香川大学経済論叢
第71巻 第3号 1998年12月 151-164

日本語・満州語の辞書作成のための補助システム（II）

本田道夫

I はじめに

約3年ほど前に，満州文字を扱える入出力システムとテキスト編集のためのエディタを開発した。[2] それまでブルガリア語・日本語の辞書編纂システムを目標としながらも，できるだけ一般的に利用できる辞書編纂システムを目標に研究を進めてきた。そして，日本語・満州語辞書についても，言語固有の部分としての入出力システムを開発すれば，編集システムや，辞書システムは共通のものとして開発して対処できるとして対象に含めてきた。

そのような状況で，日本語・ブルガリア語辞書と日本語・満州語辞書の2つの辞書の作成を目標に，システムを検討・試作・試用してきた。日本語・ブルガリア語辞書の場合には，通常の辞書のような構成のものであり，特徴としては各単語に対して，例文を豊富に入れたものとし，そのための補助システムとしては，つぎのような機能や使い勝手のものをイメージしていた。

- 例文の集まりを管理し，対象とする単語に対してそれが使用されている例文を，活用形なども考慮に入れて的確に検索・表示でき，その適切な部分を指示して，辞書内容に取り込む機能を備えている。
- 各単語に関しての記述部分については，記述項目のためのフィールドをあまり細かく設けずに，入力位置の移動などが簡単に行えるような，かなり自由な形式で単語の訳，解説，例文などが記述できる。

一方，日本語・満州語の辞書の場合，最初のものとしては，日本語と満州語の双方向の単語と単語の対応だけを与えるものをということであった。つまり，

満州語の単語1つに対して複数の日本語の単語だけを対応させるようなものと，その逆のものということであり，日本語・ブルガリア語辞書と日本語・満州語辞書では，作成しようとしている辞書そのものの考え方が大きく異なるものであった。

2つの辞書作成およびそのための補助システムの開発を考えていた時点では，補助システムは，辞書をどのようなものとするかが決まれば，対象とする言語の違いはあまり大きなシステム上の違いとはならないと考えていた。しかし，逆に辞書そのものをどのようなものとするかが大きく異なれば，システムそのものも大きく異なるものとならざるを得ないものとなり，今回は，日本語・ブルガリア語辞書と日本語・満州語辞書のための補助システムとしては，別に考えざるを得なくなった。

辞書作成用の補助システムとしては，日本語・満州語辞書として考えている，単語と単語の対応を与えるものの方が簡単であることから，経験を積むことも考えて，まずそちらの辞書作成を目的として一応のシステムを考えることにした。そのような観点から，約3年前に作成した満州文字用の入出力システムと編集システムを用いることからはじめながら，実際に日本語・満州語の辞書作成を行い，その過程で試行錯誤的に改良・利用を繰り返してシステム設計の変更も何度か行いながら，現時点で一応の形が整ってきた。

システムは，(1)ローマ字入力からの変換機能を備えた満州文字の入出力機能とJIS漢字以外の漢字についての入出力機能を備えた，デバイスドライバとして組み込むサブシステム，(2)データの入力ができ，また入力後にローマ字部分や日本語部分の辞書式順序での確認などのチェック・修正もできるデータベース機能を有した辞書管理のサブシステム，および(3)辞書作成のためのデータ入力・編集用のエディタサブシステム，からなる。以下この3つのサブシステムについて述べる。

II 満州文字入出力サブシステム

1. 満州文字列の入力

最初に設計したシステム[2]での満州文字入力は，文字種が多いため，通常の英字・数字・記号キーとシフトキー，NFER キーを組み合わせて用いることでキーボードのキーに満州文字を対応させ，直接入力するものであった。つまり，単に通常のキーを押す，シフトキーと通常のキーを押す，NFER キーと通常のキーを押す，シフトキーと NFER キーと通常のキーを押す，の4通りの入力方法を用いるものであった。

当初から分かっていたことではあるが，実際に満州語文字入力の作業に取りかかると，文字種が多く，キーと文字の対応が簡単には覚えきれないために，その作業は非常に大変なものとなった。そこで，単語ごとに，ローマ字表示で入力し，それを満州文字に変換する方法を再度検討した。しかし，以前からの問題である単語内での区分けについて十分な情報が得られないため，変換のためのアルゴリズムが確立できなかった。

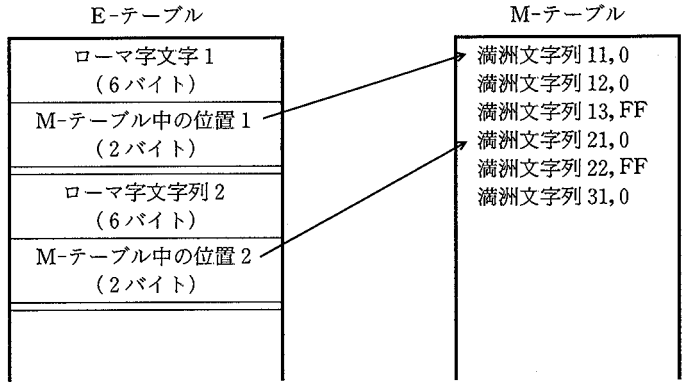
一方，満州文字での入力を担当する者にとってはその区分けは明確であるので，入力時に区分け単位ごとにローマ字で入力し，スペースキーで変換する方法を検討した。しかし，現在得られている変換の知識・情報だけでは，ローマ字列の前後の文字に依存して満州文字列が決まる場合があり[2]，後側の文字に依存する場合には，区分け部分だけが入力されて変換が指示された時点では，変換が確定できない。

そこで，現在までに得られている変換のための情報を整理した結果，区分けされる部分に現れるローマ字文字列は，全部で1,482通りあり，各ローマ字列に対応する満州文字列の候補は，最小で1個，最大で7個，平均3.9個であることが分かった（ちなみに，そのローマ字文字列の文字列長の最大は6文字である）。つまり，前後に依存するものも含めて，区分け部分に対する候補としては，1つのローマ字列に対して高々7個程度である。したがって，変換の方式としては，ローマ字で入力して日本語へ変換する日本語フロントエンドプロ

セッサのように，区分けされたローマ字列に対して，候補となる満州文字列を画面下の方に表示し，その中から適切なものを選択する方法をとることにした。

１つのローマ字列に対応する満州文字列は，個数もその長さもまちまちであり，単にローマ字列と対応する候補の満州文字列を単純に並べただけでは，適切な検索方法が利用できないので，高速な変換を実現するための満州文字列の候補の管理として，最初につぎのような方式を考えた（図II-１）。

区分けされた各ローマ字列に対して，その変換候補の満州文字列を候補間は０で区切り，最後の候補の後はFF（16進数）でで区切って格納する（以後M-テーブルと呼ぶ)。つまり，M-テーブルは満州文字列だけが，０あるいはFFで区切られたテーブルである。さらに，ローマ字列と満州文字列の対応は，６文字（６バイト：この大きさは，区分けされたローマ字列の最大長）分の固定長のローマ字列用の領域と，そこに入るローマ字列に対応する候補の満州文字列の始まるM-テーブル中の位置を指すための２バイトからなるテーブル(以後E-テーブルと呼ぶ)に格納する。そして８バイト/１レコードのE-テーブルへの情報の格納は，ローマ字列部分で辞書式順序でソートしておく。このような二段構えのものとしたのは，１レコード８バイトの固定長レコードのE-テーブルであれば，二分検索で，指定されたローマ字列を含むE-テーブル中の位置を素早く見つけ，そこからM-テーブル中での開始位置がわかり，その位置から始



図II-１　ローマ字から満州文字への変換のテーブル

まり文字列の後の区切りがFFのものまでを満州文字列の候補として見つけられるからである。

この設計に基づいて入出力サブシステムを実現するときに，「E-テーブルとM-テーブルはかなり大きくなり，プログラム全体がデバイスドライバとして組み込むには大きすぎる」という問題が生じた。そこで，E-テーブルとM-テーブルを合わせたものを1つのファイルとして，デバイスドライバとしての入出力サブシステムから独立させ，そのサブシステムが組み込まれた直後に，「初期化にだけ必要なプログラム部分のメモリを解放し，さらに上記ファイル分のメモリを確保し，ファイルから読み込む」ように変更した。しかし，この方法もメモリ確保の段階でエラーが生じた。原因は初期化にだけ必要な部分のメモリを解放しただけでは，確保しようとしたメモリ容量に足りないためだと思われる。

したがって，これら2つのテーブルを全部一度に読み込むのではなく，最小単位を読み込む方法を検討し，その程度のメモリであれば確保できることが分かった。最小単位の大きさとしては，E-テーブル用として1レコード分8バイト，M-テーブル用に7×20バイト程度である。ここで7は1つのローマ字列に対する候補となる満州文字列の最大個数であり，20は「候補となる満州文字列の最大長＋1＋余裕」の値である。

そこで，メモリ確保の大きさは8＋20×7バイトとし，さらにE-テーブルとM-テーブルをそれぞれ1つずつのファイルとした。ローマ字列に対する候補となる満州文字文字列の検索は，E-テーブル用のファイルを読む位置を指定してのダイレクトアクセスの方法を用いたディスクファイル上での二分検索法とした。この方法では，ディスクアクセスが多くなり，変換時の応答スピードが遅くなるのではないかとの心配もあったが，最近のディスクのアクセススピードは高速であるので，応答スピードはほとんど気にならないものとして実現できている。

2. JIS漢字にない漢字の扱い

満州語辞書では中国語単語のフィールドがあり，JIS漢字には割り当てられ

ていない約2,000文字程度の漢字の追加が必要である。これらの漢字（以後，追加漢字と呼ぶ）については，辞書データを入力していく過程で徐々に判明し，フォントエディタで字形のデザインをし，ある程度の個数が揃った段階で，システムで扱えるように組み込むという方法をとっている。システム作成の点からは，必要な追加漢字を最初にすべて洗い出して登録処理する方が手間もかからず簡単であるし，読みの順，あるいは画数など何らかの規則に基づいてコードの割り当てを行うほうが，簡単な入力方法を用意できる。しかし，徐々に追加漢字が判明する現状では出現順のコード割り当てしかできなかった。なお，追加漢字用のコードは16進でF020～FA7Fを割り当て行くことにしている。

このような状況であるので，登録し扱えるようになった追加漢字の入力については，16進でのコード入力の方法とせざるを得なかった。ただし，必要に応じた追加漢字のフォント作成などの担当者と，入力の担当者が同じ人であるので，現在のところ追加漢字のコードを知ることがそれほど負担とはなっていないようである。なお，追加漢字のみからなるファイルも用意しており，編集用のエディタを用いてのデータ入力中に，そのファイルも開き目的とする漢字を取り込む方法も可能である。この方法であれば，追加漢字のコードを調べる必要もない。

III 辞書管理サブシステム

日本語・満州語の辞書作成としては，まず印刷物としての辞書で，その内容は満州語の単語から日本語の単語への辞書，および日本語の単語から満州語の単語への辞書の2つを作成することを目的としている。完成したあと，電子辞書的に扱うことも視野に入れてはいるが，現時点での，辞書管理サブシステムは，作成のための補助システムとして位置づけているものである。

1. フィールド構成

辞書管理サブシステムで管理するレコードのフィールド構成と各フィールドの大きさは，次のようなものとした。

フィールド名	サイズ	
登録番号	4バイト	
参照番号	4バイト	
リンク	4バイト	
満州語	制限無し	
翻字	制限無し	
中国語	制限無し	
中国語読み	制限無し	
日本語	制限無し	”日本語”と”日本語読み”は組で通常は
日本語読み	制限無し	8組、最大16組記入できる。
：		
日本語	制限無し	
日本語読み	制限無し	

”満州語”以降のフィールドサイズの「制限無し」の扱いは，まず30バイトの大きさを確保し，それ以上の大きさに対しては，30バイト単位で増加して対応し，使用上は無制限の大きさの内容を記述できるようにしている。

1つの満州語単語に対して，複数の日本語単語が対応するので，”日本語”と”日本語読み”のフィールドは最初は8組を用意している。これらのフィールドはほとんどの場合は，8組あれば十分であるが，万一不足する場合には，最大16組まで利用できるように対応している。ただし，追加の8組分はレコード内に確保するのではなく，”リンク”フィールドで指される別領域にとることにしている。この別領域は同じファイル内に1レコード分の領域を確保して用いる。領域的には無駄があるが，8 組より多い場合はほとんどない（実際これまでに入力されたデータではそのような場合は生じていない）こと，および処理が簡潔に実現できることなどからこのような方式とした。

2. 検　　索

上記のフィールドのうち，”リンク”以外のフィールドを検索に用いることが

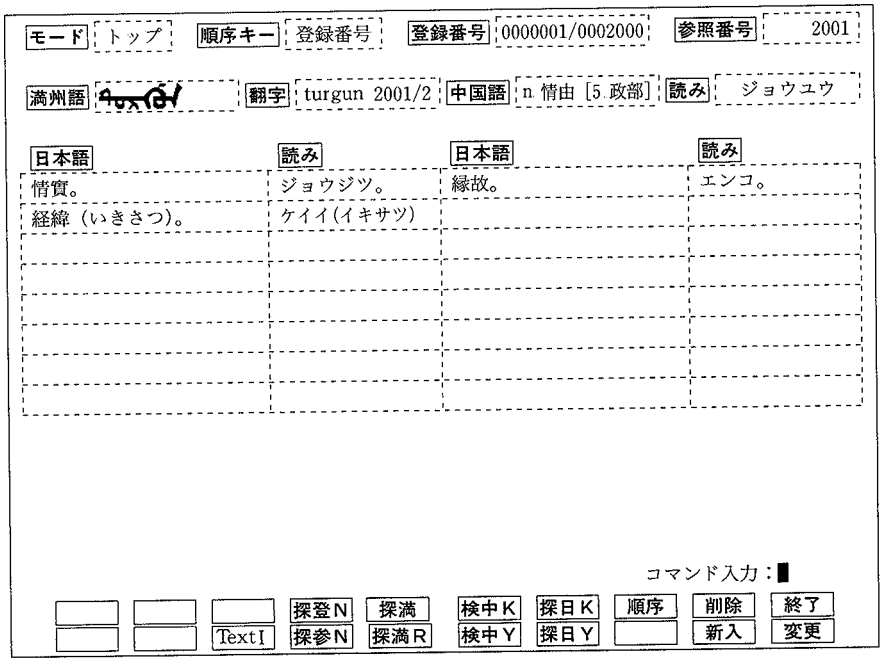
できるように設計した。つまり，登録番号，参照番号，満州語，中国語，中国語読み，日本語，日本語読み，での検索ができる。市販されている多くのデータベースソフトでは，ここでの"日本語"フィールドのように同じ性格のフィールドを複数持つレコードに対して検索する場合でも，それぞれのフィールド毎に検索をしなければならないが，(1) 本システムでは，"日本語"での検索のときは，複数ある"日本語"のフィールドすべてを検索対象とするようにしている。このことは，本システム実現のために市販のデータベースソフトを用いずに，ISAMライブラリ[1]を用いてC言語での自作のシステムとした1つの理由である（もう1つの理由は満州文字の表示の可・不可である）。"日本語読み"のフィールドも検索においては同様な処理としている。

なお，"登録番号"と"参照番号"以外のフィールドの場合，フィールド長は無制限であるが，検索キーとしては入力されたキーのうち先頭から最大30バイトを用いる。ただし，それ以下の入力（8バイト以上のキー値を検索する場合には，最低8バイト分は必要）でも可能とし，その場合には前方一致検索を行う。

3. 画面表示と操作

辞書管理システムのCRT画面での表示を図III-1に示す。1画面には1レコード分の情報を表示する。なお，"満州語"以下のフィールドのバイト数は上記で説明したように実用上は無制限であるが，画面上でのそれらの表示は高々，半角英数字で30文字分としている。現在のデータではほとんどの場合，各フィールドはこの文字数分以下であるが，もしそれ以上の大きさの内容になった場合には，以下で説明するレコード内容の変更処理のモードにすれば，画面下方に半角文字で78文字×7行分の大きさの表示域に表示できる。この文字数よりもさらに大きな内容の場合には，スクロールして表示することもできる。

(1) リレーショナルデータベースシステムで，複数のテーブル及びクエリを用いれば，1つのフィールドに対しての検索で行うことも可能ではあるが，処理が多少複雑となる。



図III－1　辞書管理システムの画面表示

モード欄に表示されるのは"トップ"，"新入中"，"変更中"，"探索中"の4つである。"トップ"は，システム起動時の状態であるとともに，操作のコマンドを受け付ける状態である。他のモードは，それぞれ，新規のレコード入力，既入力のレコードの変更，あるフィールドを対象としての探索，の処理中であることを示す。これらの処理中の画面表示も，ほぼこの"トップ"の表示と同様なものであるので，モード欄で現在処理中の内容を示すことにしている。

これらの処理を含むコマンドの指定には，"トップ"のモードで，ファンクションキーおよびシフトキーとファンクションキー，ROLL UP，ROLL DOWN，HOME，CLR を用いる。対応は，以下のようである。ただし，Fx はファンクションキーxをS-Fx はシフトキーとファンクションキーxを示すとする。

F1　　（対応無し）

F2	(対応無し)
F3	テキストファイルからのデータ入力
F4	参考番号での検索
F5	満州語翻字 (ローマ字) での検索
F6	中国語読みでの検索
F7	日本語読みでの検索
F8	(対応無し)
F9	新規レコード入力
F10	レコード内容の変更

S-F1	(対応無し)
S-F2	(対応無し)
S-F3	(対応無し)
S-F4	登録番号での検索
S-F5	満州語 (漢字) での検索
S-F6	中国語 (漢字) での検索
S-F7	日本語 (漢字) での検索
S-F8	次レコードへの移動で用いるフィールドの指定
S-F9	現レコードの削除
S-F10	辞書管理システムの終了

ROLL UP	現在選択されている順序キーでの次のレコードを表示
ROLL DOWN	現在選択されている順序キーでの前のレコードを表示
HOME	現在選択されている順序キーでの最初のレコードを表示
CLR	現在選択されている順序キーでの最後のレコードを表示

本プロジェクトでは，システム開発と平行して，市販の表形式のデータベースソフトを用いて，参照番号，ローマ字，中国語，中国語読み，日本語，日本語読みのフィールドの部分だけのデータ入力を進めていた。そのデータを利用するために，F3 でのテキストファイルからの入力の機能を設けた。この機能は，IVで述べるデータ入力・編集用のエディタシステムで作成・修正したデータファイルから読み込むのにも利用できるものである。

辞書の作成途中では，レコードをあるフィールドで検索して表示する機能が必要なことは言うまでもないが，表示するレコードの指定としては，検索機能を用いるだけでなく，たとえば，現在表示されているレコードから前後に移動する方法も有用かつ必要である。本サブシステムで扱うデータの場合，検索対象となるフィールドに含まれるデータは，フィールドごとに数値としての大小，辞書式，あるいはコードなどで順序がつけられている。フィールドは複数あるので，前後という場合，どのフィールドでの順序での前後かを決めておかなければならない。そこで，S-F8 を押すごとに順序の対象とするフィールドが変わるようにした。順序の対象とするフィールドを決めれば，ROLL UP，ROLL DOWN，HOME，CLR のキーを用いて，その選択されている順序でのレコードの移動・表示を行うことができる。ただし，複数個ある"日本語"は，この順序の場合には１つのフィールドとして扱う。そのため，現レコードが選択された要因となった"日本語"のフィールドがどの部分であるかが分かるように枠を紫色で示している（通常，枠は緑色で表示）。"日本語読み"についても同様に処理している。

F10 でレコードの内容の変更を指定した場合，モード欄は"変更中"になり，画面下方に 78 文字×７行の編集用領域が示され，最初に満州語単語のフィールドの内容が表示される。この表示域内では，カーソル移動には矢印キーなど，IV で述べるデータ入力・編集用のエディタのサブセットではあるが，通常の編集作業には十分なコマンドが利用できる。別のフィールドを編集対象とする場合には，ROLL UP，ROLL DOWN のキーを用いる。ただし，登録番号のフィールドは新規入力時に自動的に決められるので，当然のことであるが変更できず，したがって ROLL UP，ROLL DOWN のキーでも，そのフィールドは選択できないようになっている。この"変更中"のモードを終了する場合には，保存の有無により F10 あるいは S-F10 を用いる。

現在の辞書管理サブシステムは，大まかには以上のようなものであるが，最終的な目標である紙面への印刷と，日本語単語から満州語単語への辞書の作成の機能については，まだ実現していない。本プロジェクトの進行として，現在

は，データ入力に全力を注いでおり，それが一段落した時点で，印刷の形式を検討することにしている。印刷で用いるユーティリティとしてはTeXを考えており，Tex用の満州文字フォントを用意するなど，そのための調査・試作も行っている。

現在のデータ入力が完了すれば，日本語単語での検索機能と同じ方法で，ISAMライブラリ関数を用いて日本語単語から満州語単語への辞書のためのデータをプログラムで作成することは比較的簡単に実現できると思っている。

IV　データ入力・編集用のエディタシステム

IIIでも述べたが，本プロジェクトでは，システム開発と平行して参照番号，ローマ字，中国語，中国語読み，日本語，日本語読みのフィールドの部分だけのデータ入力を，市販の表形式のデータベースソフトを用いて進めていた。したがって，そのデータを有効に利用するために，データをテキストファイルの形で取り出すことが必要であった。テキストファイルでの辞書データは，辞書管理システムに読み込むために，レコード間の区切りをコントロールEと改行で，レコード内のフィールド間の区切りをコントロールNと改行で示すことにした。フィールド数を決めて，各フィールドの区切りを改行だけとしなかったのは，長いフィールドの場合を配慮したためである（ただし，現在までのデータ入力では，そのような長いものは出現していないようである）。

当初は，既に入力されたデータをテキストファイルとして取り出し，IIIで述べたデータベース機能を備えた辞書管理サブシステムに取り込み，内容的なミスの修正，満州単語と中国単語の部分の入力，さらには新規のレコード（新規データ）入力も行うことを予定していた。しかし，初期のデータの入力は数人の学生アルバイトで行ったということであり，参照番号の重複や，フィールド記述の順序が学生により異なるなど多くのミスを含んでいたため，単純に変換するだけではなく，辞書管理サブシステムに読み込ませる前に，そのようなミスを修正しておく必要があった。

ミスの検出はプログラムを作成して計算機で行えたが，ミスが多様であった

ためその修正はプログラムで機械的に行うことはできず，編集用エディタで行った。その修正作業の経験から，編集用エディタでは，満州文字や中国語用の追加漢字も含めて全フィールドの入力もできるので，単なるデータ入力であれば，編集用エディタが有効に利用できると考えられた。

そこで，意味的入力ミスのチェック・修正や，新規レコード（新規データ）の入力を行う担当者に，試験的にその作業に編集用エディタを試用してもらった。その結果，辞書システムに取り込んでの作業では，"変更中"のモードにし，フィールド移動にROLL UP，ROLL DOWNキーを用いるなど，データ部分の入力以外のキー操作が多くなること，1つの満州語単語の表示で1画面を占めるため，入力しているレコードの前後を見るのにも，"変更中"のモードを終了して"トップ"のモードに戻り，ROLL UPあるいはROLL DOWNで前後のレコードを表示するようにキー操作が多くなること，などから単なるエディタの方がデータの新規入力・修正・編集をしやすいということであった。したがって，現在は，修正や新規レコード入力は主として，編集用エディタを用いて行っている。

V おわりに

IVで述べたように，そもそもは入力作業用に考えていた辞書管理サブシステムではデータ部分の入力以外のキー操作が多くなるため，レコードとフィールドの区切り用に，コントロール文字を用いる程度の単純な構造のテキストファイルとしてデータを扱えるエディタの方が，最初のデータ入力・編集・修正には利用しやすいということであった。もちろん，ランダムな順で入力されたデータをいくつかのフィールドについて辞書式の順序で整理して検索を行うためには，辞書管理サブシステム的なものが必要であるし，満州語単語を主体にした辞書データから，日本語単語を主体にした辞書データを作成する場合も，辞書管理サブシステム的なものが必要ではある。また，電子辞書的なものとする場合には，最終的には辞書管理サブシステム的なものが必要であろう。しかし，電子辞書の場合も含めて，データ入力などの辞書の作成過程では，この場合の

ように，必ずしも最終的なものを用いるのではなく，それぞれの過程で適切なものを利用することも作業のし易さ，作業効率などからは重要なことであろう。このことは，我々のもう一つの目的である日本語・ブルガリア語辞書の作成およびそのための補助システムの開発にも大いに参考になることである。

なお，本辞書作成システムは，開発時に主流であったNECのPC-9801シリーズ上のMS-DOS上でのみ稼働するものとして開発してきたが，現在ではDOS/V機も含めてWindows 95/98が主流となってきた。このように開発開始時点と，現在ではパソコン機種およびオペレーティングシステムなどの環境が大きく変わり，編集用エディタ，辞書管理サブシステムなどで，早急なWindows 95/98あるいはWindows NTなどへの対応が必要になってきている。そのため，現在，満州文字やスラブ文字などのTrueTypeフォントの作成，登録，利用などについても対応を検討し，試作を行っている。

本研究は平成9年度に認められた科学研究費補助金「少人数での辞書編纂のための支援システム——日本・ブルガリア語の活用・学習辞典編纂を例にして——」（平成9年度から平成11年度までの3年間）を受けて進めているものの一環であり，平成9年度および10年度の成果の一部である。

参 考 文 献

[1] 本田道夫，藤健児「データ管理の技法 ISAM ライブラリとその応用」CQ出版，1992

[2] 本田道夫，今井慈郎「日本語・満州語の辞書作成のための補助システム（Ⅰ）」『香川大学経済論叢』第67巻第3・4号（127-141），1995