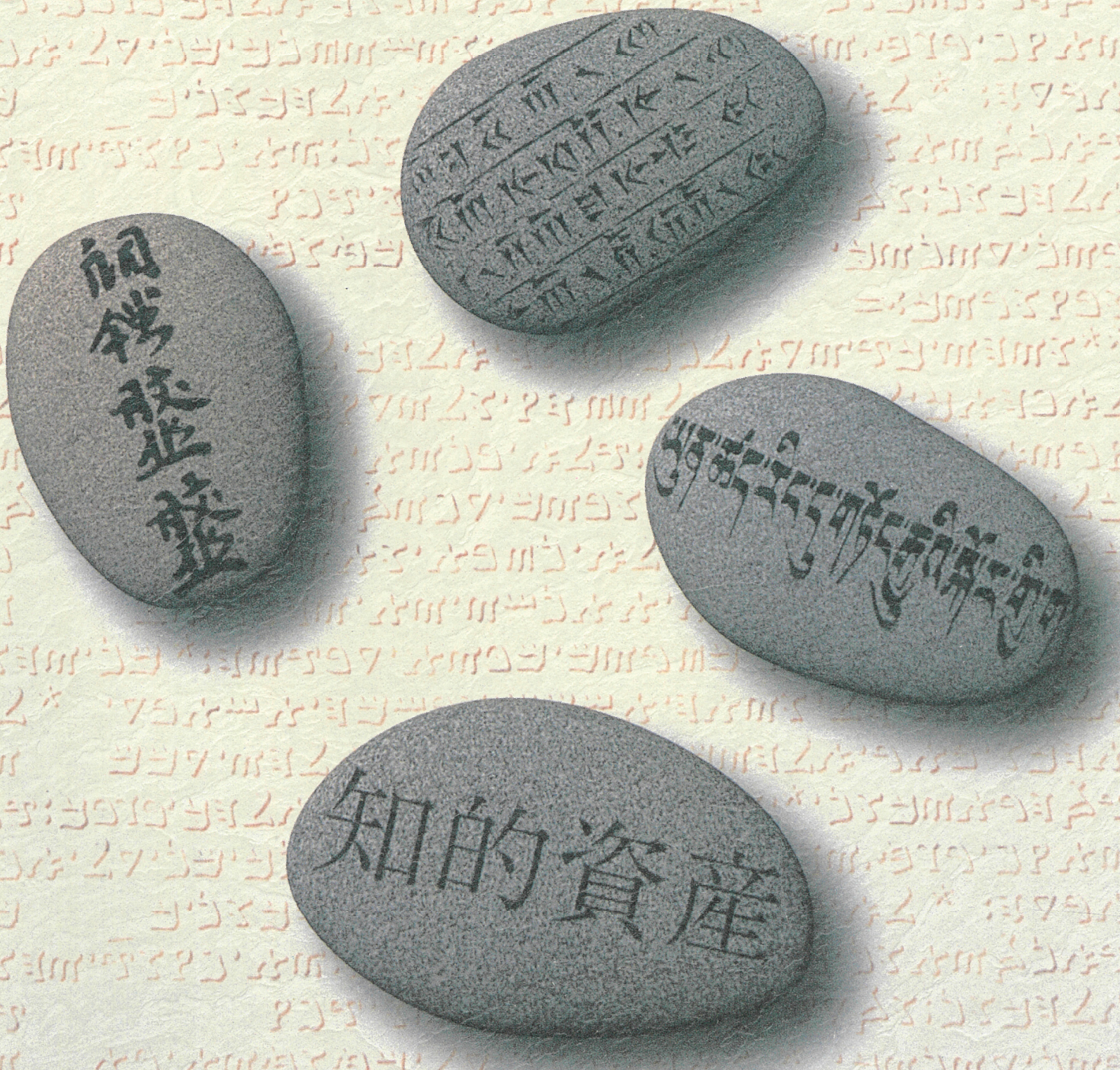
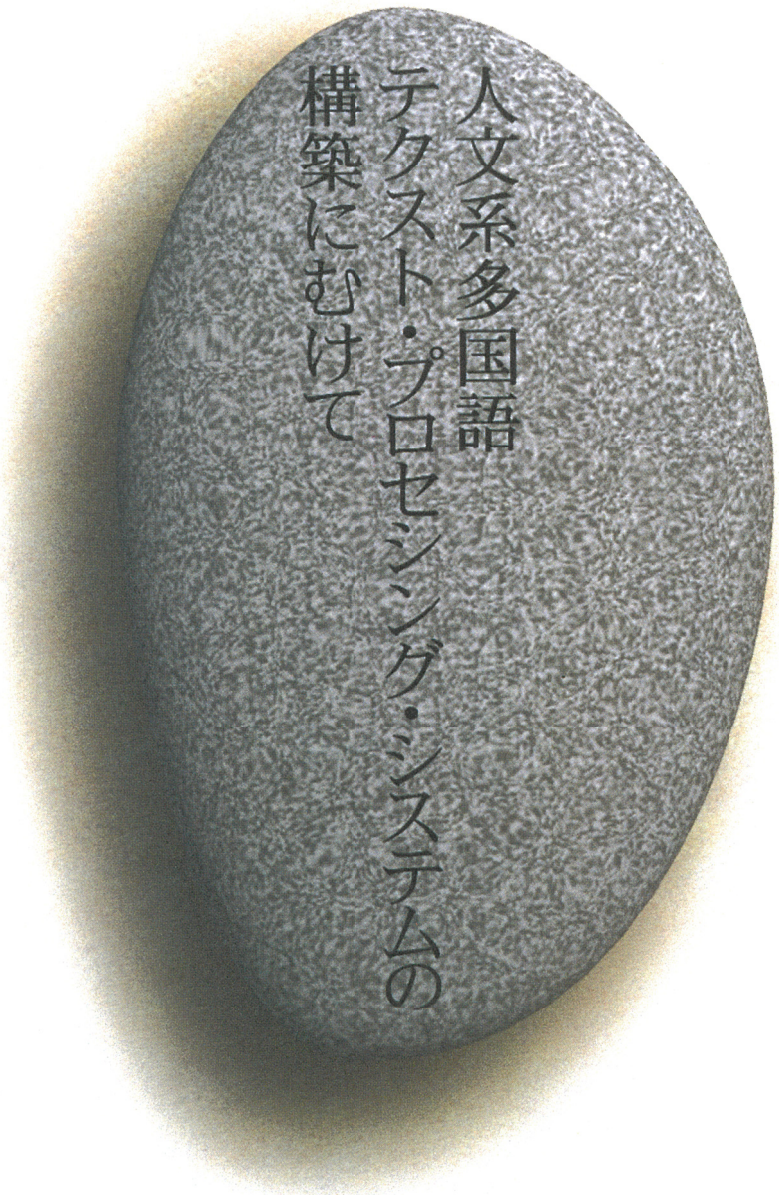
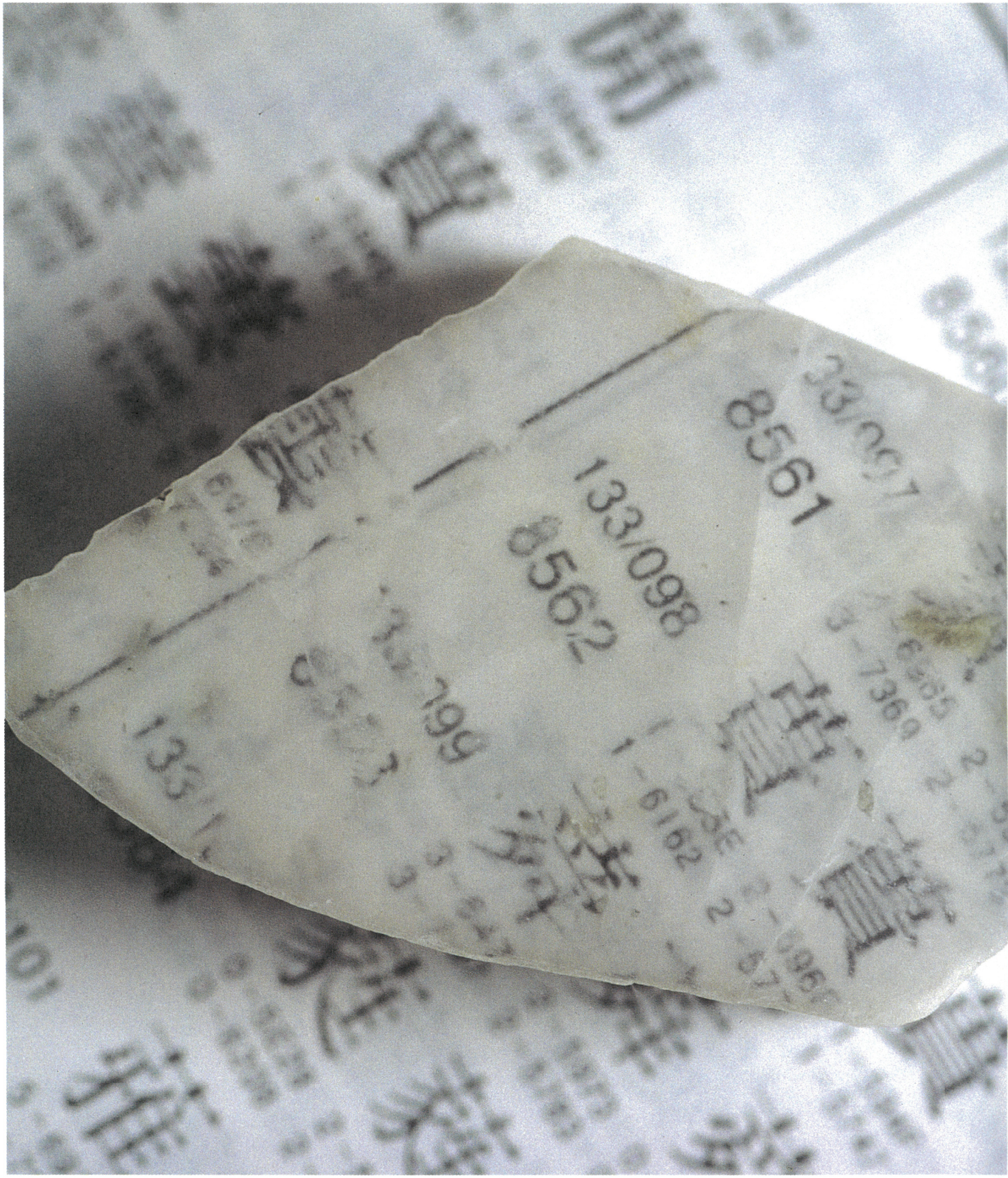
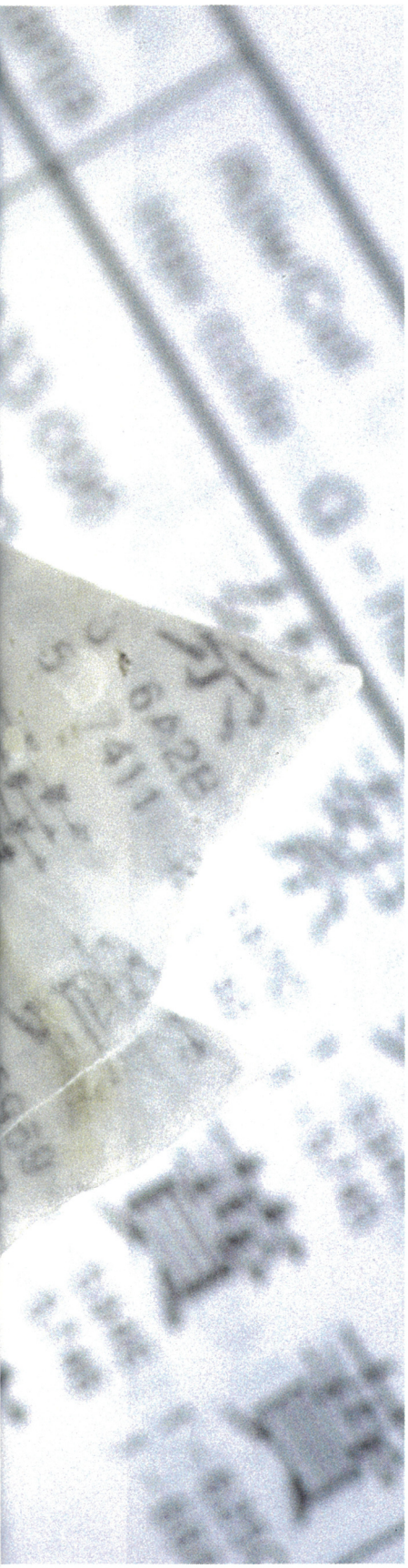
人文系多国語
テクスト・プロセシング・システムの
構築にむけて




人文系多国語
テクスト・プロセシング・システムの
構築にむけて


133/007
8561
133/098
8562

知的資産の継承と創造



まずはじめに、人文学の研究対象となっている言語はいくつあるのでしょう。身近にいる研究者の専門領域を思い浮かべてみると、日本語を基盤にして、日本語とも密接な関係をもつ中国語、韓国語、台湾・高砂族などの言語、ヴェトナム語、シャム語、タガログ語など東南アジアと環太平洋圏の諸言語、スワヒリ語などのアフリカの諸言語、サンスクリットとドラヴィダ語などのインド諸語、チベット語、西夏語、ペルシャ語、アラビア語などがあります。そして欧米系では、古くはギリシャ語、ラテン語、ヘブライ語、コプト語から、英独仏露伊西などの西洋近代諸語があります。

これは主に国や地域別の地理的分布の視点からの分類ですが、各言語にはそれぞれの時代に固有の言語体系と文字とがあり、時系列でも研究領域はさらに分かれますし、また、近代国家形成にいたる中央集権化の過程で排斥されてきた地方語や少数者言語も、近年ますます活発に研究されていることを考え合わせると、地理的系列もさらに詳細な地図を作成する必要がありましょう。固有の文字をもたない言語、あるいは旧植民地のクレオール語もあります。

このような言語の多様性こそが人類の知的資産の豊かさを象徴するのであり、その多様性を許容し、再認識する視点から、次代の新たな知的創造活動が生まれてくるのでしょう。人間はいつの時代でもことばで考え、ことばで意思の疎通をはかってきたのです。文字はその知的活動の記録係です。

ロゼッタ・ストーンの碑文に刻まれた古代エジプトの象形文字、楔形文字、西夏文字などの解読は、永年の文字学者の夢の実現でした。死滅した言語の復元、あるいは文字をもたない少数者言語の記録と体系的研究は、まさに現代的な課題でもあります。このような多様な言語によって創成されてきた知的資産の継承に、コンピューターはどのような役割をはたすのでしょう。世界語としてのコンピューター用英語の便利さの陰に、多様性のもつ創造力が損なわれることはないのでしょうか。

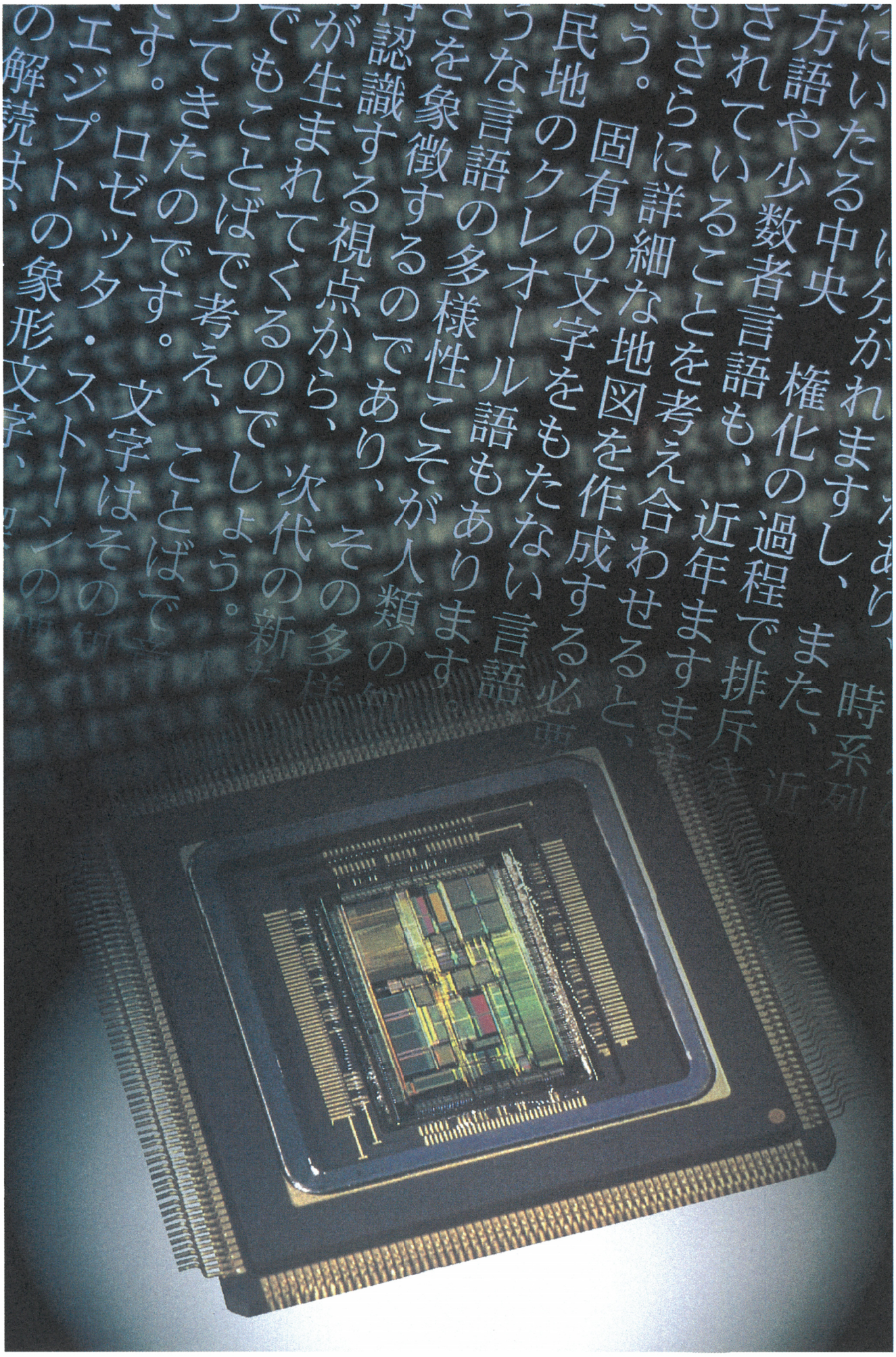
コンピューターは豊かな言語表現を保証しうるか

さて、同じ言語でも、人文学の研究者は「コンピューター言語」に弱いというのが、皮肉めいた定説ですが（もちろん例外は多くあるでしょう）、もともと「計算機」にすぎなかったものに、さしたる言語表現能力を期待してこなかったというのが、偽らざる気持ちでもあり、貧弱だったのはむしろ機械の能力のほうだったことは確かです。限られた文字数しかないアルファベットのタイプ・ライターをもとにしたキーボードに、ひらがな・漢字を打たせるほうが間違いだ、などと今さら言っても始まらないかもしれませんが、ワープロで使用しうる字種の問題一つとっても、かなり限定されたものでしかないことは明白です。

ところが、一方では、活字による印刷はみるみるうちに姿を消し、電算写植・組版へと移り、近年ではパソコンでも、手軽に高速かつ良質のプリントができ、印刷所に依頼しなくてもある程度までは手元でできるようにまでなりました。グーテンベルク（あるいはコステル）以来の活字印刷の歴史を考えると、革命的な変化を遂げつつあるように思います。小さな活版印刷所は姿を変え、大きな印刷会社は大量の印刷物だけを扱うようになっています。かつて美しい活字と見事な組版の技術をもち、面倒な文字が多数混在するような、主に人文学系の印刷物を、出版社と協同して書物に仕立てていた中堅の印刷所は、これからはどのような形で存続するのでしょう。そしてもっと深刻なことに、コンピューターで使える漢字は、実際にはあいかわらず1万字程度のままであり、やがては2万字になるにしても、扱われる言語種類は限られています。

他方で、学術文献の情報化がもっとも遅れているのが人文学系であるとも言われています。情報化が「電子化」を指すのならば、これもまた計算機の多言語処理能力の貧弱さ、というより多言語はおろか、日本語の漢字処理能力でさえもおぼつかないからに、ほかなりません。もともと計算機には英語しか入らなかったからでしょう。

しかし、日本の人文学がこのような活字・文字文化の変革に積極的に対応してこなかったことに（むろん例外的な研究者もいるでしょう）、責任の一端はあるのかもしれません。それにしても、使用する漢字が、電気器具などと同じように、なぜ工業用JIS規格で制限されるのかという、素朴な疑問はなお残ります（さしずめフランス語ならアカデミー・フランセーズ、日本語なら学士院か国語審議会の所轄でしょうに）。


に分かれますし、時系列
方語や少数者言語も、近年ますま
されていることを考え合わせると、
もさらに詳細な地図を作成する必
う。固有の文字をもたない言語
民地のクレオール語もあります。
うな言語の多様性こそが人類の
さを象徴するのであり、その多
が生まれてくるのでしょう。
でもことばで考え、ことばで
てきたのです。
す。ロゼッタ・ストーン
エジプトの象形文字
の解読は、


①
弐
参
四

マルチ・メディアは マルチ・ランゲージになりうるか

本研究プロジェクトの起源はそもそも、1994年秋に、総長の諮問機関の一つとして全学的に組織された「東京大学マルチ・メディア研究会」（座長：藤本強 人文社会系研究科委員長・文学部長）にあります。これはマルチ・メディアと大学教育・研究のあり方を検討する、全学的な自由な情報交換の場です。その場で、いわゆるUNICODEで想定されている漢字コードが、中国語、韓国語、日本語を併せて2万字分しか用意されていない事態に、文系諸学問はどう対応するのか、という問題が提起されたことから始まりました。

他方では、図書情報の電子化を進める上での障害が、まさに多言語処理であり、とりわけ膨大な漢籍の書名をいかに正確に入力してゆくのかという問題もあります。学内に所蔵される膨大な資料（書物だけではなく、動植物・人体標本、考古学資料、実験器具、等々）を、マルチ・メディアを利用して、画像として公開する方向も検討されました。それに並行して、われわれは多言語の「文字」の検討を理学系研究科（理学部）の坂村健氏と始めました。

坂村氏は、「歴史を通じて、世界に存在し、存在した、ありとあらゆる文字の収集と電子化、そしてコード化」という大胆な提案をされ、この提案はある意味では、「文字とはなにか」、「なにを文字として認めるか」という言語学的・哲学的命題でもありますので、言語・文学を専門とするわれわれを大分悩ましてきました。とりあえずは、時系列での区分、例えばグーテンベルクの活字印刷以前か、以後か、あるいは社会史的に文字の形態が画期的に変わる節目（例えば国家・民族の存立、産業革命、明治維新、等）の設定を、コード化にも導入すべきであるという、逆提言をしました。

マルチ・メディアの流行とはいえ、いかに立派な電子辞書や電子百科事典を構想しても、その基盤である「文字」の制約を解決せずには、結局のところ、情報網上で交換・交信不可能な、中途半端なものに終わってしまうでしょう。そして、日本においては、多言語のなかでも、中心となるのは当然日本語であります。

日本語漢字の画定作業と統一コード化の緊急性、多言語処理可能なプロセシング・システム構築の必要性を、吉川弘之総長に申し上げたところ、総長のご推薦をいただき、日本学術振興会によって承認されたのが本研究プロジェクト「人文系多国語テクスト・プロセシング・システムの構築に関する研究」であります。これは平成7年度「日本学術振興会産学共同研究支援事業」の一貫であり、8月から正式に発足しました。

日本語漢字の画定作業

まずは、柔構造であるがゆえに全体像の把握が難しい日本語、とりわけ漢字の壁に突き当たったわけであります。漢字の文字種は幾つあるのか、既存の統一的なコードはないのか、中国語漢字のどこまでを日本語として認めるのか。中国語の漢字との差異、異体字の処理、国字（日本で作った漢字、人名、地名、等）、いずれもすでに国語学者、中国語学者、漢字研究者が研究されてきた問題だと思いますし、すでに漢字フォント電子化の作業を進めている研究グループ（例えば勝村哲也・丹羽正之『漢字典』プロジェクト）もあり、先行作業をよく参照する必要があります。

われわれは、日本語に関しては、人文社会系研究科（文学部）・国語学の山口明穂教授に監修をお願いしました。具体的には『（諸橋）大漢和辞典』（58,000字）、『漢語大字典』（約60,000字）を基準とし、国字（約3万字）を収集する計画を立てました。本年度はそのうち頻度の高い2万字、すなわちJIS第1・第2水準、補助漢字を合計した12,156字と、今日の国語辞典や百科事典などで「外字」として使われている比較的頻度の高い漢字、約7,000字種を収集、選定することを目指しています。JISの規格の検討から始めるのは、既存コードとの上位互換をはかるためであり、後述の漢字データ・ベースができれば、あるいはその構築過程で、コンピューター用とは別途に、純粋に国語学的見地からの統一コード化も可能になります。

日本語漢字の統一コードの割当

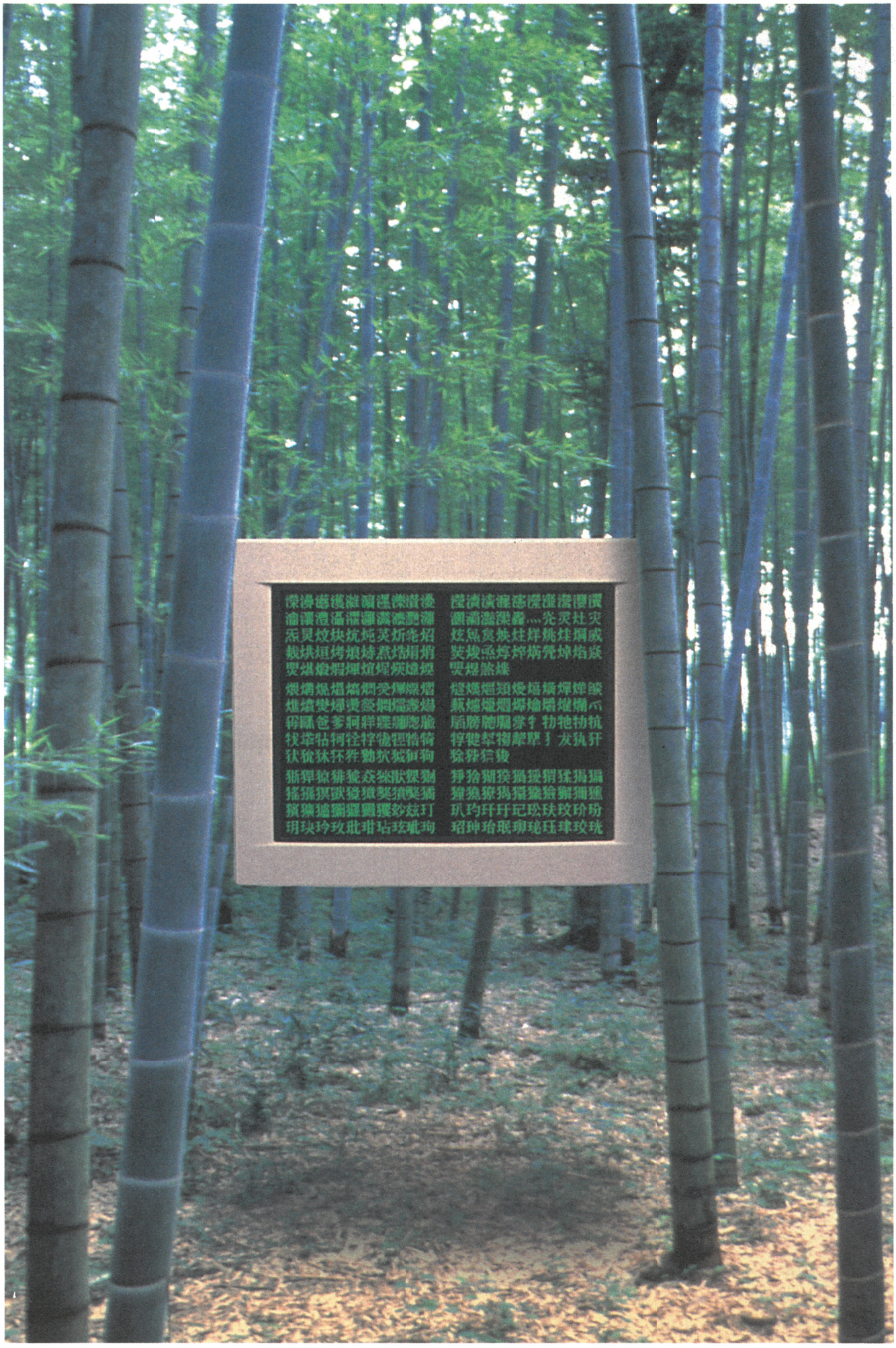
日本語漢字のコンピューター用統一コードの割当は、坂村氏が担当します。本来の漢字コードそのものは、「外字」の概念などとは無縁のものであり、特定の機種やOSからも独立した、統一的かつ開放的なコード系の構築であるべきです。文字数も予め限定できるものではありません。現在のところでは、UNICODEの2万字の無国籍的な漢字枠に、無理にあてはめることを前提として作業を進めることはできませんので、おそらく柔軟に拡張可能な、約10万字のコードを日本語漢字専用に用意し、中国漢字は別途に専用に約10万字かそれ以上のコードを用意すればよいと考えています。

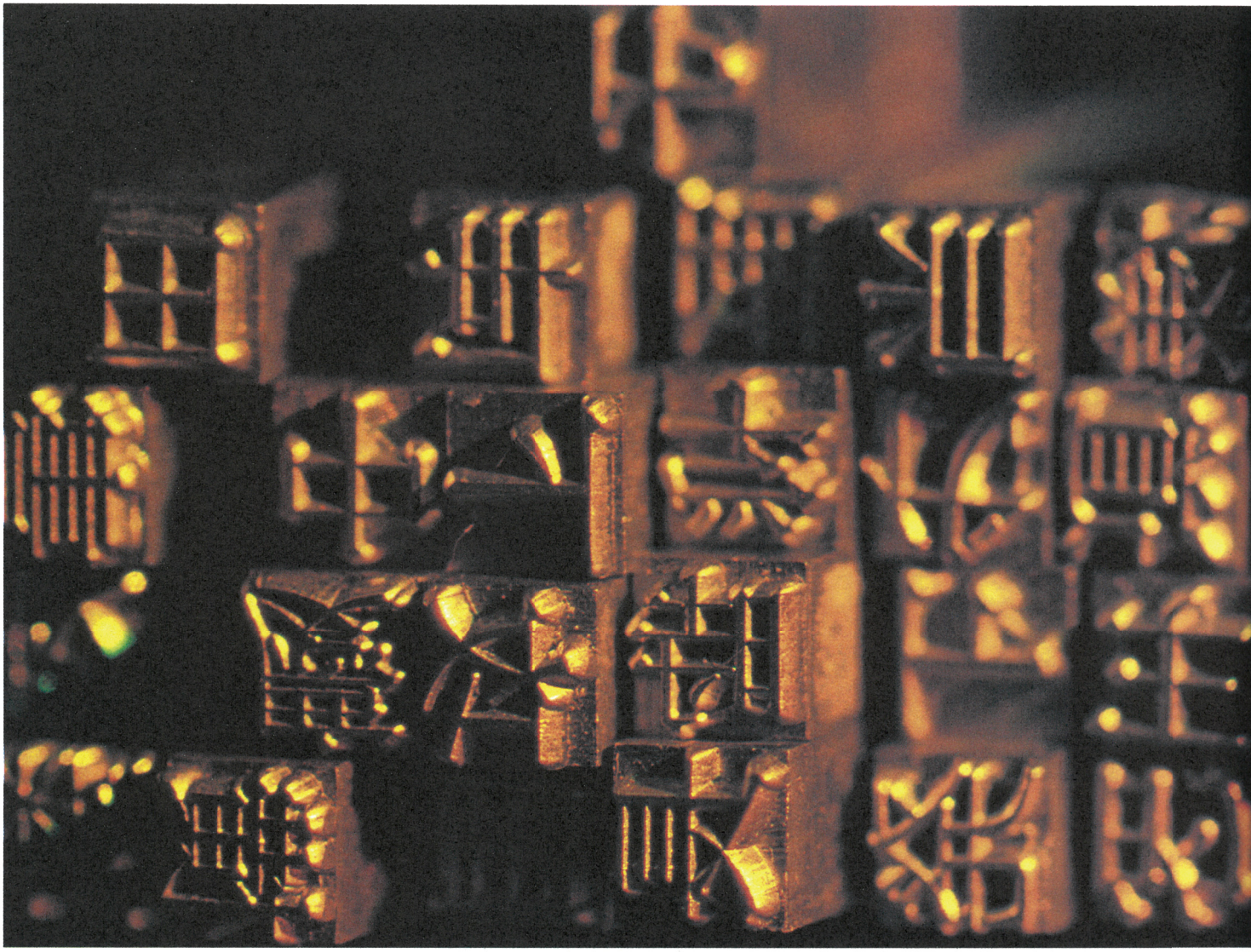
当然のことですが、日本語漢字と中国語漢字は、時代を遡れば遡るほど、同じ字形のものが重複するでしょうが、効率性の見地から、予断をもって無理に統合することの愚かさは避けたいと思います。すなわち、歴史的に中国の漢字を用い、漢籍を学んだ過程で、日本人が何を選び、何を選ばなかったのか、そして国字としてどのような漢字を作ってきたか、その文字の歴史と現状を明らかにするためにも、日本語と中国語とはたとえ重複するものがあるにせよ、一旦は区別して出発すべきでしょう。それがそれぞれの文字文化に対するわれわれの敬意です。

先に述べたように、時系列の観点から、現用（今日主に用いられている文字）を重視すれば、やはり教育漢字、常用漢字、あるいはJIS第一・第二水準、補助漢字、それ以外の頻度の高い7千字を加えた、約2万字が優先されるべきであろうと思います。

このような柔軟な枠組みで、現在すぐに文字種約8万字の漢字をコンピューター・システムに組み込めるのは、坂村氏のTRONであろうと思いますし、またそのようなご提案をうけました。詳細は坂村氏ご自身による紹介文をお読みください。何よりもわれわれの身近にいるOS創案者ですから、人文学系の研究に役立つ多言語処理用コンピューターを作っていただければ幸いです。

日本語漢字約8万字を、どのようにして機種に依存





せずに通信・交換可能にするか、いかに互換性をはかるかは、当初からの課題です。社会的な要請とコンピューター設計者をはじめとする関係者の対応を待つしかありません。

われわれが提案するコンピューター用の日本語漢字統一コード表、その後に作成が予定される国語学的見地からの漢字データ・ベースは、ご関心のある方々に印刷物として配布し、あるいはオンラインで公開する予定です。

漢字データ・ベースの構築と公開

収集・選定された漢字は、漢字データ・ベースとして、漢字固有の統一コードのほかに、先行する各種コード（JIS、『大漢和』の番号、等々）、画数、部首、読みなど、必要なデータを入れ、順次公開してゆきます。

この文字種としての漢字データ・ベースに、将来は教育的な配慮を加えて小学生や中学生に開放し、オンラインで読めない漢字を見つけたり、難解な漢字の意味や用例、異体字や中国語漢字との比較などのデータを加えてゆくことによって、漢字に対する興味を喚起できればという夢をみておりますが、それは将来の漢字専門家にお任せしたい仕事です。

日本語漢字フォントの作成

日本語漢字の統一コード化が本来の目的ですが、画面や紙面に出すためにも、標準的なフォントが必要です。

漢字フォントの使用にはさまざまな権利関係が生じますので、将来的には「東大・日本語漢字フォント・セット（約8万字）」を作成し、教育的・学術的

な使用に限り無料で公開できることが望ましいと思っております。例えば海外の図書館や研究機関から、国内の日本語や漢籍のデータ・ベースにアクセスすることが可能なように「フォント・セット」を予め配布する事態が予測されます。

台湾や中国ではすでにそのようなフォント・セットが準備されており、また日本国内においても『漢字典』（勝村哲也・丹羽正之）プロジェクトで57、415字のフォント・セットができています。また、規格協会で制作した平成明朝体も公開を検討中と聞き及びます。

当面は、本年度の2万字の選定と同時に、ご協力いただける方々が提供くださる既存のフォントを再検討し、日本語漢字として修正すべき点を再検討しつつ、利用させていただきたいと思っております。よろしくご協力をお願いいたします。そして次年度以降に向けて、フォント制作用の研究費の申請をしてゆく予定です。

「活字」保存のために

本研究プロジェクトとは直接関係しませんが、活字による活版印刷が電算写植へと移り変わる過程で、鉛活字そのものが廃棄され、姿を消しつつあります。できれば字母と鋳型とともに貴重な資料として保存したいと思います。本研究プロジェクトの一員でもある東京大学資料館館長　青柳正規氏に相談したところ、資料館で保存に努力していただけることになりそうです。本来は印刷所の命とも魂ともいうべきものでしょうが、技術的変革で不要となり廃棄処分にする場合には、お知らせくださるよう、あるいはそのような伝聞があればご紹介くださるよう、お願い申しあげます。

多言語テクスト・プロセシング・システム構築への展望

本年度は日本語漢字のコード化の作業だけでも大分難航しそうですが、幾つかの計画を萌芽的に始めておきたいと思っております。

多言語文字コードの設定によって実現可能となるものは、まず第一に「多言語電子辞書」があります。比較的に容易な西洋近代諸語と日本語との対応関係となる語彙集の収集・作成・電子化作業から始めますが、むしろ学術的により必要度が高いのは、既存のコンピューターには組み込まれない少数言語や古語の語彙集でしょう。字母が用意できれば、それをもとにフォントを作り、多言語システムに組み込むことも可能です。

第二には、大規模な語彙集作成のために不可欠なのは、精度の高い光学的読取装置OCR(Optical Character Recognition)の開発です。一定量のテクスト・データをOCRによって入力することによって、語彙分析と語彙集の作成が可能になります。逆に、作成された語彙集を参照辞書とすることによって、OCRの精度を高めることも可能になります。

第三には、東京大学に所蔵される一級の一次資料（貴重書など）の画像と電子テクストのクロスリファレンスの公開、流通が可能になります。例えば、古い絵巻物や草紙であれば、画像として原資料を示し、テクストで用いられた通りの漢字・用字法で文字を正確に電子化し、さらに分かり易い現代国語・漢字による注釈的テクストを添えるという、三段階のクロスリファレンスが考えられます。

欧米系文字コードについては、古代から近代までのラテン文字をすべて表記できるように拡張を行い、古ロマンス語文献の電子化を推進するとともに、『19世紀ラルース辞典』を素材として、データの大量入力を実験的に試みます。精度の高いOCRを利用した入力法が確立されれば、電子化テクストの集積が容易になり、学術的な「電子図書館」構想への展望が開けることになります。

古代ギリシャ文字については、米国カリフォルニア大学の＜Thesaurus Linguae Graecae＞プロジェクトが設定しているベータ・コードに対し、上位互換性のある文字コード系を画定するとともに、対照表を作成する予定です。

知的資産形成における国際協力

日本からの文化的知的資産の輸出が、商品に比べて少ないとかねがね批判されております。人文系のデータも国際的には本来は交換が原則ですが、日本から提供するよりは提供されるほうが多い現状のようです。

われわれが当面考えている海外協力研究機関とそのプロジェクトとしては、連合王国・大英図書館British Libraryにおける多国語検索システム、フランス・国立図書館Bibliothèque Nationaleの電子図書館構想、国立科学院CNRS INaLFの電子化テクスト・データ・ベース、Chicago University ArtFL Projectの同様の構想、米国カリフォルニア大学の<Thesaurus Linguae Graecae>プロジェクトであり、ギリシャ・ラテン文字系の文字コードの標準化の観点から、研究交流を行います。

また、中国、台湾、韓国などの漢字使用諸国における標準文字コード策定と実装の現状も調査し、関係機関との交流をはかり、その成果を随時漢字データ・ベースに収録するつもりです。

将来的には海外の日本語・日本研究者及び研究機関に「東大・漢字フォント・セット」を提供し、日本のデータ・ベースへのアクセスの便宜をはかりたいと希望しています。OSや機種の壁を超えた交換・交信可能性をいかに保証するか、互換性をいかにはかるかの問題を解決する課題がいぜんとして残ります。

多言語文字資産の継承に産学共同研究は可能か

すでに述べましたように、本研究プロジェクトは、吉川総長がご推薦くださり、日本学術振興会により「産学共同支援事業」の一環として承認されて、発足いたしました。はたして産業界にたいしてどれだけ直接的に貢献できるかはわかりません、というよりむしろ、非営利的な、実用性を超えた文化事業であると断言できます。総長のお考えは、人文系の研究者も知的資源を社会に対してより積極的に公開せよ、という問題提起だと拝察しました。

しかし、文化事業であっても、統一的な漢字コードが必要か否かと問われれば、今日早急に解決しなければならない、焦眉の急の問題であると答えざるをえません。約8万字の日本語漢字がコンピューターに載るか載らないかは、次世代への文字文化の継承が問われているように感じられるからです。危機意識さえ生まれます。

私自身の認識可能な漢字の字数は、1万字にも達しないのではないか、ということもよくわきまえており、専門からいっても頑固な漢字論者とは対極にいるとも思っております。では、倍の2万字あれば実際的に、効率的に、十分ではないか、という反論もすぐに聞こえてきます。それはそのとおりでありましょう。しかし、問題は便利か否か、効率的か否かではなく、文化的基盤として、歴史的に用いられてきた約8万字、あるいはそれ以上の数の日本語漢字の電子化データが、次世代に対して用意できるかどうかだと思います。誰が、いつ、どこで使うか分からないような、一生に一度も使わないような漢字まで、大げさに言えば、次の百年のうちに誰かが一回だけ使うかもしれないような漢字を、用意する気があるかどうかの問題です。また、逆に言えば、約8万字もの知的文字資産を、誰でも、いつでも、自由に使えるように、開放することこそが、技術革新の方向であり、大学の使命であるべきではないでしょうか。問われているのは、人文学系の研究者だけではないと思います。

（田村　毅）

世界規模通信基盤としての多国語処理

——既存多国語処理の破綻とTRONコードの考え方——

はじめに

現在、パーソナルコンピュータの急激な価格低下に伴う大量普及とデジタル通信回線の普及に伴い、インターネットに代表されるネットワークで情報をやりとりすることが世界的に注目を浴びるようになってきた。

特にインターネットの普及を助けているのが、World Wide Web (WWW)のような、ネットワーク環境においてマルチメディアデータをハイパーテキスト的に扱える枠組みである。NetscapeやMosaicのような手軽に使えるナビゲーションソフトウェアが開発されたことにより、インターネットは一段とその普及に拍車をかけ、今や今世紀ならびに来世紀の情報交換のプラットホームとしての地位を得ようとしている。

ところが、世界にこういうものが広まるにつれて、日本人のような大規模文字集合の言語を使用しているユーザが、WWWのような環境を使おうとしたときの問題点もあきらかになってきた。このような問題は日本以外でも、中国、韓国を始め同じような状況の国でおこっている。

日本では、漢字というアルファベットと大きく異なった大規模文字集合言語を使っているために、自国の言語の記述には特殊な2バイトコードの文字コードを使う。これによって、確かにコンピュータの画面上に自国語が表示されるようになる。

しかし、そういうような大規模文字集合を持っている国でWWWを利用すると、問題があることがわかる。最近のWWWブラウザでは、確かに日本語コードも扱えるのだが、日本語のWWWページを見て、例えばフランスのルーブル美術館のページを同時に見ると “Musée” が “Mus仔” に化けてしまうというようなことがおこる。

これは西欧のWWWユーザには起こりにくい問題である[注1]。英語のページとフランス語のページを同時に見ても問題はないし、ドイツ語のページでも見えるであろう。しかし、日本語ページとアクセント記号付きの文字を使うページを同時表示しようとすると、せっかくのウインドウシステムも有効に働か

注1 ラテンスクリプト系の言語のための1バイトコードの文字コードにも何種類もあり、‘é’などが含まれるMSBがオンの上位128文字に関してはコード系が異なると文字化けの可能性がある。しかし、HTMLではそのような文字は“Musée”のようにマンリーダブルなエスケープシーケンスで記述することが可能で、その方法で記述してあれば、最終的には表示側のシステムでそれを評価して適切な変換を行うことになっているため、問題がおきにくい。ただし、エスケープシーケンスを使わなかったりすれば文字化けしうる。また、ヨーロッパ系の言語に必要とされるアクセント記号付き文字のすべてを考えると1バイトコードにおさまらないため、たとえエスケープシーケンスを使っていても、表示用に指定されているフォントの上位128文字にもないアクセント記号付き文字を指定していればやはり文字化けする。

注2 'ö'は'o'としてソートして、つぎに'o'と'ö'を区別してソートしなければならない。

ず——現在のWindowsやMacintosh上のビューアでも、いろいろのテクニックを使うのだが——やはりどちらかが文字化けすることになる。

このような文字化けの理由は、システムがそういう大規模文字集合言語との混在に完全には対応できていないからである。インターネットの発展とともに、今後ますます世界規模での情報交換がさかんになる。8ビットコードでおさまるようなラテンスクリプト系の言語だけでなく、世界中の大規模文字集合を使う言語まで含めたすべての言語を取り扱える情報交換のインフラストラクチャの必要性が、インターネットの発展により、よりリアルなものになってきているのである。また、ISO10646-1という形で世界標準の文字コード体系が決められるという動きもあった。

本論文では、このような状況変化を背景として、TRONプロジェクトの多国語処理[1]に対する考え方を述べるとともに、その観点から現在の考えられている多言語処理系を評価しその問題点を指摘する。また、TRONの考え方に従った具体的な多国語処理システムの提案ならびに、その実現のために私たちが行っている作業を紹介する。

TRONはこう考える

多国語処理に関するTRONの考え方は以下のようなものである。

アルゴリズムを含めて考える

多国語の混在した文書を取り扱うことを考えた場合、単にコードに複数の言語の文字を割当てるだけでは不十分である。英語は左から右へ文字が書かれるが、アラビア語は右から左である。行が変わる場合、日本語では原則的には文字のどこで切ってもよいが、英語では単語の途中で切ってはいけない。そのかわり日本では行頭、行末の禁足がある。数値や金額の表記法も言語により異なる。

このように言語に対して、文字の形の違い以外の言語独特の規則——正書規則があり、文書の正しい表示を行なうためにはこの規則が適応されなければならない。細かく言えば、正書規則がなければ文字を表示できないといってもいい。

また、例えば、日本語ならば入力時にかな漢字変換のような特殊なアルゴリズムを使用するし、ドイツ語において'ö'は'o'の後であるが、'König'は'Konzert'より前であるというように特殊なソーティングアルゴリズムが必要になる[注2]。このように、言語により入力アルゴリズムやソーティングアルゴリズムなどの切り替えも必要である。

従って、TRONでは単に文字コードだけでなく、その文字を使う言語に関係するアルゴリズムを含めて多国語処理を考える。このため、多国語を扱うTRONの規則を単に多国語コードと呼ばず、TAD多国語環境と呼んでいる。

OSのレベルで考える

先に述べたような文字化けの回避についていえば、個々にアプリケーションレベルで対応することも可能である。WWWでの文字化けについていえば、HTMLの拡張とブラウザ側での対応により回避することは可能であろう。しかし、このようなアプローチは望ましくないというのがTRONの考え方である。

ネットワークから取得した情報の他のアプリケーションによる再利用や、さらには異なる正書規則の文字列表示機構の組み込みまでふくめて考えた場合、個々のアプリケーションがそれぞれ似たような言語処理機構を独立に複数持つことになり効率が悪い。TRONではHTMLのような上位プロトコルではなく、文字コードのレベルで多国語に対応し、表示や編集

プリミティブをふくむ処理機構をシステムレベルでサポートすることを目指している。

すなわち、オペレーティングシステムは多国語対応となっていて、アプリケーションは言語と独立して作ることができなければならない。これにより、アプリケーションプログラムを言語独立として、国際的にソフトウェアが流通できるようになる。

公平に考える

TRONプロジェクトは「どこでもコンピュータ」[2]という未来のコンピュータ環境を前提に、そのための望ましいインフラストラクチャの構築を目指したプロジェクトである。「どこでもコンピュータ」環境におけるコンピュータは真の意味で「誰でも使える」コンピュータでなければならない。その「誰でも」の言葉のなかには、子供から大人なまで、健常者から障害を持った人まで——そして、もちろんどのような自国語を使う人も含まれるのである。

今後21世紀に向かって、情報交換はますます世界的レベルで盛んになる。そのときに、使用している言語により情報格差が生じるようなことは許されない。8ビットコードでおさまるようなラテンスクリプト系の言語だけでなく、世界中の——大規模文字集合を使う言語まで含めた——すべての言語をフラットに取り扱える情報交換のインフラストラクチャをTRONでは目指している。

理想を考える

多国語処理へのアプローチについては、まったく例がないわけではない。例えば多言語コードについていえば、ISOをはじめとして関係機関によりISO10646-1などという世界文字コードも作られている。ところが残念なことに、それらのアプローチにも多くの問題が含まれている。過去との互換性を考えるあまり小手先の多国語対応に終わったり、コンピュータ側の処理上の都合を優先するあまり本来の要求仕様を曲げてしまったりという姿勢が、これらのアプローチに問題をのこす結果となっている。

これに対しTRONではインフラストラクチャの改善を行うならば、基礎になるものほど徹底的に考えて理想を追求するべきだと考えている。基礎になるものほど変更時に既存システムに与える影響が大きいからである。

影響が大きいからこそ、不満足なものでは、切り替える価値があるのかといったことが問題になる。切り替えリスクに見合わないということだけでなく、問題が先送りされただけで結局すぐにさらなる改善が必要になるならばリスクがまったくの無駄になってしまうからである。変更するとしたら、そのリスクに見合うだけの利点があり、さらに将来にわたってまで使い続けられるものでなければならないと、TRONでは考える。

WWWはほんの数年で、しかも全世界レベルで、インターネット環境を変化させてしまった。ネットワークの急激な発達は、ただでさえ激しいコンピュータ関連技術の変化スピードをさらに加速するであろう。変化のスピードは、インフラストラクチャのライフサイクルの問題をより切実にする。理想を前提としない、不満足な改善では一夜で状況に取り残され、新たな要求に対する足枷に変わりかねないのである。

何が問題か

つぎに、このようなTRONの考え方を前提に、既存の多国語処理系を見てみよう。

現在、多国語処理系として最もまとまっているのは、Apple社のMacintoshであろう。Macintoshはシステムレベルで多国語をサポートしようとしているし、

ある程度それに成功している。

また、文字コードだけについていえば、Apple社も含めた米国のコンピュータメーカーの主導で決められたUniocdeがあり、これがISO10646-1となった。Macintoshでは1996年リリース予定の次期OSでこれを採用する予定である。

Windowsの多国語処理について言えば、Windows-NTではすでにに採用しており、システムレベルで多国語をサポートについても、早晩Macintoshの後追いをすることになろう。

そこで、以下では世界文字コードとしてのISO10646-1と、多国語処理系としてのMacintoshのシステムを検討する。また、これらと別のアプローチとしてUNIXでの多国語処理についても触れる。

ISO10646-1の問題点

原則1文字32または16ビットの固定長で、かつ、ISO2022系のコードのように文字セットの切り替えがないのがISO10646-1の最大の特長であり、処理系を作る側から見たメリットである。しかし、次に述べるように実際にはこれらの利点は崩れている。

●背景

1992年6月、ISOはそれまで日本側も推していたDIS10646第1版を捨て、マイクロソフト社などの米国コンピュータメーカーが推していたUnicodeを中核とするDIS第2版を未来の世界標準文字コード体系として決定した。

過去5年間にわたりディスカッションされてきたDIS第1版が否定された理由は、米国のコンピュータメーカーが作る側から見たシンプルさを重視して、いちはやく8ビット複数バイトコードのDIS第1版を使わないと宣言したためである。実質的に世界のほとんどのコンピュータを支配している米メーカーがUnicodeを使えばそのデファクトスタンダードとISOが並立してしまう。そのような事態を避けるため、16ビット固定長コードのUnicodeとDIS第1版の統合化作業が行われ、Unicodeとほとんど同じ16ビットコード平面をトップに持つDIS第2版が作られ、ISO10646-1として可決された。

ISO10646のコード体系は本来、群、面、区、点とよばれるそれぞれ1オクテット、合計32bitのコードであらわされる。単純計算でここに収容できる文字数は43億種類あり、巨大である。しかし、今回規定されたISO10646-1は基本多国語面と呼ばれ、先の巨大コードの第0群、第0面に相当する部分のみである。そして基本多国語面のみの使用には2オクテットで表現するUCS-2という短縮形式を許しており、現実的にはUCS-2がISOの多国語コードとして使われる。

ISO10646-1 (part-1) は、この全体構成と、基本多国語面の中身を規定しており、他の面の中身については、Part-1以外の別のpartを新しく作って追加することになっている。

しかし、この基本多国語面は実はUnicodeをほぼそのまま採用したものであり、UCS-4に切り替える方法も決っていない。そのため、実質上はISO10646-1はUnicodeと同等になってしまっているといってもよい。

しかも、今後partが増えてUCS-4が決っても対応するかどうか未定であるという態度のUnicode推進メーカーも多く、将来も実質上はUnicodeという状態が続くことが十分考えられる。そのため以下では、ISO10646-1とUnicodeを同等のものとして話を進める。

●ユニフィケーションの問題

ISO10646-1ベースの文字コードの問題はいろいろあるが、本質的な問題は、本来16ビットで表現できる文字数（65536字）[注3]に入りようもない世界の文字を、その中に詰め込むために多くの無理をしているということである。

注3 16ビット=1バイトとし8ビット単位での処理を考えないとすることで、従来の8ビット=1バイトの2バイトコードでは使えなかった上位もしくは下位バイトが通信用コードと重なる部分も文字表現に使用できる。

その詰め込みを可能するトリックが、ユニフィケーションである。

ユニフィケーションは本質的に同じシンボルなら言語の枠を越えて一つにまとめてしまおうということである。これによって、16ビットの枠内に世界のほとんどの言語のための文字を納めることができるというのである。

しかし、今まで文字学の成果として、消滅した文字を含めれば約400種類の文字セットが世界中で知られており、これをたった16bitで表現するのはそもそも無理がある。漢字では特にこの影響が大きく、中国、台湾、日本、韓国の国内規格の合計81635文字からの中から54015文字を選び、さらに由来が同じあるいは多少の違い無視して20902文字に統合するということが行われた。しかも元になったこの国内規格も十分なものではない。例えば、日本最多の収録字数の大漢和辞典には4万9千字収録されているし、中国の漢字にいたっては、現在編纂中の漢字辞典に6万字以上収録されているといわれる。

結局、Unicodeの文字割り当てのあらゆる基本ポリシーがその理想から外れ、単に文字セットを小さくするという基準で制定されることとなった。

しかも、その妥協と無理の仕方がラテン語圏（さらにいえば英語圏）の人間のセンスで行われているため、我が国のような大規模文字集合を使う国や、彼らにとって縁の薄い東南アジア付近の文字セットの扱いに非常に不満が多いという結果になっている。

例えば、ラテン系言語のAとギリシャ文字のAとキリル文字のAなどアルファベット系では、異なるスクリプトに属するということでいくら見た目が同じでもユニファイせず別のコードが割り当てられている。これに対し、日本語と中国語や韓国語の間では、明らかに見た目が異なる漢字までルーツが同じということで無理やり同一コードにユニファイしている例がいくつもある。

中国と韓国と日本の漢字では同じルーツから派生して今は見た目が違ってしまった文字もあるし、逆にいつのまにか同じ見た目でも意味の変わってしまった文字もある。このようなものをとにかく16ビットに納めるという前提条件のもとに無理やり同じコードにユニファイするというのは強引である。

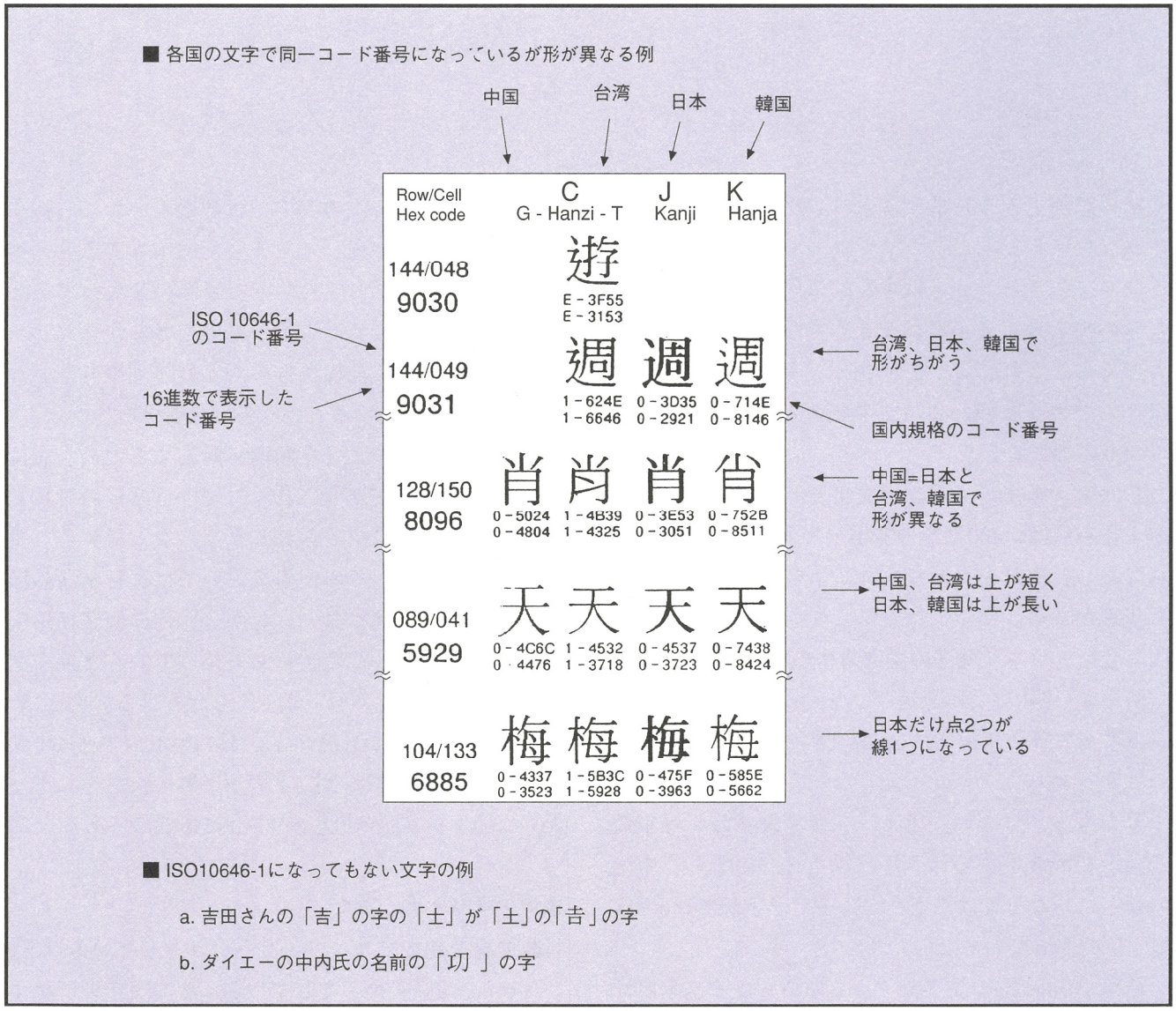
実際には、このような問題に対応するため、各国語プロファイルが用意され、コードと字形の具体的対応を国語ごとに決められるようになっている。これにより日本国内での使用においては、日本語の漢字が表示され問題はない。しかし、他の漢字使用国のコンピュータにこのデータを持っていくと違う文字に化けてしまう、ということも起り得る。例えば、日本語の「辻」はISO10646-1では8FC0で、これは中国のコンピュータでは「辻」と表示される。さらにやっかいなのは日本語の「辻」が8FC2にあるのである。

●何の言語かわからない

言語の枠を超えてユニフィケーションしている以上、一度Unicodeに変換してしまうと、その文字の所属する文字セットの情報は失われてしまう。

しかし、正書規則の決定を行うためには、使用する文字セットの決定だけでなく、それが何の言語で書かれた文章であるかの情報が必要である。「アルゴリズムを含めて考える」で述べたように、それがなければ、文面を正しく表記することはできないし、ソーティングもできない。スペルチェッカを使う場合などでも、言語がわからなければ、単なるスペルミスなのか、フランス語の引用なのかの判別がつかない。また、マルチメディア化に従い文章情報を音声合成により、読み上げさせるといった機構も今後、汎用的に実現されるようになるだろうが、この場合にも言語情報が必須である。

従来は各国語専用のワードプロセッサを作ることにより、アプリケーションレベルでこれを回避する



問題例

ようなアプローチも多く見られたが、世界規模の通信環境を考えた場合、WWWブラウザのような一つのアプリケーションで、送られてくる各国語のデータを正しく表示することが求められる。このため、送られてきたデータが、なんの言語で書かれた文章であるか、データから判別できるように——それも、同一文中での混在使用まで含めて、判別できるようになっている必要がある。

さらに、多くの人々がインターネットを利用するに従い、自動翻訳の需要はますます増えているが、これもデータに言語情報がなければ不可能である。

●やっぱり足らない——だから外字

私用コードである外字が多くの問題を持つことについては、異論のないところであろう。通信データの回復不能な文字化けの原因になるだけでなく、共同利用データベースにおいては、同定不能のキーワードの原因になる。

今後、WWWのような通信環境の発展により、異なるソースから引き出した情報をマルチウィンドウで一つの画面上に混在表示するということが増えてくるであろう。このような状況では、異なる外字が同一のエントリーポイントに割り当てられたデータ

を同時表示するといった状況も考えられ、外字データを事前にダウンロードして登録する方法でも文字化けは回避できなくなる。

しかし、ISO10646-1では外字が容認されている。しかも非常にまれにしか必要とされない例外手段ということではなく、仕様の不備を埋めるために実際に使わざるをえない状況になっているのが実情である。

例えば、特定地方自治体で必要とされる文字は、ISO10646-1の内字ではおさまらない。この多くが地名、人名等の固有名詞で使われながら、内字にない文字である。最も簡単な例としては、日本人の人名でよく使われる「吉」の字があげられる。人名では「吉」の字の上が「土」と「士」を区別して使われるが、従来のJISコードはもとより、ISO/IEC 10646-1でも「吉」は使えない。

無理なユニフィケーションを行ったにもかかわらず、ISO10646-1で定義する文字では文字が足りず、救済策として6400字の外字エリアを設けているのである。

日本国内で使う外字についてこの取り扱いを標準化しようという動きも一部にあるが、韓国や中国などでも状況は似通っており、それぞれ独自にこの外字エリアを必要としている。韓国で希にしか使われないハングル文字に外字エリアを使おうとしているなど、これらの外字エリアの利用法は統合不可能であり、これを標準化することは作業的な問題ではなく、物理的に不可能である。

つまり外字の問題は、世界規模で考えた場合、文字割り当て作業が行き届かないという問題よりむしろコードエリアにあるエントリー数が足りないとい

	国内規格の名称	国内規格等の漢字数	収容した漢字数
中国	GB2312	6763	6763
	GB12345	6866	2192
	GB7589	7237	4835
	GB7590	7039	2842
	現代漢語通用字表	7000	42
	GB8565	7053	290
	香港文字	58	58
	吏読	92	92
台湾	TCA-CNS11643の1面	5401	5401
	同上非漢字	9	9
	TCA-CNS11643の2面	7650	7650
	TCA-CNS11643の14面	6319	3951
	CCCII規格	237	237
	広東語用文字	10	10
日本	JIS X0208	6355	6355
	同上非漢字	1	1
	JIS X0212	5801	5801
韓国	KS C5601	4888	4620
	KS C5657	2856	2856
		81635	540152

ISO/IEC 10646-1(JIS X 0221)国際符号化文字集合基本多国語面（UNICODEとも同様）に収容した漢字

うのが、その本質である。

●拡張の余地がない

エントリーが現状でも足りないということである以上、将来の拡張の余地は当然ほとんどない。今後、発展途上国が自前で文字コードをつくったり、点字などの特殊用途の文字セットを導入しようとしてもうまくいかない。

ISO10646としては、これらにはUCS-4で対応するということであろうがISO10646-1のUCS-2からの切り替えリスクを、少数者に押しつける結果となる。ハードウェア資源等で先進国ユーザより厳しい状況の人々に、彼らだけの言語ならまったく不必要であろう4バイトコードを押しつけるのは、「言語に対しフラットな多国語処理とする」というTRONの考えからいって容認できるものではない。

●似た文字はあり過ぎる

無理して文字数を減らしているISO10646-1であるが、逆に類似文字の過剰という問題も指摘されている。

例えば日本語で使われない漢字の場合には、他の国のフォントが表示されるわけだが、これが日本語で使う他のコードの漢字と似ている場合、片方が本来日本語にない文字であるだけにその見分けは難しい問題になってしまう。個々の言語のなかだけでたがいに見分けの付くように変形してきた文字セットを、合体させたことによる問題である。

また、ラテンスクリプトでも文字のアクセント記号つき文字は、元の文字とアクセント記号の2文字分のコードから合成で表示でもきるが、よく使われるアクセント記号付きの文字には独立したコードが別に存在する。このため、見た目は同じ文字でも単独のコードのものと、合成で作った文字と二種類の表現が可能になる。

このように、文字に対してコードが一意に決らないということは、データベース検索など多くの点で混乱のもとになる。

●実際には不定長コードである

コードエントリーが足りないため、ラテンスクリプトの言語では、需要の少ない言語の文字でアクセント記号付き文字が独立に存在しないものがある。このような文字については、合成が必須となり不定長コードになってしまう。また、韓国のハングル文字でまれにしか使われないものなども合成で作るということになっており、多くの言語の処理系が固定長コードのメリットにあずかれない。

これも無理なユニフィケーションまでした結果が、無駄になっている例である。

●誰のための、何のための文字コード？

このような問題点の指摘を見ていくと、ISO10646-1はその仕様書で示している、当初のゴール（宣伝文句）すら結果的に満たしていないことがわかる。仕様書には「プロフェッショナルな組版やDTPを満たすのに有効な…」とあるが、プロ用の組版ソフトには不十分だし、不定長の原則すら壊れている上に、希な文字の必要性がもっとも高いデータベース応用には表現できる文字が足りないということで、どっちつかずのものになっている。そして、そのわりには弊害が多い。

Unicodeでは言語情報を持たず、インプリメントに問題を先送りしているため、結局インプリメント側で、フォント切り替え機構を流用するか、もしくは言語指定のエスケープコードを導入するといった対応をすることになり、無理なユニフィケーションまでして導入した非エスケープ固定長コードは結局は成り立たなくなる。そのうえ、言語切り替えをインプリメント依存にしたため、システム間でのデータ交換では言語情報が失われ、正しく表示できないこ

注4 本来のJIS漢字コードは2バイト固定長コードであり、アルファベット用の1バイトコードとは、エスケープシーケンスで切り替えることとなっていたが、パソコンではこれを嫌い、MSBが1のバイトの半角カナ外のコードは次のバイトとあわせて2バイトコードとして評価するという可変長のコード（いわゆるShift JIS）を採用した。これが、文字化けの大きな原因となっている。

注5 ワードプロセッサ等では、欧文フォントを使用している文字列は和文フォントへの変更を許さないとか、フォントの変更時には常に和文フォントと欧文フォントをペアで指定させるといった対応をしているが、アプリケーションレベルでの対応にすぎず、他のアプリケーションとデータのやりとりをするなどするとやはり文字化けがおこる。しかも、その文字化けを直そうとすると、今度は変更がゆるされないなど、処理が複雑なわりには弊害が多い。

とが十分考えられる。

結局、ここまで原則を曲げ当初目指したメリットの多くを不完全にしてしまったコード系を、既存コードを捨てて、コード変換とシステム更新のリスクをかけて、将来にわたってまで使い続ける次期標準として採用する価値があるのかといったことが真の問題といえる。

人間がコンピュータに合わせるのでなく、コンピュータが人間に合わせるべきだという思想が認められるようになってきた現代。言葉つまりは個々の文化をこわさないように体系を決めようというのでなく、コンピュータにとっての処理のしやすさという観点のみから強引に16bit固定コードという枠組みを決め、それで文字が収まらなければユニファイしてしまえ、というISO10646-1の思想そのものが、すべての問題の原点なのである。

インターネットがあぶり出すパソコン多国語処理の欠陥

●Macintoshの多国語処理

現在Macintoshを始めとするパソコンが日本語処理のために採用しているShift JISコードはエスケープシーケンスなしでアルファベットと日本語を混在する可変長コード[注4]でありコードから言語の特定ができない。そのため、多国語処理を行うには言語の特定機構を独自に持たなければならない。

Macintoshは早い時期からマルチフォント切り替え機構をシステムレベルでサポートしており、この機構を言語処理環境の切り替えに利用している。文字列に日本語対応のフォントが割り当てられていれば、それを日本語として解釈するというように、言語とフォントを対応させているのである。

これは、Macintoshの日本語化の時に、主に過去との互換性の維持のために行われたことであるが、次期OSでISO10646-1を採用しても、すでに述べたように言語情報が失われるという状況は改善されない。

●フォントと言語の混同

このような、フォントと言語の混同が理想から外れたものであることは明白である。例えば、フランス語の単語を引用している日本文を選択し、フォントを他の和文フォントに変更すると、先の“Mus仔”のような文字化けが起こってしまう[注5]。

概念的に考えても本来フォントは言語とは独立している。現状のシステムでも使用フォントにより決定できるのは文字セットであって、言語そのものではない。例えばISO8859-Part.1のラテンアルファベットNo.1を使えば、英語だけでなくフランス語やドイツ語も表記できる。しかし、先に述べたようにソーティングアルゴリズム等を切り替えるには、文字コードがISO8859-Part.1であるということ以外に、それがドイツ語の文章であるという情報が必要である。また、それとは逆に日本語のローマ字表記のように、言語は同じだが、使用する文字セットが異なるといった場合もありえる。

ISO10646-1を採用すれば、フォントの言語独立性はよりはっきりするはずである。例えば、日本語と中国語の混在した文章があった場合、同一のフォントを使っては言語情報が失われる。そのため、ISO10646-1の主旨からいってほとんどが同一のはずの文字について、二系統のフォントを持つ必要がでてくるのである。そして、フォントの変更においても見た目でなんの変化もない文字について、和文フォントと中国語フォントを混同しないように注意して変更する必要がでてくるのである。

さらに、先に述べた通信環境から送られてくる情報を自動翻訳して表示するといった場合、かならずしも送られてくる言語のフォントはシステムに登録されている必要はない。フォントが登録されていな

注6 アプリケーションによっては、それぞれのインプットメソッドごとに使用フォントを記憶しているものもあるが、例えばTimes等のセリフ系の欧文フォントを使っていて、日本語インプットメソッドに切り替えるとそのデザイン的な選択は反映されず、サンセリフ系の和文フォントで続きの文字が表示されるということになる。より高度な、ワードプロセッサなどでは、フォントを常に欧文フォントと和文フォントの組で指定させたりするが、この対応も第三の言語があれば破綻する。

いと言語が判別できないのでは、問題がある。

また、ドイツ語のデータを英文に翻訳するといった場合は、フォント上は両方とも欧文フォントであるため、区別できないというのも問題である。

●バラバラの言語指定

Macintoshでは言語に関する指定が3系統ある。一つは先に述べたフォントを利用しての、言語指定である。もう一つはインプットメソッドと呼ばれる言語依存の入力方式の指定である。そして、最後がシステムのコントロールパネルでユーザが明示的に行う「数の書式」や「日付&時刻」の指定や、システムマクロ言語であるAppleScriptでの「使用言語」等の個々の指定である。

Macintoshではこれらの言語依存部をパッケージ化して、システムに追加登録できるように考えられているのだが、これら3系統の指定間での連携はうまくとれていない。

例えば、欧文フォントで英語の文章を入力していて、インプットメソッドを日本語に切り替えるには、まず使用する和文フォントを選択するといったことが必要になる。これを行わなわずインプットメソッドのみを切り替えた場合の結果については、完全にアプリケーション依存である。最悪の場合はアクセント付きアルファベットからなる、意味不明の文字列が入力される[注6]。

また、数の書式などについていえば、例えば英語とドイツ語では小数点に'.'を使うか','を使うかという違いがあるが、フォントでは区別できないのでユーザが指定した書式が使われ、言語に従って自動整形したい場合であっても表記を変えることはできない。

上記の例でもわかるように、連携の欠如も基本的にはフォント指定を言語指定に流用しているための問題といえる。

UNIX系の対応

インターネットの世界——というより、それを支えるUNIXの世界では、MIME[3]やMule[4]のように、転送データの先頭に使用する文字コードの宣言を付加し識別できるようにしたり、処理系の方でも他言語を含め言語コードを自由に追加できる枠組みを作る方向へ向かっている。つまり、多国語処理について、新規の体系を作るのではなく、既存の体系を混在使用できるシステムにより、問題に対応しようとしているのである。

多様な仕様の混在を前提とするUNIXらしい解決方法であるが、同時に豊富な計算資源を前提とするという意味でもUNIXらしい解決方法であるといえる。しかし、最近のコンピュータの性能向上は目覚ましく、パソコンが少し前のUNIXワークステーション以上の性能と容量を持つ現状では、このような解決方法はパソコンにおいても十分なりたつのである。例えば、最悪のケースで2バイトコードを2バイトコードに任意に変換するとしても、写像テーブルの容量はたかだか128Kバイトであり、現在のパソコンにとっては大きな負担になるものではない。

このようなコードの相互変換を前提としたような「現状追認型」のアプローチでも十分成り立つとしたら、ISO10646-1のような不満足な改善の価値はますます低くなる。事実UNIXの世界では、ISO10646-1も追加するべき新しいコードができたという認識で、ISO10646-1に標準化しようという動きはない。

「現状追認型」のアプローチでもできる程度の改善であれば、新たなコード体系は必要ないし、それでさしあたりの要求に対応している間に、より理想的な次期標準を打ち立てるべきである。

実はこのように各国の既存文字コードをエスケープシーケンスで切り替えるという枠組みは1986年のISO2022ですでに考えられていたものであり、DIS 10646第1版はこれをベースとして多国語処理を行おう

としていた。ISO10646は本来ならばのこの方向へ行くはずであった。それが、処理系の作りやすさを考えてUnicodeベースのISO10646-1になり、結局含むべき文字の多量さにより、ユニフィケーションの理想も崩れ、インプリメンテーションでの問題や、各国語プロファイルによる標準の発散などにより、「不満足な改善」になった顛末は、すでに述べた通りである。

TRONの多国語処理環境

TRONの目指すもの

TRONでは多国語処理を考えるにあたり、満たすべき目標として、以下のような方針をたてた。

●外字を必要としないシステムとする

●全世界、古今東西のあらゆる文字を取り扱えるようにする

TRONでは文字として、使用頻度や利用者数で差別をもうけることはしない。また、あらゆる文字の中には、現代活字として使われる文字だけでなく、俗字やヒエログリフなどの歴史的な文字などもカバーする。現在利用者が一人もいない歴史的な文字でも学術データとしては必要である。

さらにいえば、経済性重視のニーズなどによって利用する文字を制限するというのは運用上の話であり、あらゆる文字を区別してコード化することとは別の問題である。コードの目標としてはあらゆる文字を取り扱えるようにするべきである。あらゆる文字が割り当てられたコード体系で運用上利用を制限することは容易に行うことができるが、制限された文字しか割り当てがされていないコード体系で割り当てられていない文字を扱うことはできないからである。

またこの目標は、ある意味で「外字を必要としないシステム」という目標から必然的に導かれる目標である。なぜならば、外字を必要とする場合の多くが、自分が使いたい文字がコード中にないということからきているからである。

●使用されている言語が判別できるようにする

●文章記述と文字表現の分離

文章記述とは、文章としてどのように記述されているかであり、文字表現とは文章記述を文字で印刷あるいは表示するときにどのように表現されているかを示す。

具体的には、合字の取り扱いで文章記述では'f'と'i'は二つの文字として汲い、文字表現では、状況により二つの文字としたり、'fi'の一つの文字としたりする。このように文字表現は、表示形式や書体により変化する。従って文章記述と文字表現を分離して考えることが重要である。

文章データとして蓄積するのは文章記述であり、文字表現は表示や印刷の時に文章記述から変換される。文字表現ではなく文章記述によりデータベースを構成することにより検索が組み版とは関係なく行うことができ、特に有効である。

●可能な限りフォントデータ容量を少なくできるようにする

現在の多国語処理において容量負担という意味でもっとも問題になるのは、必要とされるフォントデータである。ISO10646-1をサポートすれば、スケーラブルフォントの場合一書体あたり10Mバイトを越える負担になると考えられる。

これに比べ文字情報自体は、大きな文書でも高々数百Kバイトの世界であり、欧文データがISO10646-1に変換して2倍の容量になっても大きな問題はない。また、最近はファイル格納時や伝送時にデータ圧縮が行われることが多く、この場合拡張分の情報は圧縮され大きな負担にはならない。

そこで、TRON多国語文字環境では、可能な限り

フォントデータ容量を少なくできるように考える。

TRONならこうする

上に述べた目標を実現するために、TRON多国語文字環境は以下のような方針で設計することとした。

●文字コードのエントリーには事実上無限の容量を

コードエリアにあるエントリー数が足りないというのが、外字問題の本質であり、これからのネットワーク社会を前提とすれば文字収集の作業はむしろ、従来よりはるかに簡単になるであろう。例えば古代文字を必要としているグループがあれば、協力してエントリーを割り当てるといった、共同作業が、ネットワークを通して、大人数ですばやく効率的に可能になる。

代表機関が一つのコード系をメンテナンスし、流通すべき文字として提案されたものを検討の上登録し、ネットワークを通してコードを割り当て公開するといった運用により、外字という概念をなくす事ができる。

●図形を文字として埋め込める

全世界、古今東西の言語を取り扱えるようにするという目標を考えると、新しい文字の登録の必要が完全になくなるということは考えにくい。新たな俗字が生まれることも考えれば、コード割り当てされていない文字の可能性は常にある。

このようなコード割り当てされる前の文字需要に対しては、一時的な処置として図形文字という手法をつかう。TRONの標準データフォーマットであるTAD [5]では、任意の大きさの図形を文字列の中に混在させることができるため、新たな文字を文字の大きさの図形として文中に埋め込む。これにより送受者間の合意はなくとも見た目に同じものが相手に伝送できる。

他にも、会社のマークや、マニュアル等でその文書のみでの特殊記号を文中に埋め込むといった、私的な図形の文字的利用にあたっては図形文字を使用する。

●言語が何か、使用文字セットは何かちゃんと宣言

多国語処理システムでは言語の切り換えを明示的に行なう必要がある。従って、データ中に言語指定情報を挿入し、任意の文字列がどの言語であるかを明示できるようにする。

言語指定では言語を指定するとともに使用する文字セットも指定する。これにより、言語に複数の表記がある場合にも対応できる。例えば日本語は漢字仮名混じり文での表記以外に、アルファベットを使ったローマ字表記も可能である。他にもウイグル語のようにラテンスクリプトでもロシアスクリプトでも書かれるなどこのような例は多い。

このため、言語指定は言語と文字属の組み合せをエンコードしたものとなっている。この指定のための言語指定コードは可変長のコードとして、バイト単位の拡張により事実上無限の言語と文字属の組み合せの識別が可能であるようにする。

●4つの階層構造で言語情報と効率化の両立

本コード体系では、特に電子的な取り扱いのときに効果を発揮するように次の4つの階層構造を持っている。

上位の2階層は言語オリエンテッドであり、下位の2階層は表記オリエンテッドである。つまり下位の2階層は言語独立であり、ISO10646-1と同様に言語の枠を超えてユニフィケーションも行われる。大きく分けたこの2層間でマッピングを行うという機構により、言語情報を失うことなく、フォントの言語独立やフォントデータ容量の効率化といったISO10646-1の持つユニフィケーションの利点も持つことができるのである。

・言語層（アルゴリズム）

言語層は、言語の違いによる各種アルゴリズムを持つ。この中には入力、組版(文字属層からスクリプト層へのマッピングを含む)、並べ替え、日時や数値の表記などのアルゴリズムが含まれる。

・文字属層（文章記述）

文字属層は、文章記述をする層である。つまり組版による見た目とは関係なく文章の内容として記述する。

具体的には、文章記述に使用する文字セットが決められる。

言語の特定は言語層で保証されるので、文字属層の文字セットについては効率を重視して配分する。数字やアルファベットや約物等の言語を超えて共通に使われる文字などは、各文字属ごとに配置するなどして、文字セットの切り替えを最小限にする。

文字セットは2バイトの空間とし、大規模文字集合言語では使用する文字が複数の文字属に渡ることを許す。一方、ラテンスクリプト系の言語など、使用する文字数が2バイト空間に満たない言語は詰め合わせ、同一文字属で複数の言語の表現が可能である。

・スクリプト層（文字表現）

スクリプト層は、画面や印刷上で見える形の合字などを含むすべての文字を扱う。

漢字文字属に対応する漢字を収容するスクリプト層では、これらの文字を一括して扱えるようにし、見た目により同一形状の文字は統合する。ただしエントリーポイントが2バイト空間に限定されず十分な余裕があるので、ここでの統合はUnicodeやISO10646-1で行われたような字数を大幅に削減するための無理な統合ではなく、見た目に差があるものについては統合しない。

スクリプト層は2バイトの面複数とし、ここにすべての文字を登録する。

スクリプト層はおもに文字表示機構の内部で仮想的に使われる存在であるが、特殊なTADデータとして、スクリプトコードを直接記述することも可能で、その場合スクリプト層は2バイトプレーンを面指定コードで切り替えて使用する。現在までの調査で、3バイトのコード空間があればすべての文字を格納して余りあるとなっており、面指定コードは通常2バイトである。しかし、これも言語指定コードと同じくバイト単位で拡張可能で事実上無限の収用能力を持つ。

スクリプトコードの直接記述は、組版後の固定した版面データを送るときになどに使われ、このデータを元に言語に基づいた自動処理を行うことは考えない。

・フォント層（デザイン）

フォント層は、スクリプト層の定める同じ字形に対して、たとえば明朝、ゴチック、Times、Helveticaのようなデザインの異なる複数の書体を扱う。

フォントデータとしてはスクリプト層の文字すべての図形データを必ずしも用意する必要はなく、たとえばアルファベットのみのデータとか、中国語専用のデータといった単位での流通を許す。ユーザは各自のシステムに必要に応じてこれらのフォントデータを登録する。

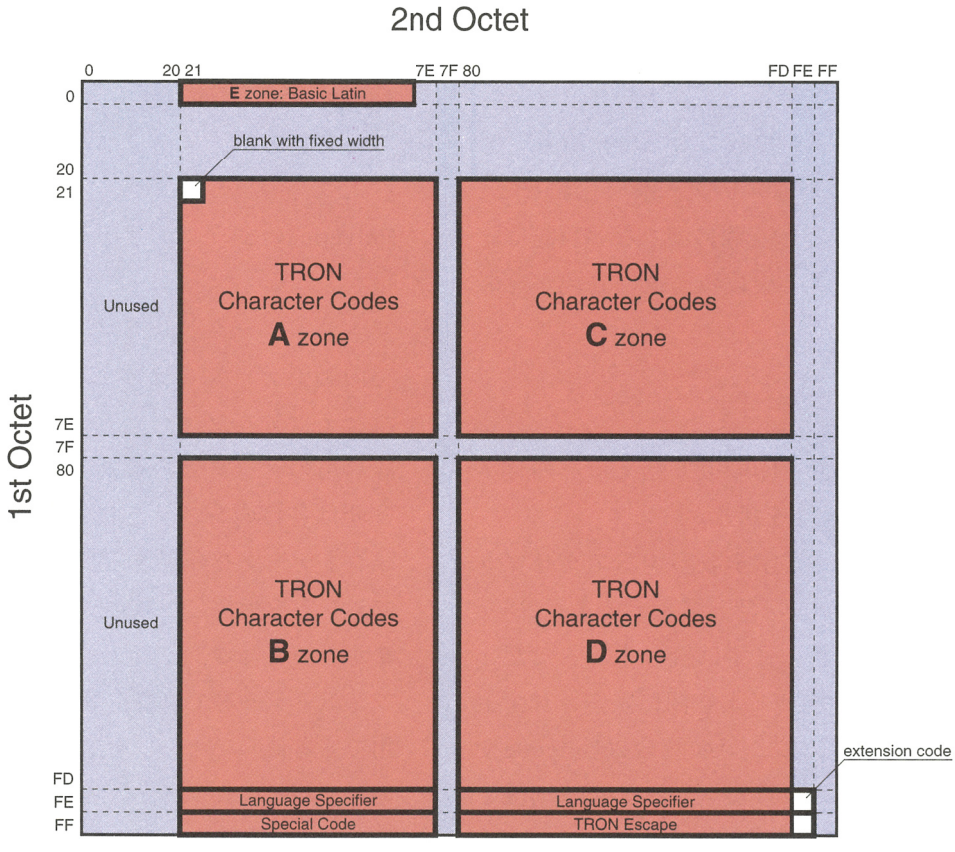
フォントを指定したときに、表示したい文字のデータがない場合は、他のフォントのデータを流用する。フォントデータでは特定コードエリアに関しては代用できる他のフォントを指定するといったことも可能とする。これより、仮名文字のみは独自の図形データを持つが漢字は他のフォントを利用するといっ

たコンバインフォントが可能になる。
それ以外に、TRONではフォントデザインを特徴数で表したデータをフォントデータに付加する。特徴数を比較することで、似たフォントを探して流用することができる。例えばアルファベットを含まない「明朝体」フォントを、英文の混じった日本語文に対して指定した場合、英文の部分にはよく似たTimesのフォントが適用されるといったことが可能になる。
このような流用は、システムでのフォント登録状態により使われるフォントが一定せず再現性がそこなわれるため、精密な組版応用には向かない。しかし、WWWブラウザのようにそのときに表示されればいいという応用には有効である。

●文字データベースの整備

文字属層からスクリプト層へは、合字など特殊な組版アルゴリズムによるもの以外は、マッピングテーブルで多対一に写像する。このマッピングテーブルは、対応関係以外にも各種の属性を保持した文字データベースの一部である。

文字データベースは言語層に付属し、属性情報として俗字、簡略字、合字等の文字の種別、他の字から派生した異字体の場合は元の文字へのリンク、漢字の部首、画数、読み等の大規模文字集合で文字の検索に使用できる各種のキー、さらには他のコード系に対応する文字がある場合のコードなどを文字ごとに保持している。日本の漢字については、学習漢字、常用漢字等の使用制限用の漢字集合の所属情報



TRONコードマップ

も持つ。これにより、文字表現時に小学校3年までの文字しか使わないで表示するなどの応用も可能となる。

このようなデータベースは、従来のパソコンでは漢字入力ユーティリティが個別に持っていたものであるが、汎用性の高い情報であるとともに、自動組版時やコード変換等でも利用されるため、システムサポートすることとした。

●多国語ライティングシステムの導入

すでに述べたように、WWWなどで送られてきたデータの使用言語に対応する、言語環境がシステムに組み込まれていない場合、スクリプト層に必要な文字セットが存在しても、実際の表示は行えない。

このような問題にある程度の解決を与えるため、TRONでは多国語ライティングシステムを開発した。多国語ライティングシステムは限られたパラメータにより言語の正書規則を表現する。このため、転送データとともに、多国語ライティングシステム向けのパラメータセットを送ることにより、複雑な組版アルゴリズムなしにある程度の品質の表示が可能になる。

TRONコードの構造

●エスケープによる切り替え方式

エスケープシーケンスにより固定長コードの文字セットを切り替える構造をとる。言語指定コードが境界に挿入されることにより一つの文章中で複数の文字セットを混在して使用することができる。

文字セットには、1バイト文字コードと2バイト文字コードがある。

なお、以下に述べるコード仕様はアプリケーション独立に使われる、蓄積および伝送用のデータフォーマットとして使われるものであり、アプリケーション独自のデータフォーマットを規定するものではない。効率化を目指して、エディタのエディットバッファ中で文字を非エスケープ固定長にすることも可能である。

●4つの文字コードゾーン

文字コードには、A,B,C,Dの4つの文字コードゾーンが割当てられている。下図はこの概要を示したものである。

●3種類のエスケープ

・言語指定コード

言語指定コード(FE)は、続くコードが(21)～(7E),(80)～(FD)のとき、言語と文字属の切り替え指定として用いられる。また、複数バイトに拡張され、次のようになる。

(FE) [(FE)] .. (xx)

但し、(xx)は (21)～(7E),(80)～(FD)

[Z] .. は、空かZの連鎖

従って、2バイトで表現しうる言語指定数は220で、1バイト追加される毎に220づつ増える。

・TRON特殊コード

TRON特殊コード(FF)は、次に続くコードが(21)～(7E)の時、TRON特殊コードとなる。TACLなどに用いられ、文章中に挿入される特殊コードとして用いられる。

・TRONエスケープ

TRONエスケープ(FF)は、次に続くコードが(80)～(FE)の時、TRONエスケープとして使われる。TRONエスケープは、文章や図形のセグメント情報の区切りや、文書中に埋め込まれる指定付箋挿入等の区切り記号として用いられる。区切り記号の後に情報が続く。区切り記号としては通常2バイ

トであるが、多バイトに拡張される。

(FF)［(FE)］.. (xx)

但し、(xx)は (21)〜(7E),(80)〜(FD)

2バイトで表現しうるTRONエスケープ数は126で、1バイト追加される毎に220づつ増える。

TRON多国語処理環境の実現のために

TRONでは、東京大学文学部と共同で、多国語処理環境の具体的な実現プロジェクトを開始した。まず日本語でも重要な漢字から整備をスタートし、多国語に渡って進行させる。コード割当とともに、標準的なフォントを用意し、国際的な情報の流通に貢献する。

コード割り当て方針

現在主に漢字を使用する国は、中国、台湾、日本、韓国などである。これらの国には既存の国内規格が存在し、利用されている。このため、文字属層はこの各国別に用意し、国内規格と整合性がよいように定める。

次のような優先度で、漢字を収集し、CおよびDゾーンあるいはA, Bの空き領域にコード割当をおこなう。さらにあふれる部分については、別の面に割当をしていく。

・JISになく、国語辞典等にある文字
・以上になく、人名・地名で使われる文字
・その他流通している俗字等
・それ以外で大漢和辞典にある文字
・歴史的に使用されたそれ以外の文字

標準フォントを作って配る

上記、コード割当だけでも自動翻訳や音声合成は有用だが、原則的にはすべての文字が表示、印刷できるべきである。WWWなどで他国語のページを見た時に、フォントがなくて'□'が並ぶというのは、やはりコンピュータ側の不備である。

TRONではスクリプト層に登録されたすべての文字の文字データを持った標準フォントを1セット作りパブリックドメインとして提供することを計画している。どのフォントにも対応する文字がない場合であっても、フォントの流用機構により最終的に標準フォントが使用されれば画面上に'□'が表示されることはない。

台湾では国家が文字フォントを作成し、自由に使えるように配布している。情報を電子的に広く交換できるようにするためには、このような方策がぜひ

TRON文字コードは、

(21)(21)〜(7E)(7E) Aゾーン	8,836文字
(80)(21)〜(FD)(7E) Bゾーン	11,844文字
(21)(80)〜(7E)(FD) Cゾーン	11,844文字
(80)(80)〜(FD)(FD) Dゾーン	15,876文字

の合計48,400文字からなる。(21)(21)はブランクであり、与えられた一定の幅を持つ空白として扱われる。

TRON 2バイト文字コード

とも必要である。本プロジェクトでは、ここまで踏み込み、標準フォントを作成し、広く利用できるように配布することを行う。標準フォントがないとコード表が作れないという実際的問題もあり、標準フォントもコード割り当てと連携してまず日本語部分の整備から行う。

おわりに

ISO10646-1の問題点指摘であげた外字の問題や類似文字の過剰といった問題などは、大規模文字集合言語を使わないユーザにとっては、関係のない問題と映るかもしれない。しかし、すでに述べたように近年の世界規模通信環境の普及は状況を大きく変えている。WWWページの形で、従来縁のなかった多国語の文章が簡単に呼び出せるようになった。原則的に言うなら、ユーザがその言語を読める読めないにかかわらず、コンピュータとしてはそれらを当然正しく表示するべきである。

また、実用的な視点で言うならば、読めなければ読めないだけ自動翻訳の需要ができる。事実、日本ではインターネットの普及に伴い、WWWブラウザに組み込める自動翻訳プログラムが商品化され、新しい需要を開拓している。このようなことを考えれば、表示できるできないにかかわらず、多国語を言語を特定して送れるインフラストラクチャが必要なことは明らかである。

TRONの多国語処理は、システム側に多国語対応機能を取り込み、言語層、文字属層、スクリプト層、フォント層と階層を設けることにより、多国語対応に対してより理想的な環境を提供できる。これにより、世界的な文化・学術・産業にわたり多大な貢献をするものと確信する。

（坂村　健）

参考文献

[1] *K. Sakamura, "Multi-Language Character Sets Handlind in TAD," TRON Project 1987 (Proc. of the Third TRON Project Symposium), Springer Verlag, 1987, pp. 95-111.*

坂村 健, "TAD言語環境と多国語対応," TRONプロジェクト'87-'88, パーソナルメディア, 1992, pp. 133-150.

[2] *K. Sakamura, "After a Decade of TRON, What Comes Next," Proc. of the Eleventh TRON Project Symposium, IEEE Computer Society Press, 1994, pp. 2-16.*

[3] *N. Borenstein, N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part 2: Message Header Extensions for Non-ASCII Text," RFC 1522, Bellcore, Innosoft, September 1993.*

[4] *錦見 美貴子, 高橋 直人, 半田 剣一, 戸村 哲, "Mule: マルチリンガル文書処理系," 言語・音声理解と対話研究会資料 (SIG-SLUD-9501), 人工知能学会, 1995, pp. 49-56.*

[5] *坂村健 監修, "BTRON1プログラミングハンドブック," パーソナルメディア, 1992.*

[6] *Kenneth Katzner, "The Languages of the World," Routlege & Kegan Paul, 1986.*

[7] *Akira Nakanishi, "Writing Systems of the World," Charles E. Tuttle Company, Inc., 1980.*

[8] *諸橋 轍次, "大漢和辞典 修訂版," 大修館書店, 1986.*

[9] *江守 賢治, "解説 字体辞典," 三省堂, 1986.*

[10] *日本工業規格, "国際符号化文字集合 (UCS) — 第1部 体系及び基本多言語面," 日本規格協会, 1995.*

このような文化的事業に対して、どのようなかたちでさまざまなご協力をいただいたかは、あらためて報告書に記させていただきますが、本研究プロジェクトが正式に発足する前後からすでにお会いして、貴重なご意見や資料をいただいた方々に、あえてここではお名前を挙げませんが、この場をかりて厚くお礼を申し上げます。また、ご関心のある方々、ご協力いただける方々は、下記にご連絡くだされば幸いです。

1995年9月
研究代表者　田村　毅

平成7年度日本学術振興会産学共同研究支援事業
「人文系多国語テクスト・プロセシング・システムの構築に関する研究」

研究代表者	田村　毅	東京大学大学院人文社会系研究科・教授（フランス文学）
企画委員	青柳　正規	東京大学文学部・教授（ギリシャ・ローマ考古学）
	片山　英男	東京大学大学院人文社会系研究科・教授（西洋古典学）
	坂村　健	東京大学大学院理学系研究科・助教授（情報科学）
日本語漢字監修	山口　明穂	東京大学大学院人文社会系研究科・教授（国語学）

連絡先　東京大学文学部仏文研究室　田村　毅
〒113 文京区本郷7-3-1
電話　代表 03-3812-2111（内線3482）
FAX　03-5800-5916

東京大学・人文系多国語テクスト・プロセシング・システムの構築に関する研究グループ