5 a　廣輯詞隱先生南九宮十三調詞譜26卷卽南詞新譜
明・沈璟（詞隱先生。1553－1610）撰　明・沈自晉（鞠通生。1583－1665）重定
北京、北京市中国書店、1985　影印　2冊
原本：清順治十二年乙未（1655）呉江沈氏不殊堂刻本

5 b　廣輯詞隱先生南九宮十三調詞譜26卷卽南詞新譜
明・沈璟（詞隱先生。1553－1610）撰　明・沈自晉（鞠通生。1583－1665）重定
1985年北京市中国書店用順治十二年乙未（1655）呉江沈氏不殊堂刻本影印　2冊

東アジアの書誌データベースとISO 10646 UCS

宮澤　彰（学術情報センター）
Miyazawa Akira

1．10646 UCS

1993年5月に新しい国際標準文字コードであるISO-IEC 10646が出版された。Universal Multiple-Octet Coded Character Set、略名をUCSという。この文字コードは約3万の文字を含んだ巨大な文字表で、754ページにも及んでいる。この文字コードはこれまでの、JISコードやGBコードのような各国語用のものではなく、国際的にこの文字コードだけで（一般的な）すべての言語が表記できることをめざしているものである。当然、書誌データベースの国際流通、あるいは国際的なネットワークに及ぼす影響は大きいものとなることが予想される。ここでは、この文字コードについて若干の紹介を行い、東アジア書誌データベースの国際流通におよぼす影響を考察する。

1．1　これまでの文字コード

たとえば日本ではほとんどのコンピュータがいわゆるJISコード、JIS X 0201およびJIS X 0208を使用している。一方、たとえば中国ではJISのかわりにGBコードが、また韓国ではKSコードなどが使われてきたわけで、どの国も国内的な交換と処理に関する限り、各国文字コードにより（それぞれ、外字や、非標準の文字コードといった問題をかかえながら）コンピュータ処理を発達させてきた。

ところが、この状況はいったん国際的な交換をしようと思うと大きな問題となってしまう。KSコードで記録された韓国・朝鮮語の書誌データや、文書を日本で受け取って処理しようと思っても、JISコードの中にはハングルが存在しない。中国語をGBコードで記録した場合、よほど運が良ければJISコードにある漢字だけで表せるかもしれないが、一般には変換できない漢字が多すぎてそのまま処理することは難しい。もちろん、欧米のコンピュータでこういった東アジアの言語データを処理することは、特殊な環境を用意しなければできなかった。

1．2　UCS

この問題を技術的に克服するためには２つのアプローチがあり得る。１つは、各国コードをすべて取り扱えるようなシステムを用意することである。もう一方はJISやGB、KSなどの各国コードをすべて含んだ１つの文字コードを作って、各国のコンピュータがこの新しい文字コードを扱えるようにする方式である。

ISOの2022という番号の標準は前者の方式で、複数の文字コードを切り替えて使えるような標準である。しかし、実際のコンピュータシステムでこの方式に基づいてすべての文字コードを扱えるようなものはほとんど実現しなかった。もともと、そんなに多くの、しかも大きな文字コードを扱うことを想定していなかったため、きわめて複雑な拡張をせざるを得なかったためであろう。

そこで、すべての文字を含んだ１つの文字コードというアプローチがとられることになった。結果としてUCSがようやく出版されたわけであるが、そのスタートからみれば８年ほどの時間を要した。情報処理の世界の標準としては異例なこの長い期間を要したのは、ひとつには、米国が途中からUNICODEコンソーシアムの方式への切り替えを提案してきたなど、標準の世界での手続きとコンピュータビジネスの主導権争いの場となってしまったことがあり、また、一方、３万字という大きな文字集合を選定、コード化するという作業そのものが大変な仕事量であったためもある。（辞書の編纂を考えれば、むしろ、短い時間といえるかもしれないが）。

いずれにしても結果としてできあがったUCSは、米国のUNICODEの提案を基本として、65536字まで（最大で約40億字まで）はいるコードスペースに約３万字を配したものである。ラテンアルファベット、その拡張（ウムラウトやアクセントのついた文字）はもちろん、ギリシャ、キリル文字をはじめ、アルメニア、ヘブライ、アラビア、デバナガリ、タミルなどインドの諸文字種、タイ、ラオ、グルジア、そして、ハングル、漢字を含んだ巨大なものである。さらに、もちろん、これですべての文字種というわけではないので、今後とも拡張されていくことが予定されている。

1．3　漢字

約３万のUCSの文字のうち20902字が漢字CJK Ideographsである。もちろん最大の部分であるが、ページ数でも６割以上を占める。ほかの部分と変わっているのは、１文字につき４カラムを使って、中国、台湾、日本、韓国の字形を並列表示している点である。この表示方法は、UCSの作成時漢字部分の最大の論争であった「統合化」Unificationの結果を示しているものである。

UCSの作成時、漢字に関する当初のアイディアは、JIS、GB、KSの基本集合のそれぞれ漢字部分を、UCSの別々の場所に独立に割り当てようというものであった。この方法では各国文字コードからUCSへの変換は（漢字であれば）計算だけですみこれまでのデータからの移行性がよい。しかし、各国に共通な文字は３カ所に別々に現れることになり、コードスペースの使用効率は当然悪くなる。（全体として収容できる文字数が減る）。そのうえ、80年代の後半から、各国とも従来の基本集合に加え補助集合の類の文字コード標準を成立させてきた。このため、各国に共通に持つ字の数が増え、別々に扱うと無駄が多いという状況になってきた。そこで、「同じ文字」は１つに統合して配列し直す「統合化」案が有力となってきた。

この問題は、主として東アジアの各国間と米国とで話し合われ、結局 CJK-JRG (China、Japan、Korea — Joint Research Group) という団体が組織され、中国、日本、韓国、米国などが参加して、統合化の作業を行うことになった。この作業が行われたのは1991年の７月から、1992年の３月にかけてである。この会議の中で統合化のためのルールや配列方法の議論を行い、それまでに決まっていた各国の標準文字コードのほとんどを統合化して２万字あまりのレパートリを決定した。

統合化の際もっとも問題となるのは字体の差である。CJK-JRG では日本の提案に基づき文字の形による統合化のルールを定め、小さな字体の差は統合化することにした。何が統合化される小さな差か、については、統合化するもの、しないものを定めた表を用意して、この表に基づくというルールとしている。この結果、統合化されて１つの文字コードが割り当てられているが、小さな字体の差があるという場合が数多くみられる。これが無視されることのないよう、元の字体の差のわかる、４カラム並列表示を採用したわけである。

１．４　実装

94年夏現在、UCS が実装されたコンピュータはまだ普及していない。文字コードが変わるということはコンピュータシステムにとって非常に大きな変更となるためである。本格的にはオペレーティングシステムの中核部まで及ぶ変更であり、既存のアプリケーションソフトウェアも新しい版を必要とする。こういった点が UCS 製品の登場を遅らせている原因であろう。

しかし、UCS を扱えるようなオペレーティングシステムは登場しつつある。これらの製品が容易に手にはいるようになれば UCS を使えるようなアプリケーションソフトウェアの開発も容易になってくるであろう。

２　書誌情報データベースと言語環境

２．１　これまでの状況

日本、中国、韓国など各国はそれぞれの全国書誌を MARC フォーマットでデータベース化してきた。これらのデータベースは各国内では流通、利用されてきている。しかし、国境を越えた利用という点ではきわめて限定的であった。

原因はいくつかあろうが、最大の問題は各国別の文字コード環境では他文字コードのデータが扱えないという点にあった。受け取り側で処理を行うために特殊な環境を必要とするようでは、データ交換は発達しないものである。

２．２　UCS の利用

実際に交換が意味あるものとなるためには、受け取った側で特殊なシステムでなく処理可能なものとなる必要がある。このためにもっとも望ましいのは各国のシステムが UCS をベースとし、フォントさえそろえれば他言語のデータも表示可能なものになることである。UCS を単に交換のためのコードとするだけでなく、各国のシステムでの処理用のコードとしても共通化を図っていく必要がある。

２．３　ローカライゼーション

各国の処理システムが、同じ UCS をベースとすべきであるとは言っても、それは同じシステム環境を揃えなければならないと言うことではない。たとえば、文字の入力方法は言語依存であ

る。かな漢字変換は日本語では有力な入力方法であるが、中国語では一般的には使用できない。書誌データベースで必要とされる索引語の切り出しアルゴリズムなども言語依存である。一般的な、アプリケーションシステムの場合でも、メッセージやメニューは言語依存である。

オペレーティングシステム、アプリケーションシステムの双方で、こういった言語環境に依存する部分としない部分の切り分けを行っていくことが、各国環境へのローカライゼーションのために必要である。

2．4　書誌データフォーマットと目録法

UCS による文字コードの共通化と、それを扱える各国語環境が利用可能になることは、書誌データベース交換と相互利用のための大きな一歩であるが、最終的な解決ではない。次の段階としてはデータフォーマットの問題があり、さらに書誌データベースの作り方としての目録法の問題がある。

フォーマットの問題として、一般に UNIMARC と USMARC の問題が取り上げらる。しかし、タグ番号が200か245かといったレベルのことは、実際上ほとんど問題ではない。

メインエントリーシステムを使用しているかどうかというような点はやや問題である。この場合、メインエントリーを採用している方からしていない方への変換には問題がないが、逆は機械的には変換できない。

さらに厄介なのはレコード単位の問題である。日本で「セットもの」とか、中国で「叢書」と呼ばれる類の出版形態では、レコードの単位を集合的にとるか、各巻単位とするか、分析的にするかという書誌単位問題がある。こういった点は明確にしておかなければ、データ変換そのものが意味がなくなってくる。

言語レベルに関しては、たとえば日本語の読み、中国語でのピンイン等を記述するか？記述に含めるとすればどのような形にするかといった問題がある。また、索引のための語分割を記述するか？といった問題もある。言語レベルの問題は言語処理機能の発達によって状況が変わることもあり（たとえば語の自動分割など）、何をどのように記述すべきか慎重に検討する必要がある。

2．5　おわりに

UCS は他文字種、他言語の書誌データベースを構築、交換、処理する基礎として、現在最良の解決を与える文字コードである。

その交換、相互処理が実現するために、書誌データの作成、交換、処理に関わる人々は UCS が使用可能なシステムを育てていく必要がある。

また、UCS という文字コードのレベルのみでなく、各言語の言語レベルでの処理機能を発展させ、各言語での環境を確立できるようなローカライゼーションも必要である。

さらに、書誌データのフォーマット、目録法での相違点を明らかにし、共通化あるいは変換方法の確立という作業が今後必要となるであろう。

上記のような標準化ないしは変換仕様の確立という作業にあたっては、各言語、出版、目録法等の伝統において同じ部分と異なる部分を冷静に分析し、何を共通化すべきかを洗い出していくということが必要である。