人文科学とコンピュータ　39－9
（1998．8．7）

Unicodeの動向と利用

織田 哲治　　榎本 義彦　　小田 明
日本アイ・ビー・エム株式会社

数年前にソフトウェアメーカーを中心とするグループによって開発された国際統一符号化文字集合Unicodeは ISO 10646 としても規格化され、その実装が進んでいる。それにともなって、一般利用者の間でも漢字にかかわる問題と関連してUnicodeが関心を集めるようになってきた。本稿では、Unicodeの概略、及びその動向を説明した後、文字コードにおける文字の定義の観点から、Unicodeの利用に際しての問題について考察する。

Current Status and Usage of Unicode

Tetsuji Orita　　Yoshihiko Enomoto　　Akira Oda
IBM Japan, Ltd.

Unicode developed by mainly computer vendors and also defined as ISO 10646 is now in the implementation phase. Users are concerning about Unicode related to some discussion on Kanji. This paper clarifies these discussion from viewpoint of character definition in coded character set, and consider the processing to utilize Unicode.

1．はじめに

インターネットによる情報交換が頻繁に行われようになるに従い、文字が消える，あるいは文字が化けるといった情報交換における基本的な問題点が指摘されるようになってきた。インターネットのように、異なるシステム間で情報交換が行われる場合、コンピュータ上での情報の表現に使われている文字コードが送信者と受信者の側で異なっている可能性が高く、その場合、両者の間で文字コードの変換が行われるが、当然ながら、両者の文字集合に共通に含まれていない文字についてはその情報交換は保証されない。一般的には、システム固有の文字，例えばユーザ外字やベンダー外字，はその交換が保証されない。また、両者に共通に含まれる文字でも、各システムで実装しているフォントの違いのために結果として送信者が意図した字形で受信者に届けられない場合がある。さらに、それ以前の問題として、そもそも必要な文字が共通の文字集合に含まれていないといった問題もある。

このように、文字集合／文字コードに関わる問題が指摘される中で、話題として必ず取り上げられるのがUnicodeである。Unicodeは、従来の文字コードよりも大きな文字集合を含んでいる点や、Windows NTなどでも実装されている点、また Unification(漢字統合)などその特徴が話題となって、日本でも注目されてきた。本稿では、まず、2章～4章においてUnicode及び他の符号化文字集合についてその概略、及び動向を説明し、5章では Unicode の利用にあたって、文字コードにおける文字の定義とその処理について考察する。

2．Unicodeとは

Unicode[1]は、2バイト(16ビット)で世界中の言語で使われている文字をできるだけ多く含めた符号化文字集合である。本来、ソフトウェアの各国語実装をより効率よく行うことを目的として米国のソフトウェアメーカーが中心となってコンソーシアムレベルで設計した Unicode は、ISO(国際標準化機構)の活動と相まって国際規格10646-1 規格[2]として発行されている。日本で

は、対応するJIS規格としてJIS X 0221「国際符号化文字集合(UCS) - 第1部 体系及び基本多言語面」が発行されている。JIS X 0221では、ISO 10646の日本語翻訳に附属書として日本文字部分レパートリを規定し、さらに、既存のJISの符号化文字集合規格JIS X 0201「7ビット及び8ビットの情報交換用符号化文字集合」、JIS X 0208「7ビット及び8ビットの2バイト情報交換用符号化漢字集合」、及びJIS X 0212「情報交換用漢字符合－補助漢字」内の文字との対応を参考としてあげている。

図1に示すように、ISO 10646-1の符号空間は、4バイト(32ビット)の正規形式と2バイト(16ビット)のBMP(Basic Multilingual Plane)形式として表現され、それぞれをUCS-4(Universal Character Set four-octet), UCS-2(Universal Character Set two-octet)と呼ぶ。4バイトの符号化文字集合は、4次元空間のイメージとしてとらえられる。つまり、256(row) x 256(column)の2次元の面(plane)が256面集まった群(group)、その群が128個集まって構成される。そのうちの最初の面を2バイトのBMP形式として表現した仕様がUnicodeとなる。現在は、BMPのみに文字の割り当てが行われている。

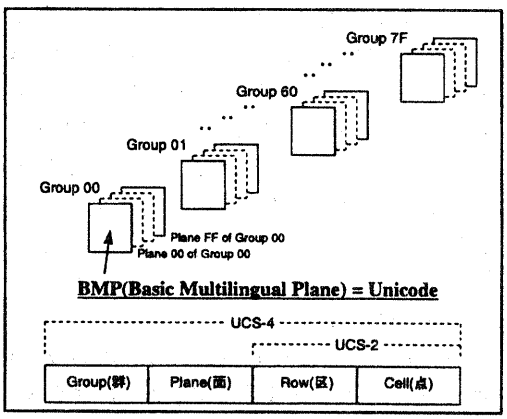

図1. ISO10646-1の符号空間

3．Unicodeの文字集合

Unicode（最新の版はversion 2.0）には、約39,000文字が含まれている。Unicodeの文字集合は、それまでの文字コードの国際規格及び各国の規格を含むように、また、それら以外で実装されている文字の中からも定義されている。大雑把に分けると、漢字が21,000文字、韓国のハングル文字が11,000文字、ラテン系の文字を含め他の非漢字,シンボルなどが7,000文字弱となっている。日本の文字コード規格JIS X 0201, JIS X 0208及びJIS X 0212内の文字は全てUnicodeに含まれている。

Unicodeの漢字は、CJK（中国, 日本及び韓国）統合漢字といわれ、各国の文字規格で定義されている文字集合から構成されている。CJK統合漢字は、図2のような考え方に従って定義されている。

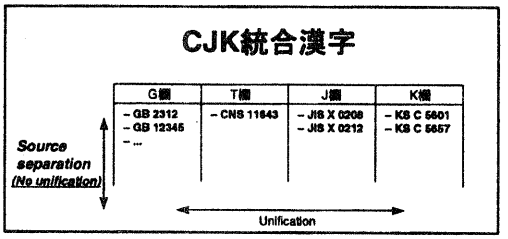

図2. CJK統合漢字

上図において、各欄内の文字には、重複なく符号位置が割り当てられるが、欄をまたがった規格間では、統合規則(unification rule)を用いて同じ符号位置を割り当てる。この統合規則というのは、規格中の文字の字形の違いが実字形(actual shape)、つまり単なる表示字形上のデザインの違いであるか、あるいは抽象字形(abstract shape)、つまりデザインをこえた字形の違いであるかを考慮し、実字形の違いと判断される場合のみ統合し、同じ符号位置を割り当てるというものである。但し、先に述べたように、既存の規格との整合性を維持するために、一つの規格内で既に別区点位置に割り当てられている漢字はたとえそれが統合規則により統合されるべき漢字であっても、Unicode上では別々の符号位置の割り当て(source code separation)を行っている。図3に、抽象字形，実字形の違いと、source code

separation の適用例を示す。ISO 10646-1 Annex S (Amendment 8)には、実字形の違い、抽象字形の違いの具体例と統合規則のより詳細が示されている。なお、ISO 10646-1 規格は、図２と同じ形式で各欄にそれぞれの規格にある文字字形を載せているが、Unicode では、一つの字形のみ載せている。

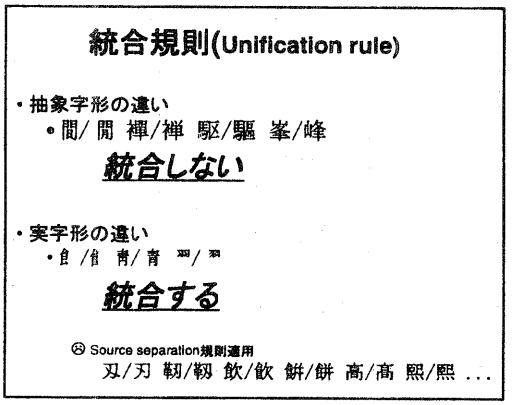

図3. 統合規則

他に、文字集合としての Unicode の特徴としてあげられるのが、結合文字である。他の符号化文字集合と同じように、Unicode は、その符号化空間の一つの点に一つの文字を割り当てており、それが文字データ処理の単位となるが、Unicode の文字集合の中に結合文字として分類される文字が含まれている。この結合文字(combing character)には、欧米の文字で用いられるアクセントなどのダイアクリティカルマークや、タイ語やラオス語など東南アジアの文字が含まれる。図４に示すように、一つ以上の結合文字が、基底文字(non-combing character)と結合されて合成列となる。

合成列は、Unicode の文字集合の一部ではないが、既に Unicode 上に含まれている他の基底文字("pre-composed"形式)と同じ形で表現される場合や、複数の結合文字が異なる順序で結合された場合に二つの合成列が同じ形に表現される場合がある。

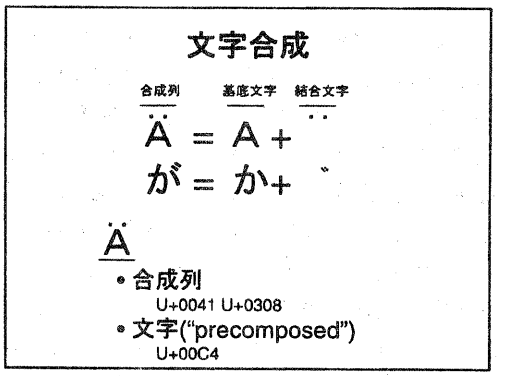

図4. 文字合成

現在も、ISO SC2/WG2 によって、ISO 10646-1/Unicode への追加文字の検討が行われている。これには、少数民族の言語、古典・古語などの文字の他に、約 6,000 文字の漢字の追加提案が含まれている。この中には、約 600 文字の日本提案の漢字が含まれている。これにより、ISO 10646-1 BMP(plane 0) / Unicode の 65,536 文字分の区点位置の中で、大きな連続区域の文字の割り当てはほぼ終わろうとしており、今後は、plane 0 以降、すなわち、plane 1 に非漢字、plane 2 に漢字をそれぞれ追加するという枠組みのもと、追加文字の検討が進められることになっている。

４．Unicode 以外の文字集合規格の動向

Unicode 以外の文字集合規格としては、日本では、JIS X 0208 規格が代表的で、ここには 6,879 文字が定義されている。JCS (Japanese Coded Character Set) 調査研究委員会は、この JIS X 0208 改訂の原案作成を行い 1997 年に発行しているが、このときには、文字の種類を増やさない方針で、この符号化文字集合にこれまで定義した漢字の典拠を再調査し、漢字に対する包摂(unification)の考えを明示的に導入し包摂基準を設けている。JIS X0208:1997 規格の包摂基準は、Unicode の統合規則とは別のものである。なお、JIS X 0208 が同規格の以前の版との互換性のために包摂している漢字は、Unicode では統合され

ていない。

このJCS 調査研究委員会は、引き続き JIS X 0208 とともに用いる拡張文字集合の調査検討に入っている。この拡張文字集合(新 JIS 漢字コード)[3]は，JIS X 0208 の文字集合を補い，JIS X 0208 が当初符号化を意図していた，現代日本語を符号化するために十分な文字集合を提供することを目的としており、約 5,000 字の漢字・記号類が新たに追加される予定になっている。俗に、第3水準と呼ばれることのある、JIS X 0212 補助漢字とは独立に定められるので、新しい拡張文字集合に、JIS X 0212 に含まれている文字が入ることもあれば、入らないこともある。これは前述の基準で JIS X 0208 に対して文字集合の拡張を考えるからである。なお、この拡張漢字集合のうち ISO 10646-1 に含まれていない文字は、新規格制定後、追加提案されると思われる。

一方、他の漢字圏，中国や韓国では Unicode/ISO 10646-1 が今後の中心となろうとしており、ISO 10646-1 に対応する規格として、中国では GB 13000、韓国では KS C 5700 を制定している。台湾でも、対応する規格の制定作業を行っている。

Unicode の CJK 統合漢字には、中国で主に使われている簡体字、及び台湾で主に使われている繁体字にそれぞれ別々の符号位置が割り当てられているため、この両国間で繁体字、簡体字を同時に扱いたいという要請を Unicode の文字集合では満たしている。また、韓国では、漢字にではなく、ハングル文字に対する要求が高く、従来の文字コードでは 2,350 文字、また以前の Unicode (Unicode 1.1 時点) では 6,656 文字のハングル文字しか扱えなかったのに対して、最新の Unicode 2.0 の文字集合では、すべてのハングル 11,172 文字が定義された。

これらの国々では、Unicode を推進する傍ら、既存の PC(パーソナルコンピュータ) の文字コードを拡張して、Unicode の文字集合を含める動きがある。図5に示すように、中国では、GB 2312 の標準に基づく PC コードを拡張して、Unicode の CJK 統合漢字部分を全部含めたものを実装し始めている。台湾，韓国でも、一部で同様の動きがあるが、全体の方針としては Unicode を推進しており、これらは、あくまでも Unicode への移行を目的としたものと考えられる。

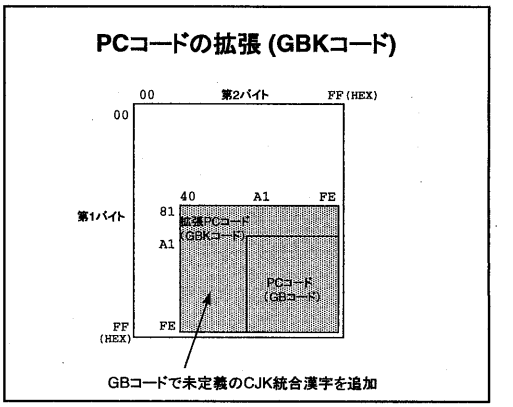

図5. 従来 PC コードの拡張

5. Unicode の利用にあたって

Unicode の利用を考えるにあたって、筆者らは[4]に「利用形態から見た実装」について述べた。本章では、Unicode 利用のために最も本質的である文字コードにおける「文字」の問題を情報処理の立場から考察する。ここでは、文字とは何かを言語学や文化論の観点から議論するのではなく、「文字」というものが情報処理システムのなかでどのように定義され（あるいは、定義でき）それをどのように利用、処理していくのかを考えていく。これは、Unicode の設計意図を理解した上で処理システムを構築し、利用して行くためである。

ところで、Unicode に対しては、いろいろな批判がされてきている。やはり[4]のなかで、それらの批判をおおよそ、①規格制定時における経緯に対する批判、②文字合成シーケンスに関するもの、③漢字の統合(Unification)および多国語処理に関するもの、④文字の不足を訴えるもの、という観点に分類し、②から④の批判に対して、情報処理の側面から解説した。文献[5]も批判を分類した上で意見をのべている。[5]では、従来の国内規格と

の互換性をあげている点と文字合成シーケンスと多国語処理をあげていない点が少し異なるが、[5]が漢字処理を話題にしたセミナーであったためであろうと思われる。以下では、文字の不足および漢字の統合に関する問題を情報処理の観点から考察していく。

5．1 文字コードと文字の定義

一旦 Unicode をはなれて、まず文字コード規格における文字の定義とは何かを考えるためにIBM の文字コード標準についてみてみる。

IBM は自社製品の実装の一貫性をはかるために、各国文字、各種符号系のための社内文字コード標準を制定している。これらは、ISO の国際規格や各国の国家規格に基づいたもの、PC 系、UNIX 系、IBM ホスト用の EBCDIC 系などの文字コードである。ところで、この文字コード標準は文字の定義から始まり、各文字がどういう形でどういう意味をもつのかという定義づけがもとになる。この各文字に GCGID (Graphic Character Global Identifier)という 8文字の英数字からなる便宜的な ID をふっている。文字集合はこれらの ID の集合で定義される。また、符号系の枠組みに沿ってこの ID それぞれに符号位置を割り振ったコードページを定義することによって文字コード標準が定義されている。この方法では、一つの文字集合を定義し、それにたいして PC 系、UNIX 系、EBCDIC 系といった異なる符号系の文字コードを定義できる。

この文字コードの定義方法で注目すべき点は、文字に ID をふる過程に文字の定義に関する問題がすべて押し込まれ、以後はこの ID にもとづいて情報処理システムを構築している点である。ある意味で、自然言語で使われる文字にこの段階で何らかの定義づけをしているのである。漢字に関してはラテン系の表音文字に比べると、ここでの定義が難しく、あいまいなものになり易い。（例えば、「真」と「眞」、「高」と「髙」、「鴎」と「鷗」などをどう扱うかである。）後述するが、JIS などの日本語の文字コード規格をめぐる議論の一つもこの点にある。Unicode では、漢字に“CJK UNIFIED IDEOGRAPH-“ の後に 16 進表記の２バイト符号をつけて名前が付けられているが、文字の定義は CJK 各欄に示された原規格の規定によるとしている。

この文字コードの定義方法でもう一つ注目すべき点は、文字集合の定義と符号化を明確に段階わけしている点である。文字集合は、本来情報処理の対象として扱いたいものの集まりであり、符号化はそれを情報処理システム上でどう表現するかである。

5．2 文字の不足

文字の不足に関しては、Unicode への批判というより、日本語の文字規格自体の問題としてとりあげられることも多くなっている。３章に述べたように、Unicode は JIS X 0208 および X 0212 を含むように定義されている。したがって、日本語の文字に関してだけ言うなら、不足の要求は Unicode のみならず JIS X 0208 とその補助漢字としての JIS X 0212 にも共通することである。４章で述べたように、JIS では新しい拡張文字集合の検討が進められており、これはいずれ Unicode にも反映されると思われる。

ところで、文字の不足が問題にされるとき、ここでの要求の一つは、漢字の表示・印字字形の観点からくるものである。確かに、現在提供されているパソコン、ワープロ、一般の情報処理システムで文芸書の記述や人名などを適切に表示しようとすると文書の作者の思うような文字の表示・印字字形であらわせないことがある。たとえば、「高橋」ではなく「髙橋」と表示したい場合である。このような問題に対処する方法の一つは、表示・印字字形ごとに、それぞれを１文字として符号を割り振る方法である。

通常、利用者は、文字コード表に載っている字形をみて、あるいは単に自分の使っているシステムの表示・印字字形をみて、それが処理系の文字の定義のように思いがちであるが果たしてそうであろうか。先に述べた、1997 年の JIS X 0208

の改訂作業では、包摂基準の明確化が行われている。この包摂基準では例えば、「高」と「髙」、「間」と「閒」を包摂する、「真」と「眞」は別の文字符号をわりあてるといったことが決められている。過去の規格との整合性をとるために例外的に設けられた基準があるために多少混乱するが、この包摂基準の考え方をつきつめていくと、文字に符号を割り振る単位は、文字の図形表現のわずかな差異を包摂したものにたいしてであり、図形表現の差異は必要であればフォント情報などの文字コード以外で指定する（したがって、規格でいえば、これに関する指定方法は文字コードの規格ではなく、例えば、電子文書の規格になる）ことになる。

漢字には義、音、形、すなわち、文字が運ぶ意味、発声されたときの音、そして文字の表示図形（字の形）があるといわれる。したがって荒っぽい言い方ではあるが、文字定義の情報処理のモデルの両極端としては、形に着目して過去の経緯により発生した形の違いごとに文字コードの最小単位を設定する方法と、こういった形の差異はひとくくりにして文字としての意味単位で文字コードの最小単位を設定する方法が考えられる。このそれぞれの文字コードを採用すると、アプリケーションの処理上どういったことが必要であるかは次節でもう少し検討するとして、Unicodeあるいは JIS 規格に文字が足らないと言ったときに、数百字、数千字、数万字と足らない文字数の意見に隔たりがあるのが、どうもこの文字コードの最小単位の設定に差があるためと思われる。

5．3 文字コードの定義と処理

まず、文字コードを表示・印字の図形表現の単位で定義していくモデルではどうなるか。フラットテキストのように、特にフォント情報を持たないものでも、符号の違いによって「高」と「髙」を区別して扱うことができる。ところで、このようなデータにたいして、「高橋」と「髙橋」を区別せずに検索しようとすると、それぞれの符号は異なるために、これらの OR 条件で検索にいくか、処理システムで異体字辞書のようなものを持っている必要がある。

一方、文字コードを包摂単位に定義したモデルでは、「高」と「髙」の区別は符号からはつかないために、フラットテキストではこの差異を表現できない。これらを区別するためには、文字コードとは別の枠組みで、異体字番号またはフォントインデックスといったものを導入する必要がある。このモデルでは、「高橋」と「髙橋」は文字コードからは区別されずに検索される。逆に、この両者を区別して検索するためには、異体字番号またはフォントインデックスといった情報もみる必要がある。

ここでは、文字コードの定義が処理にどういう影響を与えるのかを考察するために、一般的に包摂と表示・印字字形の両極端のモデルを示したが、包摂の考え方を明示しているのは JIS X 0208 である。Unicode では原規格として JIS X 0208 が参照されているが、Unicode が明示しているのは CJK 欄に関する統合規則である。

5．4 漢字の統合にたいして

Unicode にたいする批判のなかで、文字の定義のモデルに関連するものとしては、CJK（中国・台湾・日本・韓国）の文字統合がある。起源は同じ文字であったとしても、これらの国の間では文化・言語・歴史的な経緯の違いが存在するため、その意味が同じということで統合されているのではない。統合規則は、前述のように、実図形に基づくとしているが、よく例に出される「骨」や「刃」のように違いのよくみえるものや、そうでないまでもそれなりに全体のフォントの印象は異なる。したがって、Unicode では CJK の統合を行ったために、文字コードからその文字がどの言語かわからない、またこのために多言語処理ができないといった批判になる。これは、フラットテキストのなかで ISO2022 系の文字コードが使われている場合のように、各国の文字コードの切り替えと改行のような何種類かの制御コードが使用されるような処理モデルを想定しての批判

であると思われる。Unicode を用いてこのような処理を行うためには単なるフラットテキストではなく、文字コードの外の枠組みで、フォント情報を指定する構造やタグをもったものでなければならない。

5．5 おわりに

本章の目的は、Unicode に対する批判や問題の議論に対して、反論することではない。これらの議論のなかで、感情的な要素のあるものや誤解からくるものを除くと、それぞれ想定された処理モデルがあり、それに基づくと、文字が足りない、文字コードがこのようである方が最適であるといっているように思えるからである。むしろ、本章では、Unicode に対する批判をもとに文字の定義のモデルからその処理について考察した。

Unicode の利用においては、ソフトウェアの国際化に利点をみることができる。Unicode の当初の設計意図のなかには、2バイト固定長で文字を扱うことで文字ストリング処理などを簡単にするといったねらいがある。これに関しては、文字合成の話やさらに多くの文字を2バイト体系で表現するための UTF-16[4]といった考え方が固定長での文字処理の利点に混乱を与えていることも事実である。しかし、これまで米国などで1バイトの文字コードを基本に特別の注意を払うことなしに開発されたソフトウェアを、日本語環境やその他の各国語環境に移植する際に多大な労力を払ってきたことを考えると、Unicode という共通の文字コードを用いてソフトウェアの開発を行うことによって、こういった手直しが最小化され、結果的に利用できるソフトウェアが増える、あるいはコストが下がるといった効果があると思われる。Unicode が世界中の文字を集めていることから、一般の利用者にとってもいろいろな文字が使える利点がある。ただ、Unicode を利用するにあたって、本章で述べたように、今までの各国ごとの文字コードを使っていた処理系の文字コード部分だけを Unicode に置き換えるだけでは済まないものもあり、機能によっては文字コードの枠組みの外の処理を含めて考えていく必要がある。

[参考文献]

[1] The Unicode Consortium, The Unicode Standard Version 2.0, Addison-Wesley, July 1996.

[2] ISO/IEC 10646-1:1993, Information technology-Universal Multiple-Octet Coded Character Set(UCS)-Part 1: Architecture and Basic Multilingual Plane, 1993.

[3] 符号化文字集合調査研究委員会 第2作業分科会：JIS 漢字策定の進捗状況 (http://jcs.aa.tufs.ac.jp/jcs/new-jis/),June 1998.

[4] 織田哲治，木戸彰夫，小田明：Unicode の最新動向，情報処理学会デジタル・ドキュメント研究会資料 12-3, pp17-24, 1998.

[5] 伊藤 英俊：XKP 最新動向セミナー資料、Windows NT 漢字処理技術協議会, 1998.