インターネットの多様な利用者環境における多言語文書の閲覧機能について

阪口 哲男, 高 東梁, 吉川 晃生
筑波大学 大学院 図書館情報メディア研究科
〒305-8550 茨城県つくば市春日 1-2
E-mail: {saka, gao, aky}@slis.tsukuba.ac.jp

概要

インターネットの普及と共に様々な言語による情報提供やサービスの要求が高まっているが、まだまだ実際に用いることができる言語は限られている。また、利用者環境が多様化しているため、言語の問題とは別に情報アクセスの技術的な障壁も残っている。本研究では多言語対応において解決すべきと考えられる縦書きと横書きの混在の際の取り扱いの提案とそのためのライブラリの試作を行った。また、多様な利用者環境への対応の一手法として様々なデータ形式の電子図書を利用者環境に合わせて変換・提供するシステムの開発も行った。本稿ではその両者の基本的な提案内容と試作の概要について述べる。

キーワード

モンゴル語, モンゴル文字, 縦書き, WWW, HTML, 電子図書

1. はじめに

インターネットが普及するにつれて、人々が自らの母語を用いて情報の交換や提供を進めたいという要請は高まっており、またソフトウェア等によってサポートされる言語の種類も徐々に増えてきている。以前はオペレーティングシステム (OS) 等のソフトウェアは、そのサポートする言語によって異なるパッケージやバージョンを用いるのが通例であったが、最近では一つのパッケージで設定によってサポートする言語を切替えられるようにもなっている。

このように情報技術の発達と共に World Wide Web (WWW) や電子メールなどで使用可能な言語は増えてきている。しかしながら、開発効率やコストの兼ね合いなどもあるため、サポートされる言語は利用者数が多いと考えられるものや、既にサポートされている言語と似通った性質のものが中心となりがちである。その結果、利用者数がさほど多くなく、既にサポートされている言語とは性質が大きく異なる場合にはなかなかサポートされないのが実際である。

インターネットでは様々な人々が自分の都合にあった機材やソフトウェアを用いているため、新たな機能を追加しようとしても、特定の環境でのサポートは比較的容易に実現できるが、他の環境までサポート範囲を広げるのはなかなか難しい。このことは様々な言語をサポートしようという場合にも当てはまる。

著者らのグループではかつて海外で日本語の WWW ページを読めるようにしたいという要求をきっかけに様々な言語の WWW ページを様々な環境で読むことを可能にする MHTML Browser[1] の開発を行った。現在は既に一般的に利用されている環境の多言語サポートの水準が MHTML Browser 相当以上になりつつあるが、まだまだ前述のようにサポートから取り残されている言語がある。

以上のような問題を踏まえて著者らが最近進めてきた、多言語サポートをより進めるために必要な機能、そして最大限の表現能力を保ちつつ多様な利用者環境に適応して文書を提供する枠組に関して検討し、試作を行ってきた。

2. インターネットにおける多言語文書の閲覧

インターネットにおいて、情報を提供しようという場合、そこに用いることが可能な文書形式には様々なものがある。ここでは、マークアップ言語、特に HTML や XML についてこれまでの著者らの試みを中心に述べる。

MHTML Browser においては多言語の情報提供において、問題点を以下の 2 点に絞って開発を進めた。

- HTML 文書に用いられる文字コード
- 各文字コードで書かれた HTML 文書の表示に必要な文字フォント

90 年代後半では既に Unicode 規格の策定が行われていたが、インターネットやコンピュータで用いられている例が実際にはまだ少なく、各国で定められた個々の言語に対応した文字コードを用いるのが主流であった。例えば、日本語の場合、JIS X0201 または ASCII のいわゆる半角英数字と、JIS X0208 の第 1、第 2 水準の漢字や仮名を同時に使用する、ISO-2022-JP やシフト JIS、EUC-JP などの文字コードが主に使われていた。

MHTML Browser ではこのような各国で用いられている個々の文字コードを解釈する機能を備えており、そしてそれらの各コードに含まれる文字を表示するために必要な文字フォントの図形（グリフ）データを文書に添付することで各利用者の端末での表示を可能にしている。文字フォントを添付した文書形式は独自に定義しており、その閲覧のためには専用のビューワを必要とするが、これを JavaApplet として供

給することで特にソフトウェア等をインストールすることなしに一般的な WWW ブラウザでの閲覧を可能にしている。

このような文書データに表示に必要な文字フォントデータを付加する方式はそれほど特殊ではなく、例えば Adobe が定義・公開している PDF 形式にも備わっている (Adobe 製品では日中韓の漢字フォントを埋め込まないが、フリーソフトや他社品では埋め込むことが可能である)。

MHTML Browser では特に日中韓の仮名や漢字の表示を第一目標としており、欧州語やタイ語などもサポートしていた。文字の表示として単純に文字フォントデータを並べて表示するものであり、改行や文字の進行方向に関しては特に言語依存の処理は行わなかった。そのため、次の 2 点の問題が残された。

- 単語の途中で改行されるなど欧州語等の word wrapping に未対応
- アラビア語やヘブライ語のような文字の進行方向が異なる言語に未対応

後者についてはともかく前者の問題は対応への要望が多かったものの、MHTML Browser の基本的な原理である「言語依存の処理を行わない」という原則の範囲での対応が難しく、未対応のままとなってしまった。つまり、プレーンテキスト程度の文書表示においても、単純に文字フォントデータを並べるだけでは不十分である。

このような MHTML Browser 開発成果を受けて著者らは多言語 XML 文書表示のためのブラウザの開発を行った [2][3]。この開発では表示のための基本的な機能を設計し、実装が可能であることの確認を主たる目的としていた。言語依存の処理をサーバ (ゲートウェイ) 側で行い、クライアント (ビューワ) 側では言語独立な処理のみを行うような構成となっている。これにより、クライアント側に手を加えずに対応言語を増やすことを可能にしている。

この XML 文書ブラウザでは、日本語の禁則処理や欧州語の word wrapping に対応するために改行の可否を表す制御コードと、アラビア語などが交じった場合のための文字の進行方向を表す制御コードを導入している。元の XML 文書に対して、サーバが文字フォントを付加すると同時に言語に依存した解釈を行い、改行可否コードと進行方向のコードを文字コードの間に埋め込む。クライアントはそれらの制御コードに基づいて機械的に改行の可否と文字を表示する際の進行方向を判断するので、言語に依存した処理を行わなくて済む。

サーバにおいては言語依存の処理を Unicode のコード範囲に応じて行うようにプログラムコードで直接記述していた。そのため、サーバにおいて新たな言語への対応をするにはプログラムコードの変更を行う必要があるため、あまり容易に追加することができない。また、サーバ側での追加作業を行っても対応できない言語が存在することがわかったので、それへの対応が必要となることもわかった。その典型的な事例が縦書きの言語である。

3. 縦書きを含む多言語表示サポート の試み

現在、筑波大学図書館情報メディア研究科にはモンゴル語を母語とする留学生が複数人いる。その出身地は大別するとモンゴル国と中国の内蒙古自治区に分けられる。どちらもモンゴル語を用いているが、歴史的な経緯もあってその記述に用いる文字が異なっている。モンゴル国ではロシア語と同じキリル文字が用いられ、内蒙古自治区ではモンゴル文字が主として用いられている。

キリル文字を用いる場合にはコンピュータによる処理や表示については大きな問題とはならない。しかしながら、古くからモンゴル語に用いられていたモンゴル文字については次のような 2 つの特徴があり、コンピュータ処理・表示について工夫が必要となる。

- アラビア語などと同様に単語を構成する文字は続けて筆記し、各文字には独立形、語頭形、語中形、語尾形の 4 種類の字形がある。
- 文字を上から下へと縦書きし、行は左から右へと進む。

1 番目の特徴については、アラビア語などにも見られることであるため、対応は難しくはない。しかしながら、Unicode などでは字形の違いに関わらず同じ文字コードを付与しているため、表示や印刷の際は文字コードの並びを解釈して、単語のどの位置（語頭、中間、語尾）にあるかを判断した上で文字の字形データを選ぶ必要がある。そのアルゴリズム自体は定まったものがあるので実装が難しいわけではない。しかしながら、通常のソフトウェアは 1 文字毎の文字コードのみで字形も一意に定まることを前提としており、アラビア語などの一部の文字コード領域の部分については追加処理をするようにしている。そのため、文字フォントデータを追加するだけでは正しい表示は行われず、表示ルーチンの追加なども必要となる。

より大きな問題となるのは 2 番目の特徴である。縦書き自体は日本語や中国語でも行われるので珍しくはないが、日本語や中国語の行が右から左へと改行していくのに対して、モンゴル文字では逆に改行することになる。また、現代では日本語も中国語も英語などと同じ横書きも行われているため、コンピュータや WWW ブラウザでの表示も通常横書きになっている。これは、日本語や中国語で用いられている文字が横書きにしても文字そのものを横倒しにする必要がなく、それほど可読性が失われないので、横書きが受け入れられているためである。

一方、モンゴル文字で記述するモンゴル語（以降、単に「モンゴル語」と記す）については、文字を連ねる「続け字」であるために、縦書きであるものを横書きにしようとすると、文字ごと横倒しにする必要がある。もし、文字を正立させたまま横書きにするとすれば、文字同士の連なりも切断することになり、字形を大幅に変えることになりかねない。

以上のような点から縦書きによる表示・印刷はモンゴル語をサポートする上では必須である。ところが、縦書き機能をブラウザに単純に組み込むだけでは済まず、文書を表示する際のレイアウトが問題になる。つまり、文字を書き進める方向や改行していく方向が 90 度異なるものが混在すると、画面上での配置、そしてブラウザとして考えるとスクロール機能などにも影響が出る。

このような観点から著者らは WWW において情報提供側や利用者に大きな負担を強いることなく、横書きの言語と同様に縦書きの言語での情報提供も行える環境整備が必要であると考える。現状では Microsoft Internet Explorer (MS-IE) など一部の WWW ブラウザでは縦書きのサポートは始まっており、Cascading Style Sheets (CSS) に従って、WWW ページ内をブロック単位で縦書き表示させることが可能となっている。しかしながら、アラビア語やヘブライ語など横書きで文字の進行方向が異なるものが交じっている場合には Unicode における Bidirectional Algorithm (Bidi) のような一般的な規則は定められているが、縦書きと横書きの混在の場合には前述のレイアウトの問題もあるために、そのような一般的な規則は定められていない。その結果、縦書きと横書きの言語が混在するような場合には、CSS 等で詳細な指定をしなければ読み易い表示を行えない状況にある。

そこで、本研究では縦書きと横書きが交じりあっている文字列を表示する際の一般的な規則を定め、それに従って自動的に表示ブロック内の文字表示を行うことを提案する。具体的に規則を定めるためには、様々な条件を精査し、検討を重ねる必要があると考えられるが、本研究ではその大要として以下のように定めた。

- ブロック内に表示しようとする文字列中における各書字方向の文字の比率を調べ、最も高い比率のものによってブロック全体の書字方向を決める。

- ブロック全体の書字方向と異なる書字方向の文字列については、その文字列の長さが可読性を妨げない程度に短ければ本来の書字方向で、そうでなければブロックの書字方向に合わせて表示する。その際、文字列の長さは改行可能な位置で分割した個々の部分列（通常は単語）毎に調べる。

このように定めた規則に従った表示が可能となった場合、どの程度アプリケーション等の開発が楽になるかを確かめるために、今回は JavaServlet 用のクラスライブラリの試作を行った。本ライブラリは WWW ブラウザとして MS-IE を想定しており、表示しようとする文字列を走査し、適切な表示を行えるように CSS 記述を生成する。この結果を JavaServlet の出力とすれば、MS-IE 上では縦書きと横書きが混在した表示が行われる。

今回試作したライブラリでは縦書きとしてモンゴル語の表示を取り扱っている。前述のようにモンゴル語の表示には 2 つの問題があるが、今回は Unicode とは異なる各字形毎に独立した文字コードを付与した独自のコード体系に基づく文字フォント [4] をクライアント PC にインストールし、本ライブラリでその文字フォントを使用する CSS 記述と HTML における文字参照記述を生成することによって文字表示を可能とした。なお、モンゴル文字の入力環境は一般的ではないので、暫定的に独自のタグ<mongolian>で囲みローマ字で記述した部分をモンゴル文字として扱っている。

書字方向についても現在の MS-IE では日本語や中国語のための「tb-rl」つまり、右から左へと改行していく方式には対応しているが、モンゴル語に用いられる「tb-lr」には未対応である。そのため、本ライブラリでは表示領域を仮定して各行毎の小さな表示ブロックに分割して、それを左から右に並べることによって代用している。

以上のような機能を準備することによって図 1 のように縦横混在の表示が可能となった。

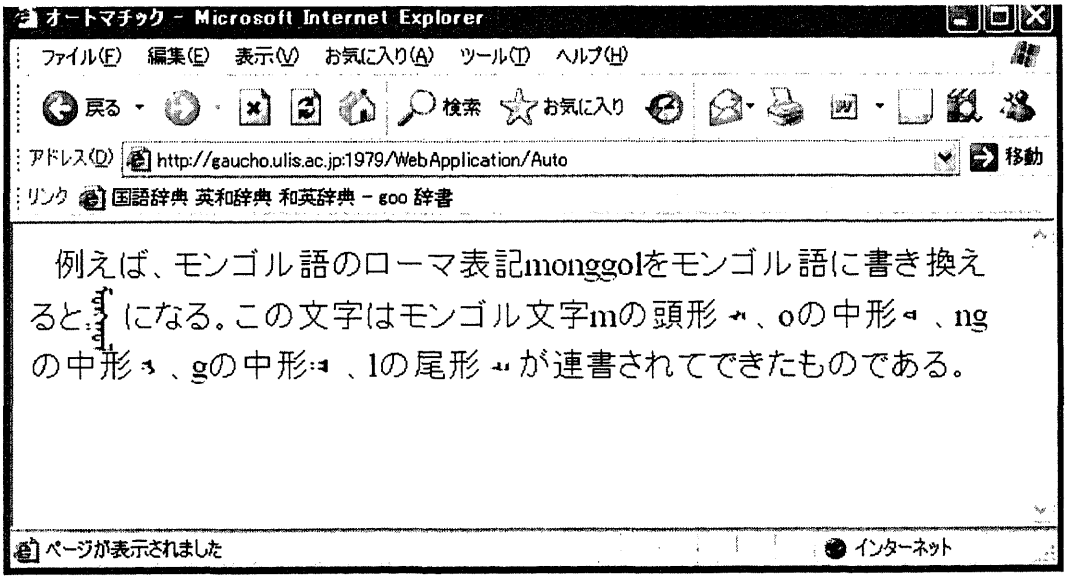

図 1 モンゴル語を含む縦書きと横書きが混在したページの表示例

4. 電子図書を対象とした多様な利用者環境への適応の試み

本節では著者らが特定の領域に限定してではあるが、現在のインターネットにおいて情報提供やサービス提供を行う際に問題となる、多様な利用者環境への適応手法に関する試みについて述べる。

前節では多言語文書表示における縦書きと横書きが混在する場合の解決案とそのための試作ライブラリについて述べた。そのライブラリではブラウザを Windows 上の MS-IE と仮定しており、他の環境では正しく表示されないことがある。これは何も多言語サポートに限らず、HTML タグや CSS 記述の解釈の違いなどによって日常的に生じている。例えば、最近は Ajax と呼ばれるブラウザの JavaScript 処理機能を活用して対話機能を高めたサービスが注目を集めているが、この JavaScript についても環境による違いが問題となり易く、WWW ベースのシステム開発者を悩ませ、開発効率を落す原因ともなっている。

このような同一規格・形式に則っているにも関わらず互換性問題が出る一方で、そもそもデータ形式そのものが多様化しているものもある。電子図書はその典型であり、ビジネスモデルの影響もあるが、業界団体などにより様々なデータ形式が定義され、電子図書提供に使用されている。それぞれのデータ形式の図書を読むためには、それに対応した閲覧ツールを用いる必要があり、利用者の手を煩わせるほか、利用者の環境ではそのようなツールを用いることができない場合もある。T-Time[5] は WWW ページなどの形式で提供される図書を読み易く表示するための独立した閲覧ツールである。また、画像データに変換することで多様な環境での閲覧を可能にする機能を備えている。しかしながら、それは利用しようとする端末の仕様に合わせなければ読み易さが損なわれてしまう。

このようにデータ形式の多様性やその互換性の問題に加えて、閲覧に用いる端末装置の仕様の影響も無視できない場合がある。特に携帯型の端末では物理的制約が強いため、その影響は大きい。例えば、ディスプレイの解像度や物理的な表面積、閲覧操作に用いるためのキーボードなどの入力デバイスなどによって制約を受ける。最近の携帯電話では C-HTML や HDML などの携帯端末向けのマークアップ言語に対応しており、それらの言語はそれぞれで規格が定められているが、実際に使用する携帯電話のディスプレイなどの仕様が読み易さに大きく影響する。

現状では利用者が自分の使っている閲覧環境に合わせて電子図書購入の際に適切なデータ形式等を選ぶ必要がある。著者らはそのような手間を省き、かつコンテンツ提供者が利用者環境毎に合わせて電子図書を準備しなくても、自動的に変換することができれば、電子図書閲覧をより気軽にできるのではないかと考え、そのシステムの試作を行った。

基本的には入力となる電子図書をまず共通の中間形式に変換し、そして中間形式のものを利用者の環境に合わせて変換する。その際、中間形式の表現能力が貧弱であると入力電子図書や利用者環境の特性を十分に活かせなくなる。そこで、中間形式の必須機能として文書構造を表すものとし、端末などの今後の技術開発等によって向上していくと考えられる文字装飾などの表示スタイル指定等については拡張部分として、その記述文法を定義しておく。これによって将来もその拡張記述を行うことによって表現力を補うことが可能となる。今回のシステムでは中間形式として JepaX[6] を採用し、それに元々備わっているクラス・タイプ定義記述について形式的な記述を導入することにした。

入力側の変換器と出力側の変換器はそれぞれ入力電子図書の形式と利用者環境によって選択されるが、データ形式等は共通でも詳細な機能の使い分けや対応状況の違う複数の候補があることが考えられる。その際は拡張記述で互換性のあるものが多いものを優先して選ぶことになる。

試作システムの構成は図 2 のようになる。試作システムは HTTP によって利用者端末と通信するものとしており、利用者環境情報はその HTTP リクエストに含まれるヘッダ情報によって得るものとする。しかしながら、リクエストに含まれるヘッダでは情報が不十分な場合があること、読み易さについては単純にソフトウェアやハードウェアだけでなく、利用者の好みの反映も必要なことから、利用者環境のパラメータは利用者自身によって修正可能なユーザインタフェースも準備している。

利用者環境のパラメータ、例えばディスプレイの大きさなどは HTTP リクエストには含まれていないことが普通であるが、携帯電話のような場合、リクエスト中に機種名などが含まれている場合が多く、その機種の仕様等をあらかじめ登録しておき、機種名等を手がかりに検索、使用する。しかしながら、この手

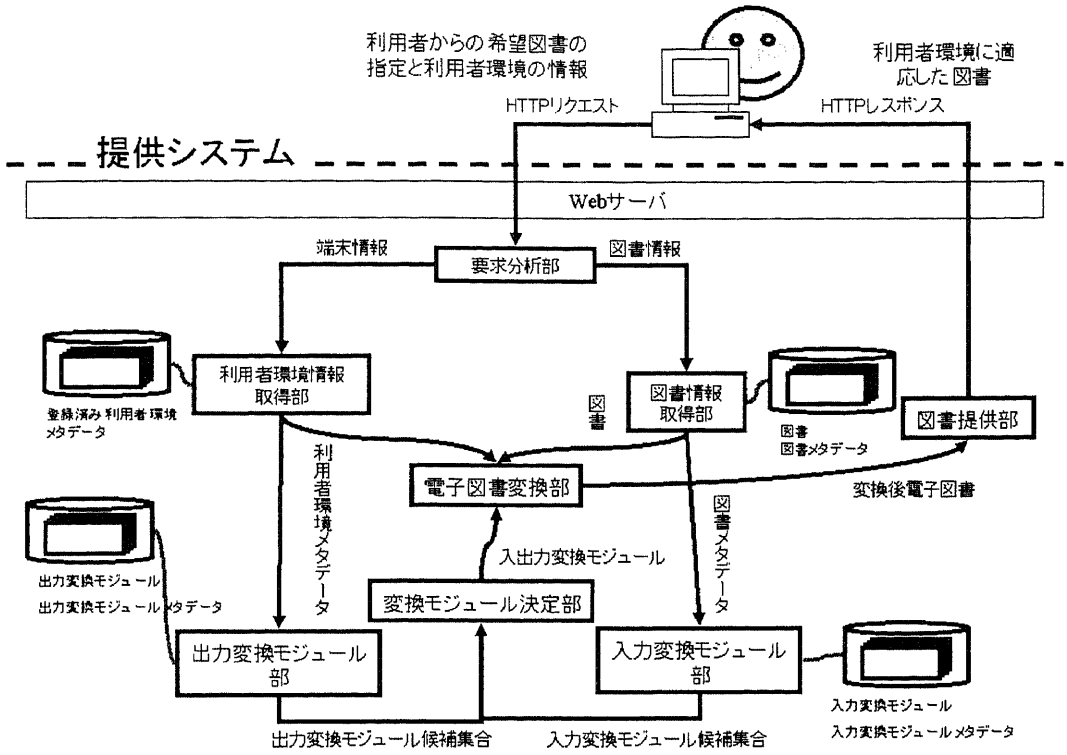

図 2 利用者環境適応型の電子図書データ形式変換システムの構成

法では限界があり、特に PC のような柔軟性の高い環境では不十分であるので、本システムの手法を実用化するには HTTP リクエストなどによる利用者環境情報の送り方などを規格化・普及させる必要がある。

5. おわりに

本稿ではこれまで著者らが進めてきたインターネットにおける多言語文書の表示に関する問題と多様な利用者環境の問題に対する最近進めてきた提案と手法について述べてきた。多言語対応については片岡らの研究 [7] のように究極には全ての言語の特性を調べあげ、それらに対応する機能を組み込むと言う手法も考えられる。そのような手法にも利点は多いが、それを組み込むにはコストがかかるので、マイナーな言語に対する対応は後回しになりがちである。ある程度の妥協はしても、低コストでマイナーな言語をそれなりの品質で取り扱えるような機構を準備することも必要だと思われる。

また、多様な利用者環境への対応という問題はインターネットでは本質的なものであると考えられる。その一方で、これはビジネスも絡むために解決は簡単ではないが、図書館と同様に様々な利用者が気軽に情報をアクセスするためにも可能な限り技術的な障壁を取り除くことが重要ではないかと思われる。

なお、本研究は日本学術振興会科学研究補助金基盤研究 C(課題番号 16500049) の助成を受けて行った。

参考文献

[1] 前田亮, Myriam Dartois, 太田純, 藤田岳久, 阪口哲男, 杉本重雄, 田畑孝一. クライアントにフォントを必要としない多言語 HTML 文書ブラウジングシステム. 情報処理学会論文誌, Vol.39, No.3, pp.802-809.

1998.

[2] 阪口哲男, 永森光晴, 杉本重雄, 田畑孝一. 多様な利用者環境で多言語文書表示を可能にする XML ブラウザ. ディジタル図書館, No.19. 2000. URL: http://dl-lab.aist-nara.ac.jp/DLjournal/No_19/3-saka/3-saka.html (2006/2/16 参照)

[3] 阪口哲男, 樋爪育恵, 加藤大博. メタデータブラウザの多言語対応に向けた課題への取り組み. 情報知識学会第 11 回 (2003 年度) 研究報告会講演論文集, pp.13-16. 2003.

[4] CMs / Classical Mongolian script / TrueType fonts.
URL: http://www.geocities.com/geseree/cms_gallery.html (2005/2/16 参照)

[5] 株式会社ボイジャー. T-Time. URL: http://www.voyager.co.jp/T-Time/index.html (2006/2/16 参照)

[6] 日本電子出版協会. JepaX. URL: http://www.jepax.org/ (2006/2/16 参照)

[7] 片岡裕, 片岡朋子, 上園一知, 大黒谷秀治郎, 大矢俊夫, 小原啓義. 全世界の文字と言語の完全混在処理環境: Internationalized Multilingual System - The Waseda I18N & ML System. ディジタル図書館, No.6. 1996. URL: http://www.dl.slis.tsukuba.ac.jp/DLjournal/No_6/kataoka/kataoka.html (2006/2/16 参照)