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

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**
* 早稲田大学 情報科学研究教育センター
〒169 新宿区大久保 4-3-1 理工学部情報学科小原研究室
** 早稲田大学 理工学部 情報学科
〒169 新宿区大久保 4-3-1 理工学部情報学科小原研究室
Tel: 03-3232-0551, Fax: 03-3232-0551
E-mail: kataoka@ohara.info.waseda.ac.jp

概要

国際化した計算機環境にあって、既存の地域化による処理では多数の文字と言語を同時に混在処理したいという要求は全く満足されていない。ところが、文字、言語、正書法の関係が、各々の理解が困難であると言う理由で解明されず、種々の混乱と誤解を招いてきた。早稲田大学では、世界の文字、言語、正書法、文字コードを調査・分析し、文字に含まれる情報と言語情報を分離し、全世界の文字の混在処理を実現する国際化、そして言語依存情報を持つテキストを処理する多言語化の環境を、各々にハード・コーデッド部分を持たせずに構築した。本稿は、地域化の問題点と不適切な文字コード、国際化・多言語化に必須な条件を示し、早稲田国際化多言語環境について解説する。

キーワード

デジタル図書館, 地域化, 国際化, 多言語化, 自然言語処理, ＩＳＯ２０２２

Keywords

Digital Libraries, Localization, Internationalization, I18N, Multilingual, Natural Language, ISO 2022

1. はじめに

国際的なネットワークが日常的に使用され、計算機を取り巻く環境は、急速に国際化した。既に、物理的なネットワークは、広く国際的に接続され、他国へのアクセスは、極めて容易に行なえる。即ち、各国家規格や国際規格の文字コードで記述されたテキストは、国境を越えて送受が可能となっている。

また、異なる国籍を持つ、異なる言語・文字の使用者が、同一の計算機を共有することも稀ではなくなっている。特に、安価なパーソナル・コンピュータのネットワークへの参加が、計算機の利用を大きく変貌させ、語学教育、図書館データベースで、複数の異なる言語で使用される文字群の混在処理が、必須となった。

ところが、全世界の文字と言語、その正書法に関し、文字の理解が困難なことから、文字、言語、正書法は、互いに依存しあうものとされ、それぞれが、異なる情報の集合として分離可能であることが理解されて来なかった。即ち、文字の表示及びテキスト処理は常に言語に依存し、言語を指定しなければ可能ではないと考えられてきた。そのため、英語と１言語のみの混在を許すＰＯＳＩＸ Ｌｏｃａｌｅの地域化モデルが考案され、計算機の利用を大幅に制限することとなった。さらに、ＭＩＭＥ、ＳＧＭＬ、ＨＴＭＬ等においても、複数の文字コードの混在は考慮されておらず、今日の計算機に要求されている、複数文字コードの混在は、実質的に不可能である。

さらに、文字に対する理解の欠如が、ＩＳＯ１０６４６などの、全世界の人口の半数以上に使用されているペルソ・アラビア文字、デーバナーガリ文字やタイ文字などの結合音節文字に対し、正しく１文字とそれに与えるべき図形を全く決定できない文字コードを制定する原因となっている。これらの文字コードは、通信にもテキスト処理にも使用できない。

このような混乱を招いた原因は、単純に、文字に対する調査と理解不足である。早稲田大学では、全世界の文字の混在処理、即ち国際化と、言語情報に依存する処理、即ち多言語化の両者を満足する、国際化・多言語環境を作成した。本システムは、実際に全世界の文字、文字コード、正書法を調査し、それらの特徴を数値化可能な情報として抽出し、言語情報に依存せずに基本的なテキスト処理を含む国際化が可能であることを示した。それに基づいて、ＩＳＯ２０２２を全て満足し、各種国家規格と国際規格を受理し、単一の文字集合に変換し、それに正しい文字図形を与え、テキスト処理を行なうライブラリを開発した。さらに、その文字に言語情報を付加し、言語固有の情報による排他的言語依存処理をも可能とした。

本稿では、地域化の問題点を示し、国際化と多言語化に関して必須事項を解説し、次にそれらの問題点を解決し必須事項を満足する、早稲田国際化多言語環境について概要を述べる。

2. 地域化の問題点と矛盾

Ｌｏｃａｌｅモデルによる地域化は、ソフトウエアに言語や文字に固有の情報を埋め込む、即ちハード・コーデッド (hard-coded) でなければ、ある言語に固有の処理が可能ではないと言う仮定によって作成されたモデルである。ＰＯＳＩＸの規格群は、実際にそれを想定しており、多数の言語もしくは文字コードを扱うと多数の問題点と矛盾を生じる [1]。

規格上、ソフトウエアの実行時にＬｏｃａｌｅを変更してはいけないという制限はない。しかし、Ｌｏｃａｌｅを変更すると、あるＬｏｃａｌｅが他のＬｏｃａｌｅに対し、排他的であることも可能であり、文字が別の文字に変わってしまったり、入出力が不可能になることもあり得る。即ち、内部コードであるＷＣが、異なるＬｏｃａｌｅであっても正しく文字を保存するとは限らない。即ち、Ｌｏｃａｌｅの実行時の変更の結果は、不定である [2]。また、ｍｂのデフォールトのデジグネーションとインボケーションが、Ｌｏｃａｌｅによってシステム依存となるため、同じＬｏｃａｌｅであっても動作の保証はない。即ち、異なるシステム間での同一動作の保証もない。さらに、フランス、インド、タイの国家規格文字コードは、

不定長文字コードであり、ｍｂｔｏｗｃ関数では、正しく変換される保証はない。また、複数文字コードを混在させた場合のコレーションも一意に定めることはできない [3]。

従って、Ｌｏｃａｌｅモデルでは、多数のＬｏｃａｌｅを集めても、同時に多数の文字コードを使用できる保証はなく、国際化とはならない。

これは、Ｌｏｃａｌｅが、真に言語に依存する処理、例えばソフトウエアからの利用者へのメッセージのように、使用者の言語に合わせるべき処理と、言語に依存せずに処理すべき、文字コードの混在などの処理を分離していないからである。即ち、１言語に真に依存する (必要のある) 排他的処理は、上述のメッセージや、ハイフネーション、スペル・チェッキングといった処理だけであり、他のほとんどの一般的処理は言語に依存しない、文字コードの混在処理として実現されなければならない [4]。

3. 国際化

Ｌｏｃａｌｅモデルは、言語への依存処理と、非依存処理の分離がなされていなかったため、国際化は不可能である。即ち、メッセージはアラビア語で、しかし、テキストの入出力は、アラビア文字と日本語の文字という分離がなされていないのである。しかし、一般に国際化の前提として、文字の入力・出力・基本テキスト処理・通信は、文字コードに対し、「他の言語に排他的な言語情報」が必要であると誤解されてきた。

文字コードも文字も、言語の情報を含んではいない。読み手がその情報を付加して読んでいるのである。文字の表示も、単に、文字が持っている表示時の約束によって表記されているのである。また、単独の文字コードでも複数の異なる表示上の規約を持つ集合は存在する。従って、文字コードに対し、「利用者」が選択・設定した表記方法の規約を文字そのものに与えれば、正しく表記することは可能である。この表は、同一の文字コードを複数の言語で使用し、その場合の表記方法が異なれば、複数必要であることは自明である。しかし、互いに排他的言語情報が文字の表示に必要なのではなく、文字に表示時の規約を与えれば良いのである。従って、文字コードに対し、たった一つの言語のためにハード・コーデッドな情報を付加する必要は全くない [4]。

このように、複数の文字コードを混在させても、その文字コードで示される文字を文字の集合に変換し、その文字の集合の要素である文字に対し、表示時の規約を付加すれば、全く言語に依存せずにあらゆる文字を混在表記することが可能となる [5]。ここに言語への依存性をハード・コーデッドする必要がないことを再度明記しておく。

同様に、基本テキスト処理である、検索、置換、挿入、削除といった処理に関しても、言語依存性は全く不必要である [6]。ここでの条件は、各文字コードが指定する文字の持つ番号が、他の文字コードの番号と重ならなければ良い、ということだけである。この条件は、正しい逆変換が可能でなければならないことを示している。

入力に関しては、入力機構のルールは言語に依存するが、プログラマブルな入力機構によって、入力機構そのものを言語依存にする必要はない。また、それによって、同一の入力機構で、多数の異なる文字コードを異なる言語との組合せで使用し、入力することが可能であり、Ｌｏｃａｌｅによって入力言語を制限される必要はない。

このように、国際化とは、言語に非依存な処理を示す。即ち、テキスト処理は、国際化可能な処理と、互いに排他的な言語依存の処理に分れる。しかし、互いに排他的な処理であっても、異なる複数の言語のテキストを同時処理できないということではない。ほとんどの基本的処理は、国際化処理で対応可能である。そして、利用者へのメッセージのように、排他的且つ１種の言語であるべき時、特定の言語を指定することこそが地域化であり、機能的には、地域化は国際化の真部分集合であると定義できる [7]。

注意: 文字コードは、文字を直接指定しているわけではない。文字の部品によって構成された文字コードも多数存在する。従って、国際化を満足するためには、文字コードから、「文字」に変換し、文字の集合として扱わなければならない [8]。

4. 多言語処理

言語情報が決定できなければ処理内容を決定可能でない処理を、言語依存処理と呼ぶ。この時、文字コードが言語固有の情報を持たなければ、処理環境全体を特定の言語に合致させなければならない。これがＬｏｃａｌｅの基礎概念でもある。即ち、文字コードが言語を示す情報を持たないので、特定の識別子に言語名を与え、特定の文字コードでのテキストを全て、識別子によって与えた言語として処理を決定する。この言語情報を分離し、処理機構の動作を決定する方法を、古典的言語依存処理と呼び、その概念から、１言語の処理のみ可能である。

古典的言語依存処理では、単一の文字コードが複数の言語を表記しうる場合、矛盾が生じる。特に、ＩＳＯ８８５９シリーズを処理する場合、複数の言語の処理は不可能である。従って、基本的にＩＳＯ８８５９シリーズをモデルに作成されたＩＳＯ１０６４６には、複数言語の同時処理に対する工夫はない。従って、ＩＳＯ１０６４６を用いた場合、一見複数言語の同時処理が可能であるように見えるが、実際には不可能である。

一方、内部コードの１文字を示す単位に言語を示すタグを付加し、各々に言語情報を与えれば、単一の内部コードであっても、なんら矛盾なく複数言語の同時処理が可能となる。本システムでは、この言語情報を持つ内部コードをＴＭＣ (Text Manipulation Code) と命名し、ＷＣから変換して作成される [6, 8, 9]。

極めて類似した言語及び単語が存在するため、自動的にＷＣのシークエンスに言語情報を付加することは可能ではない。

注意: 特定の言語情報を持つ文字コード列であっても、コーリング・コンベンションを揃え、統一された手法で言語固有処理を行なうことが可能である [6, 9, 10]。

5. 世界の文字

5.1 文字の構造

一般に文字を分析し、その情報を抽出する場合、表音文字、表意文字という分類が多い。しかし、音と意味は、文字の構造を示してはいない。また、全ての文字には名称を与えることが可能であり、さらに、発音時の音価を与えることができるので、その意味ではみな表音文字である。

文字を音価で分類することは、実際は極めて危険である。単独文字での音価と文字列中での音価は必ずしも一致しない。また、基本的には音素文字であっても、ギリシャ文字のように有気音化する記号を付加し、音節記号化する場合がある。勿論、ペルソ・アラビア文字やヘブライ文字の用に、一般に音素文字と思われているが、ロゴグラムである場合もある。さらに、本来音節文字であるデーバナーガリ文字やタイ文字を音素文字と強引にみなすと、文字としての処理単位が不明となる。即ち、文字に関しては、音価ではなく、文字の持つ構造と処理単位で分析する必要がある [3, 6, 9, 10]。

世界の文字をその構造で分類すると、構造を持つ文字群と持たない文字群に分れる。構造を持つ文字群は、結合音節文字が主である [6, 9]。これらの文字群は、一般に結合前の基本音節文字に音節のモディファイアを結合し、音節を形成する [3, 6, 9]。従って、基本的に、音節が文字の単位となる。この音節の数は、無制限にあるわけではなく、さらに、無限集合は、計算可能な集合でもない。ある種のソフトウエアのよ

うに、無制限に組み合わせることは、明らかな間違いである。組合せが間違っていると、読者が音節の区切りを発見できない。結合の組合せは、正しく文字コードがデザインされていれば、ＢＮＦで明確に記述できる [6, 9]。

しかしながら、結合音節文字であっても音節が文字の処理単位とは限らない。処理の目的によって、複数の文字の集合が存在する [9]。そして、特定の文字の集合の要素である文字に対して図形を割り当てることが出力である。

5.2 文字と表記

文字に対する図形の決定は、文字自身の表記上の規約で決定される。文字は、表記方向、語中の位置によって、その図形が定まる。日本語の句読点である「。」は、横書きと縦書きで異なる。また、ペルソ・アラビア文字は、語中の位置で図形が定まる。また、蒙古文字のリガチャーも文字の前後関係で明確に定まる。

従って、文字は図形ではなく、図形の集合に対する名称である [3]。そして、その集合から適切な図形を選択することが出力である。ある種の文字は、偶然に図形が同じであるに過ぎないので、全て同一の、次の手順で図形を決定することができる。1) 表記方向による図形の集合の選択。2) 語中の位置による最終決定。また、文字自身が表記方向を持つことによって、右書き文字と左書き文字の混在が可能となる。

以上のことから、文字の図形は、文字自身の表記方向、表記方向の原点、文字列中の位置で決定可能である。ここで、文字自身が表記方向を持たない場合があり、これは、前後の文字の表記方向及び、単語の表記される方向で定めることができる。以上の情報から、文字コードに文字の表記方向などの情報を含めることなく、無矛盾に文字を表記できることがわかる。同様に、文字を表記する場合、単独の１文字だけでは、正しい図形が定まらず、正しい図形を定めるためには、最小の長さの文字列が必要であることが明らかである。

また、文字によっては、物理的な横書きもしくは縦書きの文字がある。これらの文字を混在させる際には、必然的にどちらかの文字を 90 度回転させなければならない。また、当然、物理的水平と垂直の混在表記が可能でなければ、利用者の要求を満たさない。このことは、１行のみの表示だけでなく、文字を２次元に配置させる機能が必須であることを示している [9, 11]。

入力機構は、縦書きでの上下双方の入力と、横書きでの左右両方向の入力が可能でなければならない [12]。従って、現在の左横書きにのみ対応するＸＩＭプロトコルでは、全く対応できない。

6. 文字コード

文字コードは、次の複数のタイプに分類される。1-1) 文字の名称で定義され、１文字が決定可能で且つその図形が決定可能。1-2) 文字の名称で定義されるが、１文字が決定不可能。2) 図形で定義された文字コード。この時、常に正しく１文字とそれに対応する図形が決定可能な文字コードのみが、文字コードとして適切である。しかし、ＩＳＯ１０６４６のように、１文字の決定が不可能で、また、図形で定義された部分を含む文字コードが存在する。このような文字コードでは、正しく処理可能ではない。即ち、処理系によって、文字も図形も異なってしまう。また、不定長コードである時、開集合となり、集合の要素数が不定となる文字コードも存在する。例えば、ＩＳＯ１０６４６では、不定長文字コードとして使用した場合、開集合となる。従って、結合音節文字群は、ハングルのように始めから結合形が定義されていない場合、正しく処理可能な保証は、全くない。従って、図書館の文献データベースのように、遠隔地の異なる処理系でのアクセスが要求される場合、１文字もそれに対応する図形も同じである保証がないため、全く使用に耐えない。これは、ＵＮＩＣＯＤＥでも同様である。特に図書館では、デーバナーリ文字やタイ文字を完全に無視することは不可能である。

一方、インドとタイの国家規格は、それぞれ、正しく１文字の決定が可能であり、ＩＳＯ１０６４６で生じる問題は発生しない。

ＩＳＯでは、ＩＳＯ２０２２によって、文字コードの拡張を許している。一般にＩＳＯ２０２２の完全なサポートは困難であると言われているが、困難であるのではなく、単に理解が困難なだけである。ＩＳＯ２０２２は、単純な関数を組み合わせるだけで実現可能である [3, 5, 7, 8]。しかし、インドの国家規格であるＩＳ１３１９４は、ＩＳＯ２０２２による拡張を越えており、これらの規格も満足する必要がある。これらの拡張には、文字のバリエーションが一定の範囲内であるため、可能な種類は少ない。従って、これらを満足することは不可能ではない。

これらの文字コードの拡張を完全に満足すれば、通信上、問題の発生は極めて少ない。しかし、上述のように、文字の物理表記方向等の情報はＩＳＯ２０２２には含まれない。ＩＳＯ６４２９がそれらの情報の送受を規定しているが、プリンタなどの表示機器を対象としており、双方向通信には不適切である [12]。

7. テキスト処理

以上から複数言語の混在ではなく、複数文字の混在として国際化処理を実現するべきであることがわかる。しかし、テキスト処理は、基本テキスト処理であっても問題は単純ではない。デーバナーガリ文字では、その規格に要求されている１文字の概念とカノニカル・オーダリングでの文字の概念が異なっている。ＫＳＣ５６０１におけるハングル文字でもカノニカル・オーダリングは文字コード上の順番とは異なる。また、平仮名と片仮名における小さい文字「ゃ、ゅ、ょ」は、本来音節の構成要素であり、改行時には音節文字として扱わなければならないが、カノニカル・オーダリングでは、独立した１文字として扱う場合がある。

このように、実際のテキスト処理の単位 (対象) は、必ずしも内部コード上の文字とは一致しない。即ち処理によって異なる文字集合が存在する。これらの集合の変換情報は、文字コードの表現形式の中に保持されなければならない。従って、内部文字コードは、単純に文字のインデックス番号だけの情報では不足する [3]。

特に、全ての文字列において、自動的に正しく改行位置を決定可能なわけではないことに注意が必要である。タイ文字やラオ文字では、入力者が明示的に決定しなければ、正しい改行位置を示すことはできない。従って、WWWやＳＧＭＬにおいて、ブラウザが自動的に改行位置を決定可能であるという仮定は成り立たない。従って、何らかの構造によって、改行の抑制を示す必要がある [9, 10]。

また、文字は、二次元座標に配置され、単語の表記方向と文字の単語内の表記方向は必ずしも一致しない。むしろ、左右両方向へ表示する文字を混在した場合、もはや、メモリー上での文字の順番と表示平面上での文字の順番は一致しない。従って、メモリー上の位置から表示平面上での位置を計算し且つ、その逆の計算を行なう処理が必須となる [9, 11]。そのため、表示機能は、テキストの編集機能の一部として実現されなければならない。

この基本的な機能として、ある文字列がどの特定の文字コードに由来するかを抽出する機能も必須である。これは、特定のスペル・チェッキング・ソフトウエアが、特定の文字コードのみを受理するからである。従って、多数の文字コードからなる文字列中から、特定の文字コード由来の文字列を抽出し、特定のエンコーディングに変換する必要がある [11, 12]。これをフィルタリングと呼ぶ。このエンコーディングは、受理側によって定まるため、送信側のデフォールトのエンコーディングは必ずしも使用できない。このことから、常に自由なエンコーディングの変換が可能な機能を用意し、実行中のソフトウエアのエンコーディングから独立した通信機能として実現しなければならないことが自明である [12]。

8. 早稲田国際化多言語処理環境

早稲田国際化多言語処理環境は、前述の問題を全て解決したマルチ・ウインドウ環境である。本システムの中核 [7] は、文字コード変換機構、入力機構、出力機構、通信機構、及び、それらを利用して作成された、マルチ・ウインドウ機構、ウイジェット、プログラミング言語処理系、ＯＳの基本文字 (列) 処理を置換するライブラリ [13] から構成される [14]。さらに、基本的エディタ、印刷、文書整形などのアプリケーション・ソフトウエアを含んだ、環境全体を国際化した多言語処理環境とする、極めて広範なシステムである。新たな文字コード及び言語の追加は、単にデータ表とフォントを加え、データ表をコンパイルするだけで良い。

本システムは、メタコンバータ・システムと命名された文字コード変換機構が、外部に定義された変換データベースをコンパイルして得られた最適化関数を仮想記憶内にロードすることによって、極めて高速に動作する [2, 4, 5, 8, 14]。この中心部は、全てのプロセスによって共有されるため、既存のソフトウエアのように大量の資源を消費しない。同様の手法によって、内部文字集合に対し、最適化された関数によって図形を与え、この部分も全プロセスが共有する [2, 4, 5, 8,14]。従って、各プロセスは、変換時の情報を保持する極めて小さな資源のみプロセス固有となる。この変換時の情報は、任意の個数を保持可能なため、いかなる初期値を持つ文字コードへの対応も可能である。

本システムのモデルは、システムに何も指示をしない時には、全ての文字コードの混在処理のためのデフォールト条件が与えられ、ＩＳＯ２０２２を完全に満足し同時に各国家規格のための拡張を満足する。この状態でのモデルをＧｌｏｂａｌモデルと呼ぶ。これに対し、特定のデフォールトを選択し、特定の複数のエンコーディングの設定を可能とするモデルをＭｕｌｔｉ－Ｌｏｃａｌｅモデルと呼ぶ。このモデルでは、設定されていない初期値に対しては、Ｇｌｏｂａｌモデルでの初期値が使用されるため、使用不可能な文字コードは生じない。さらに、特定の１エンコーディングのデフォールトのみを許可するモデルが、Ｌｏｃａｌｅモデルとなる。Ｌｏｃａｌｅモデルであっても、初期値にない文字コードは、Ｇｌｏｂａｌモデルでの初期値を使用することにより、正しく動作する。当然ながら、アプリケーションの実行時にＬｏｃａｌｅを変更してもエンコーディングを変更しても、なんら問題は起きない。また、単一のアプリケーションに対し、複数個のＬｏｃａｌｅを同時に使用し正しく動作することを保証している。これらのモデルに関しては、参考文献を参照されたい。

これらのモデルは、Ｍｕｌｔｉ－Ｌｏｃａｌｅモデルがグローバルモデルの、Ｌｏｃａｌｅモデルが Ｍｕｌｔｉ－Ｌｏｃａｌｅモデルの、それぞれがサブセットとなっており、ＰＯＳＩＸに規定された関数であっても、互いに矛盾しないようにデザインされている。従って前述の問題は発生しない。また、これらのモデルは、混在実行が可能である。

メタコンバータ・システムを用いて、テキスト処理、入出力、通信機構が実現され、全体の統一性と無矛盾性を保っている。さらに、これらの機構を使用して、Ｘ１１ライブラリとＯＳライブラリが置換され、これらを先にリンクすることによって、Ｘウインドウ・システムとＯＳのＬｏｃａｌｅ関連関数が全て国際化される。また、ウイジェットは、二次元配置処理機構を実現し、ほぼ完全に新たなウイジェットとなった。しかし、コーリング・コンベンションとリソースは過去のウイジェットに対し上位互換である。

また、ＴＭＣ関数を効率良く利用する言語処理系として、ＦＯＲＴＨを作成した [10]。本ＦＯＲＴＨは、国際化多言語処理用に拡張したＣｏｍｍｏｎ ＬＩＳＰを作成するに十分な機能を持ち、極めて省資源で高効率である。本処理系は、全システム・コールをリンク可能としており、新たなエディタ等の作成に用いられている。

入力機構は、あらゆるタイプの入力手法を満足し、同時に非母国語入力者への便宜のため、極めて高い自由度を持つ構造をしており、各機能を任意にプログラマブルなオートマトンに設定可能である。これによって、容易に目的に即した入力手法を実現することができる [12]。

9. 国際化多言語環境を用いた自然言語処理

早稲田国際化多言語環境を設計し製作する過程で、極めて大量の書き言葉と話し言葉の差の情報を得た。また、今日まで注目されることが少なかった、言語分析における複数言語の重層性解析の手がかりの多くが発見され、孤立語・屈折語・膠着語も境界領域の言語に関しても分析する手法を得た。特に文字の構造により、祖語の分析が可能となり、音訳を元にする既存の分析とは異なる手法により、新たなパージングの可能性が得られている。

10. おわりに

早稲田国際化多言語環境は、極めて省資源・高効率に国際化処理と多言語処理を実現する。しかしながら、既存のXサーバーや規格の欠点を如実に示す。そのため、Xサーバーの縦書きへのプロトコルの拡張を行なっている。また、本システムは、フォントと文字コードを分離したため、文字コードが必要とする文字図形をネットワーク・ワイドに配布する、全く新しい概念のフォント・サーバーの作成を可能とした。これによって、異なるサイトで同一の文字図形を表示可能となり、外字の問題が解決される。また、前述のように、ＳＧＭＬやＩＳＯ２０２２、ＩＳＯ１０６４６の欠点を、具体的な文字とその表記情報を示して明らかにすることが可能となった。これによって、安易な判断による文字コードの選択により将来発生する問題と障害を回避することが可能となった。

また、教育・データベースにおいては、言語の依存情報と非依存情報を分離したため、極めて高度な利用が可能となった。また、分散環境における文字と図形の無矛盾性を保証できるため、計算機を使用した語学教育で、電子メイルを使用して、より頻繁な作文添削が可能となった。さらに、複数言語処理が可能となったため、ＩＣＡＩといったシステムで、文字コードや言語の制限に縛られることなく、より多数の関連言語を利用した教育が可能となった。特にデータベースにおいては、既存のデータを変更することなく、本システムによってアクセスすることが可能であり、本システムの内部コードを使用することによって、現在使用されている出力装置を活用しながら、国際化することが可能である。

これらの知見と国際化多言語環境によって、いまだ未解読の文字もしくは、歴史的な資料へのアプローチが極めて容易となった。現在、コード化されていない文字群が多数存在するが、可能な文字コードの構造を予め予測して最適化した変換関数を用意してあるため、極めて容易に使用文字群を増加することができる。従って、完全に文字コードを決定する前に、データ表を変更して、最適な文字コードをデザインすることが可能である。実際に蒙古語関連文字群のコード化に使用し、極めて短時間に文字コード群の提案が可能であった[15]。これは、極めて多数存在する異なる文字群で記述された、仏経典の処理に有効であることも示している。

本システムは、より多くの自然言語処理を可能とするため、入出力機構として、画像と音声の処理機能を付加する予定である。9章で述べた書き言葉と話し言葉の差の情報をもとに、新たな自然言語処理、特に歴史的表記法を採っているために現在の音声表現との差が大きい言語の処理に効果が得られると考えられる。

本システムは、フォントの調整が予定どおり進めば、1996年度中に第1版が配布される予定である。

11. 謝辞

本システムの開発にあたって、文字、言語、正書法の研究のため、極めて多くの母国語話者と研究者の協力を得た。ここに深く感謝の意を表する。多数の協力者なくしてこの研究が進むことはあり得なかった。そして、世界中の文字と言語が平等になんの不都合もなく処理可能であることを望む人々の声が、本研究

の原動力であることを明記しておく。不完全な国際化、多言語化は、新たな障害を生じるに過ぎないのである。

本研究の一部は、文部省及び通産省の助成を受けており、関係各位に深く感謝の意を表する。

12. お願い

最後に、本システムのフォントはＩＳＯ１０６４６にない文字も含め、極めて多数作成されているが、未だに十分とは言えない。単に本システムの利用を求めるだけでなく、フォント及び入力、表示、変換データの作成に多くの協力が必要である。本システムの一日も速い公開のため、お手伝い頂ける方々を募集致しております。

参考文献

[1] Kataoka, Y., et al. A Model for Input and Output of Multilingual Text in a Windowing Environment. ACM Transactions on Information Systems, 10(4):438–451, October 1992.

[2] Kataoka, Y., et al. Multilingual I/O and Text Manipulation System (1): The total design of the generalized system based on the world's writing scripts and code sets. Proc. 49th Annual Convention IPS Japan, 3:299–300, September 1994.

[3] Kataoka, Y., et al. Codeset Independent Full Multilingual Operating System: Principles, Model and Optimal Architecture. IPSJ SIG Notes, System Software & Operating System, 95(36):25–32, March 1995.

[4] Kataoka, Y., et al. The worldwide multilingual computing (1): Essentials, principles and scope covering all characters in the world. Proc. 51th Annual Convention IPS Japan, 3:245–246, September 1995.

[5] Uezono, K., et al. Multilingual I/O and Text Manipulation System (2): The Structure of the Output Method rawing the World's Writing Scripts beyond ISO 2022, Proc. 49th Annual Convention IPS Japan, 3:301–302, September 1994.

[6] Kataoka, T., et al. Multilingual I/O and Text Manipulation System (3): Extracting the Essential Informations from World's Writing Scripts for Designing TMC and for the Generalizing Text Manipulation, Proc. 49th Annual Convention IPS Japan, 3:303–304, September 1994.

[7] Uezono, K., et al. The Worldwide Multilingual Computing (2): Functions, Model, Design and Architecture of Multilingual I/O TM/C System, Proc. 51th Annual Convention IPS Japan, 3:247–248, September 1995.

[8] Tanaka, T., et al. Multilingual I/O and Text Manipulation System (4): The Optimal Data Format Converter to/from MB/WC/TMC, Proc. 49th Annual Convention IPS Japan, 3:305–306, September 1994.

[9] Kataoka, T., et al. The Worldwide Multilingual Computing (4): Essentials for the Multilingual Text Manipulation, Proc. 51th Annual Convention IPS Japan, 3:251–252, September 1995.

[10] Maruyama, K., et al. The Worldwide Multilingual Computing (7): Multilingual Programming Language for Advanced Researches, Proc. 51th Annual Convention IPS Japan, 3:257–258, September 1995.

[11] Oya, T., et al. The Worldwide Multilingual Computing (5): Multilingual Text Manipulation and Text Widget, Proc. 51th Annual Convention IPS Japan, 3:253–254, September 1995.

[12] Daikokuya, H., et al. The Worldwide Multilingual Computing (6): Multilingual Text Interprocess Communication and Input Mechanism, Proc. 51th Annual Convention IPS Japan, 3:255–256, September 1995.

[13] Yamanishi, S., et al. The Worldwide Multilingual Computing (8): Multilingual Basic Environment - C Language and OS, Proc. 51th Annual Convention IPS Japan, 3:257–258, September 1995.

[14] Tanaka, T., et al. The Worldwide Multilingual Computing (3): An Implementation of the Multilingual I/O TM/C System and Waseda X11, Proc. 51th Annual Convention IPS Japan, 3:249–250, September 1995.

[15] Kataoka, T., et al. Definition of the Mongolian Character Codesets Enabling Multilingual Text Manipulation, IPSJ SIG Notes, Computer and Humanities, 96(15):61–66, January 1996.