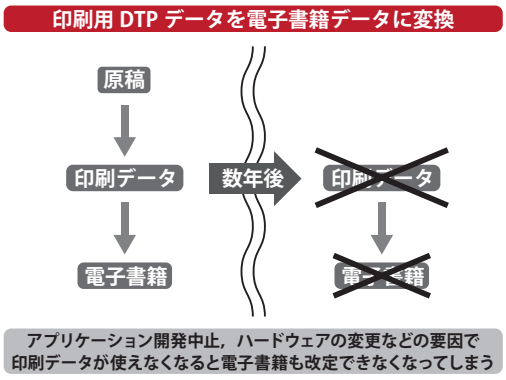
印刷用 DTP データは実はかなり賞味期限が短く，InDesign などの DTP 制作アプリケーションのバージョンアップの早さ，フォントへの依存性，ハードウェア側の世代交代の早さなどさまざまな要因が絡み合った結果として，4 ～ 5 年前の DTP データがもう読めなくなりかかっているなどという話があります。

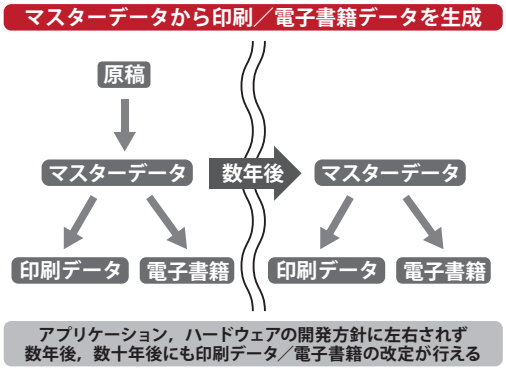
EPUB3 など電子書籍のフォーマットもまた，過去のさまざまな電子フォーマットの世代交代の歴史から考えて，4 ～ 5 年後に EPUB4 や 5 に世代交代している可能性も否定できません。デジタル，IT 産業にはそうしたダイナミズム，スピード感があります。

しかし，書籍はそうした早い流れとは相反する部分を持つ商品でもあります。私の会社で制作しているような専門書・学術書では，10 年以上前の書籍の再版修正などという依頼が普通に発生します。古いものでは，活版印刷の時代に制作された書籍の文字修正がまだ発生するような世界です。

そうした長い商品寿命を持つ書籍が存在することや，図書館でのデータの蓄積，活用などを視野に入れた場合，書籍のデータは特定メーカーの製品開発・販売方針によってデータの可読性が左右されないオープンフォーマットの XML などで蓄積しておき，それを各時代に合わせた最新の配信フォーマットに適宜変換して提供する形が将来的には望ましいと考えます。これは必ずしも電子書籍だけの話ではなく，印刷用データも今後オンデマンド印刷の流れに結び付いて，そうした方向に向かうべきものと思います。

もしこうした書籍の保存・蓄積を考えず，4 ～ 5 年でメンテナンス不可能になる形で電子書籍を量産してしまった場合，最悪の場合 50 年後，100 年後から見た場合に「文化の真空地帯」が発生することにもなりかねません。





カタログ組版などではすでに当たり前になっているこうしたデータ制作の自動化の流れが，文芸書のような一般書籍の制作でこれまで定着しなかった要因のひとつに，今回のテーマの字形の問題もあったように思います。Unicode IVS がその解決の一助になることに期待しています。

参考リンク

IVS 技術促進協議会	http://ivstpc.jp/
Wikipedia「Unicode」	http://ja.wikipedia.org/wiki/Unicode
Wikipedia「外字」	http://ja.wikipedia.org/wiki/ 外字
Wikipedia「異体字セレクタ」	http://ja.wikipedia.org/wiki/ 異体字セレクタ
Adobe-Japan1-6 文字コレクションに対応する 日本語 OpenType® フォントについて（アドビシステムズ）	http://www.adobe.com/jp/support/type/aj1-6.html
フォントの基礎知識（モリサワ）	http://www.morisawa.co.jp/font/about/knowledge/index.html
しろもじ作業室	http://shiromoji.net/
SVG フォントを使った外字表現（W3C 藤沢 淳）	http://www.w3.org/Style/Japan-2011/fujisawa-gaiji-SVG.pdf
Wikipedia「合字」	http://ja.wikipedia.org/wiki/ 合字
Wikipedia「字体」	http://ja.wikipedia.org/wiki/ 字体
JIS2004 と JIS90 との 文字の形の対照表（アドビシステムズ）	http://www.adobe.com/jp/support/winvista/pdfs/JIS2004_Comparison.pdf
電子書籍の（なかなか）明けない夜明け　第 6 回 電子書籍時代の外字問題を探る（1）（小形克宏）	http://internet.watch.impress.co.jp/docs/column/yoake/20110708_458937.html
電子書籍の（なかなか）明けない夜明け　第 7 回 電子書籍時代の外字問題を探る（2）（小形克宏）	http://internet.watch.impress.co.jp/docs/column/yoake/20110715_460703.html
電子書籍の（なかなか）明けない夜明け　第 8 回 電子書籍時代の外字問題を探る（3）（小形克宏）	http://internet.watch.impress.co.jp/docs/column/yoake/20110722_462158.html

お世話になった方々

市川せうぞーさん（ShowTime +one）…… http://www.seuzo.jp/st/index.html
小形克宏さん（もじのなまえ）…… http://d.hatena.ne.jp/ogwata/
直井靖さん（Mac OS X の文字コード問題に関するメモ）…… http://d.hatena.ne.jp/NAOI/
丸山邦朋さん（ものかの）…… http://tama-san.com
深沢英次さん（出版デジタル機構）…… http://www.pubridge.jp/

今回の内容について，小形克宏さんの執筆された技術者向けのより詳しい資料を出版デジタル機構ホームページ内「技術部だより」（http://www.pubridge.jp/info/info-cat/tech_topics/）よりご覧いただけます。併せてご参照ください。