

TTY無手順端末による自動翻訳システムATLAS II の利用

平良, 豊
佐賀大学理工学部物理教室

<https://doi.org/10.15017/1468161>

出版情報：九州大学大型計算機センター広報. 20 (5), pp.463-474, 1987-09-25. 九州大学大型計算機センター
バージョン：
権利関係：

TTY 無手順端末による自動翻訳システム ATLAS II の利用

平 良 豊*

1. はじめに

これからのコンピュータの端末はデータの転送における信頼性の向上等の理由から、しだいに有手順系のものに移行するものと思われる。実際に最近開発されたソフトウェアの大部分は有手順端末のみをサポートしている。しかし大学においては既に数多くの無手順端末が稼働しており、これらの資産を無視することはできない。また無手順端末は有手順端末に比べて安価であり、パソコンでその機能をエミュレートすることもできるので、大学の研究室等においては今後もしばらくは無手順端末が主流であると思われる。

筆者は以前この広報でTTY無手順による日本語に対応したパソコン用の端末エミュレータを発表した[1]。その後、数人の方からそのエミュレータでODMの自動翻訳システムATLAS IIを使用できるかとの問い合わせを受けた。ODMの起動が有手順の日本語端末に限られており、ATLAS IIはODMの配下で動作するように作られているので、無手順端末では利用できないとお答えした。しかし、ATLAS IIのマニュアル[2]を読み返してみても、一括翻訳(ODMの既存文書処理の翻訳(H))はバッチで処理されていることがわかった。そこで、ODM文書をTTY無手順端末で直接作ることができれば、一括翻訳によりATLAS IIを使用できるのではないかと考えた。この考えに沿ってATLAS IIのTTY無手順端末による利用を試みた。幸い、九大の大型計算機センターでは今年の4月からJOIS型漢字端末(または、そのエミュレータ)での漢字の入力が可能になり、日本語の文書ファイルを容易に作るできるようになった。(公開した筆者のエミュレータではJOISモードがこれに対応している。)

本稿では無手順JOIS型漢字端末による

- 1) ODM文書の作成
- 2) ATLAS IIによる自動翻訳処理の起動
- 3) 翻訳された文書の一般データセットへの取り出し

を説明する。

2. ODM文書の作成

ODM文書は区分編成データセット中にメンバー名ODMXXXXXとして管理されている。ここでXXXXXは5桁の整数で、これが文書識別番号(DOCID)としても使用されている。このデータセットのDCB(Data Control Block)は

- ・レコード形式 VB
- ・レコード長 258 バイト

昭和62年7月8日受理

*佐賀大学理工学部物理

・ブロックサイズ 2048バイト

である。

ODM文書は文書管理情報部と日本語文書部から構成されている。文書管理情報部は文書管理に必要な多くの項目（文書識別番号、文書名、作成日、制作者名、更新日、更新回数など）を含む。日本語文書部はシフトコードを含まないJEF漢字コードからなる文書文字列と書式制御文字からなる。日本語文書部の各レコードは先頭がX'1500'（16進数）で始まり、次に参照番号（初期値1000、増分値1000の4バイト）が続き、その後8バイトのNUL（X'00'たぶんこれも参照番号の一部を形成しているものと思われる）を持ち、15バイト目からテキストが始まる。テキストの最終レコードは制御コードX'30A9'で終る。例を以下に示す。（バイトの上位を上に、下位を下に書く）

```
10000E000000000BECDACCAACAAAEAA  
50003800000000267A4F7D47424B15
```

我輩は猫である。

```
107FFF000000003A  
50FFFF0000000009
```

（テキストの最後を示すレコード）

文書管理情報部の詳細な構成は現在のところ明らかにされていない。そこで既存のODM文書から管理部を取り出し、文書識別番号、作成日、制作者名（所有者名）を更新して、管理部として使用することにする。それらのパラメータの文書レコード中に占める位置を参考のため以下に示す。

- ・文書識別番号：第2レコードの15-19の5バイト
- ・作成日：第2レコードの80-87の8バイト
- ・制作者名：第2レコードの96-102の7バイト
（所有者名）：（第3レコードの23-29の7バイト）

JOIS型漢字端末で作った日本語入力ファイルから行番号とシフトコードを取り除き、テキスト部を作り、これに日本語レコード識別コード（X'1500'）と参照番号を付けて日本語文書部を作る。先に作った文書管理情報部に追加することでODM文書ができあがる。

以上の作業を行なうプログラム（JOISODM）と使用するDD名を示す。記述言語としてパスカルを用いた。

使用するDD名は

- ・ODMHEAD：文書管理情報部を取り出すためのODM文書を割り当てる。
（ODMHEAD. TEXT）
- ・JOISIN：ODM文書に変換されるべき入力データセットを割り当てる。
JOIS型漢字端末で制作された日本語ファイルで、内容修飾識別子がTEXT（VB、ブロック長3120、レコード長255）で行番号付き。
（JOIS. TEXT）
- ・ODMOUT：出力されるODM文書のデータセットを割り当てる。
ODM. TEXT（ODMXXXXX）
XXXXXは5桁の整数で文書識別番号（DOCID）と同じでなければならない。

- ・ INPUT : 文書識別番号, 作成日, 製作者名を与える. 与え方は次の通り.

DOCID=XXXXX

DATE=YY.MM.DD

```
00010 (* THIS PROGRAM CONVERTS JOIS TO ODM *)
00020 (* ODMHEAD : ODMHEAD.TEXT *)
00030 (* JOISIN : JOIS.TEXT *)
00040 (* ODMOUT : ODM.TEXT(ODM000XX) *)
00050 (* '87 JUL 01 Y. H. *)
00060 PROGRAM JOISODM;
00070 TYPE STRIN4=STRING(4);
00080 VAR
00090 LN0, LN1, LN2:STRING(255);
00100 BUNNO:STRING(5); DBUNNO:STRING(6);
00110 DATE:STRING(8); DDATE:STRING(5);
00120 USERID:STRING(7); DUSERID:STRING(7);
00130 C:CHAR;
00140 FO, FI, FH:TEXT;
00150 HEAD, NUL8, ENDTEXT, HEADLN:STRING(20);
00160 STR1, STR2:STRING(255);
00170 I, J, X:INTEGER; Z:ARRAY(.1..4.) OF INTEGER;
00180 PROCEDURE INIT;
00190 VAR I:INTEGER;
00200 BEGIN
00210 HEAD:='1500'XC;
00220 NUL8:='0000000000000000'XC;
00230 ENDTEXT:='15007FFFFFFFF0000000000000000000030A9'XC;
00240 READLN(DBUNNO, BUNNO);
00250 IF DBUNNO='DOCID=' THEN WRITELN('DOCID=', BUNNO)
00260 ELSE BEGIN
00270 WRITE('*** ERROR *** (DOCID=XXXXX) : EXPECTED ');
00280 WRITELN(' DOCID=00001 ASSUMED ');
00290 BUNNO:='00001';
00300 END;
00310 READLN(DDATE, DATE);
00320 IF DDATE='DATE=' THEN WRITELN('DATE=', DATE)
00330 ELSE BEGIN
00340 WRITE('*** ERROR *** (DATE=YY.MM.DD) : EXPECTED ');
00350 WRITELN(' DATE=87.01.01 ASSUMED ');
00360 DATE:='87.01.01';
00370 END;
00380 READLN(DUSERID, USERID);
00390 IF DUSERID='USERID=' THEN WRITELN('USERID=', USERID)
00400 ELSE BEGIN
00410 WRITE('*** ERROR *** (USERID=A7XXXXX) : EXPECTED ');
00420 WRITELN(' USERID=A79999A ASSUMED ');
00430 USERID:='A79999A';
00440 END;
00450 END; (* INIT *)
00460 FUNCTION CONHEX(X:INTEGER):STRING;
00470 VAR I, J:INTEGER; S:STRING(4);
00480 BEGIN
00490 FOR I:=1 TO 4 DO BEGIN
00500 Z(I.):=X MOD 256;
00510 X:=X DIV 256;
00520 END;
00530 S:='';
00540 FOR I:=1 TO 4 DO BEGIN
00550 S:=STR(CHR(Z(I.))) || S;
00560 END;
```

```

00570     CONHEX:=S;
00580 END; (* CONHEX *)
00590 FUNCTION JOISIN:STRING;
00600 VAR I,N:INTEGER; KANJI:BOOLEAN;S,STR1:STRING(255);
00610 BEGIN
00620   READLN(FI,STR1);S:='';
00630   FOR I:=9 TO LENGTH(STR1) DO BEGIN
00640     N:=ORD(STR1(.I.));
00650     CASE N OF
00660       '28'X:KANJI:=TRUE;
00670       '29'X:KANJI:=FALSE;
00680       OTHERWISE
00690         IF KANJI THEN BEGIN ;
00700           S:=S || STR(STR1(.I.));
00710         END;
00720     END;
00730   END;
00740   JOISIN:=S;
00750 END; (* JOISIN *)
00760 BEGIN (* MAIN START *)
00770   INIT;
00780   OPEN(FI,'DDNAME=JOISIN,RESET');
00790   OPEN(FH,'DDNAME=ODMHEAD,RESET,BLKSIZE=2048,LRECL=258,RECFM=VB');
00800   OPEN(FO,'DDNAME=ODMOUT,REWRITE,BLKSIZE=2048,LRECL=258,RECFM=VB');
00810   READLN(FH,LN0);WRITELN(FO,LN0);
00820   READLN(FH,LN1);
00830   FOR I:=1 TO 5 DO LN1(.14+I.):=BUNNO(.I.);
00840   FOR I:=1 TO 8 DO LN1(.79+I.):=DATE(.I.);
00850   FOR I:=1 TO 7 DO LN1(.95+I.):=USERID(.I.);
00860   WRITELN(FO,LN1);
00870   READLN(FH,LN2);
00880   FOR I:=1 TO 7 DO LN2(.22+I.):=USERID(.I.);
00890   WRITELN(FO,LN2);
00900   WHILE NOT EOF(FH) DO BEGIN
00910     READ(FH,C);
00920     IF C<>CHR('15'X) THEN BEGIN
00930       WRITE(FO,C);
00940       WHILE NOT EOLN(FH) DO BEGIN
00950         READ(FH,C);WRITE(FO,C);
00960       END;
00970       READLN(FH);WRITELN(FO);
00980     END ELSE BEGIN
00990       WHILE NOT EOLN(FH) DO BEGIN
01000         READ(FH,C);
01010       END;
01020       READLN(FH);
01030     END;
01040   END;
01050   J:=1000;
01060   WHILE NOT EOF(FI) DO BEGIN
01070     HEADLN:=HEAD || CONHEX(J) ||NUL8 ;
01080     J:=J+1000;
01090     WRITELN(FO,HEADLN,JOISIN);
01100   END;
01110   WRITELN(FO,ENDTEXT);
01120   CLOSE(FO);CLOSE(FI);CLOSE(FH);
01130 END. (* MAIN *)

```

図1 ODM文書を作成するプログラム JOISODM

: USERID = A 7 9 9 9 9 A

・ OUTPUT : モニター出力。 SYSOUT = * を割り当てる。

プログラムの説明

手続き INIT (行番号 180-450) で変数を初期化し、文書識別番号 (DOCID)、作成日 (DATE)、製作者 (USERID) を読み込む。関数 CONHEX (行番号 460-580) は行番号 (整数)

を16進数に変換して文書の参照番号を作る。関数JOISIN(行番号590-750)はJOIS.TEXTから文書を読み込み、シフトコードを削除し、JEFコードのみとする。このとき半角の文字列は削除される。行番号750からがMAINで、行番号810-1040でODMHEAD.TEXTを読み込み、文書識別番号、作成日、制作者名の更新を行なう。行番号1050-1100は関数JOISINで読み込まれた文字列に、レコードヘッド('X'1500')と参照番号を付けてODM.TEXTに書き出す。最後にテキストの終わりを示すENDTEXTを出力し、すべてのファイルを閉じて処理を終る。

使用方法

実行形式をATLAS 2. LOAD (JOISODM) に作り*、以下のJOB文で実行する。

この例ではODM文書としてODM.TEXT (ODM00001) を指定している。INPUTのデータはレコードの先頭から3行にわたって書かれる。

```
00010 //A79999AA JOB PASSWORD,CLASS=A
00020 //*****
00030 //*          JOIS --> ODM          *
00040 //*****
00050 //JOISODM EXEC PGM=JOISODM,REGION=4096K
00060 //STEPLIB DD DSN=A79999A.ATLAS2.LOAD,DISP=SHR
00070 //ODMHEAD DD DSN=A79999A.ODMHEAD.TEXT,DISP=SHR
00080 //JOISIN DD DSN=A79999A.JOIS.TEXT,DISP=SHR
00090 //ODMOUT DD DSN=A79999A.ODM.TEXT(ODM00001),DISP=SHR
00100 //INPUT DD *
00110 DOCID=00001
00120 DATE=87.07.01
00130 USERID=A79999A
00140 /*
00150 //OUTPUT DD SYSOUT=*
```

図2 JOISODMを実行するジョブ制御文

3. ATLAS IIによる翻訳

ODMの既存文書処理(3)の翻訳処理(H)でODM文書が一括翻訳される。その時、出力されるJOB制御文をもとにして、図3の翻訳のための制御文を作成した。

変更した箇所は行番号450のSYSSAVEのDUMMY(元はODMTEMP1.NXXXXXX)が作られるが、これが既に存在している時はエラーを生じる)。追加した箇所は翻訳のための専門辞書の項(行番号330-360, 410-440)で、ここでは物理(GK021E/J)と数学(GK013E/J)を指定している。辞書番号の大きいものが優先される。その他の専門辞書のデータセット名については[3]を参照して下さい。

*実行形式ロードモジュールの作り方:ソースファイルがATLAS 2. PASCAL (JOISODM)に格納されているとする。まず、つぎのパスカルコマンドで翻訳し、オブジェクトモジュールをATLAS 2. OBJ (JOISODM)に作る。

```
PASCAL ATLAS 2 (JOISODM) OBJECT(ATLAS 2 (JOISODM)) NOGO
次に、LINKコマンドでパスカルのライブラリPASLIBを結合して、ロードモジュールを作る。
LINK ATLAS 2 (JOISODM) LOAD(ATLAS 2 (JOISODM)) PASLIB
```

```

00160 //*****
00170 //*          ATLAS2          *
00180 //*****
00190 //TRANS      EXEC PGM=JSYTRNS,REGION=4096K
00200 //STEPLIB    DD  DSN=PP1.ODM41.CMDLIB,DISP=SHR
00210 //           DD  DSN=PP1.FDF41.CMDLIB,DISP=SHR
00220 //           DD  DSN=PP1.ATF30.LINKLIB,DISP=SHR
00230 //           DD  DSN=PP1.ATL21.CMDLIB,DISP=SHR
00240 //           DD  DSN=PP1.ATM21.GRAMMAR,DISP=SHR
00250 //TEXT      DD  DSN=A79999A.ODM.TEXT,
00260 //           UNIT=PUB,DISP=SHR
00270 //ODMUSERM  DD  DSN=PP1.ODM40.USERLIB,
00280 //           UNIT=SYSDA,DISP=SHR
00290 //JWDDICT   DD  DSN=PP1.ATN.JWDDICT,
00300 //           UNIT=SYSDA,DISP=SHR
00310 //JWDDICT1  DD  DSN=PP1.ATLAS2.IN210J,
00320 //           UNIT=SYSDA,DISP=SHR
00330 //JWDDICT2  DD  DSN=PP1.ATLAS2.GK013J,
00340 //           UNIT=SYSDA,DISP=SHR
00350 //JWDDICT3  DD  DSN=PP1.ATLAS2.GK021J,
00360 //           UNIT=SYSDA,DISP=SHR
00370 //EWDDICT   DD  DSN=PP1.ATO.EWDDICT,
00380 //           UNIT=SYSDA,DISP=SHR
00390 //EWDDICT1  DD  DSN=PP1.ATLAS2.IN210E,
00400 //           UNIT=SYSDA,DISP=SHR
00410 //EWDDICT2  DD  DSN=PP1.ATLAS2.GK013E,
00420 //           UNIT=SYSDA,DISP=SHR
00430 //EWDDICT3  DD  DSN=PP1.ATLAS2.GK021E,
00440 //           UNIT=SYSDA,DISP=SHR
00450 //SYSSAVE   DD  DUMMY
00460 //SYSPRINT  DD  SYSOUT=*
00470 //SYSLIST   DD  SYSOUT=(O,KNGWTR),
00480 // COPIES=1
00490 //SYSIN     DD  *
00500 / USER UID=A79999A
00510 / LIBRARY  PRIV=TEXT
00520 / TRANS   RANGE=(0001,9999),
00530           PRINT=YES,
00540           RETRANS=YES,
00550           REP=YES,
00560           TIME=4
00570 / SELECT  DOCID=00001
00580 / PAGE    FORM=66,
00590           TOPLINE=4,
00600           BLOCKL=60,
00610           SHIFT=2,
00620           POINT=9,
00630           BLOCK=1,
00640           BLOCKW=39,
00650           LPI=6,
00660           MODE=P1,
00670           PAGEW=109,
00680           SMALL=(YES,YES),
00690           IMAGE=(YES,YES),
00700           LAND=NO,
00710           DEV=(NLPE,NLPE)
00720 / FORMAT  LIST=FMT,
00730           OUTLIM=(0001,9999),
00740           PAGENO=1,
00750           DATE=870611,
00760           TIME=153409,
00770           COMMENT=NO,
00780           IXUL=NO,
00790           TEXT=TEXT,
00800           CT=NO,
00810           IX=NO,
00820           LNSPACE=1
00830 /*

```

図 3. 翻訳を実行するジョブ制御文

翻訳される ODM 文書は行番号 250 の TEXT でデータセット名を、行番号 570 の DOCID でメンバー名を指定されている。ここでは ODM TEXT (ODM00001) である。行番号 550 の REP = YES の指定により、翻訳結果は元の文章と置き換えらる。一文あたりの翻訳時間は 4 秒としている。この時間をむやみに大きくしても訳文の質には影響はなく、むしろ原文が翻訳に不相当である時にロスを発生させる恐れがある。その他のパラメータについてはセンターニュースの XFORM についての記事 [4] が参考となるであろう。

4. 訳文の一般データセットへの取り出し

翻訳された文書は一文ごとに原文と訳文が対を成して作られる。その形式は

<u>対訳はじめ</u>	<u>日本語はじめ</u>	<u>日本語文</u>	<u>英語はじめ</u>	<u>英文</u>	<u>対訳おわり</u>
<u>制⇒制</u>	<u>制日制</u>		<u>制英制</u>		<u>制⇐制</u>

である。英文は

半行 I 行行 a m 行行 a 行行 c a t . 行 半

のように格納される。ここで、半…半は…がピッチ詰め（半角表示）されることを示し、行…行は…が行で分割されないことを示す。

内部制御コードを 16 進数で示すと、

・ <u>制</u> : 3 0 E A	・ <u>制</u> : 3 0 E B
・ <u>半</u> : 3 0 C 0	・ <u>半</u> : 3 0 C 1
・ <u>行</u> : 3 0 D 8	・ <u>行</u> : 3 0 D 9
・⇒: 7 6 B 3	・⇐: 7 6 B 2 (これは富士通拡張文字)

である。

対訳文書の日本語文書部の例を 16 進表示で示す。

```
10000E0000000003E7B3E30CF3EBECDACCAACAAAEAA
500038000000000A630B0A6C0B267A4F7D47424B15
```

制 ⇐ 制 制 日制 我輩は猫である。

```
10000D0000000003EBD3E3C3DAC443D3DAEAE443D3DAE443D3DAEAEAFAA
500070000000000A110B000839000908313D0009083100090833313415
```

制 英制 半 行 I 行行 a m 行行 a 行行 c a t .

```
443D443C3E7B3E
```

```
000900010A620B
```

行 半 制 ⇐ 制

```
107FFF000000003A
```

```
50FFFF0000000009
```

制御コードを目印にして対訳文と英文を別々に取り出すことができる。後でパソコンのワープロで編集ができるように、英文を半角に変換して取り出す。

以上の作業を行なうプログラム (ODMJ OIS) を図 4 に示す。

```

00010 PROGRAM ODMJOIS;
00020
00030 (* THIS PROGRAM CONVERTS ODM FILE TO JOIS FILE *)
00040 (* USE FILE *)
00050 (* ODMOUT :ODM FILE TO BE CONVERTED *)
00060 (* JOISOUT :JOIS FILE FOR TAIYAKU OUTPUT *)
00070 (* ANKOUT :ANK FILE FOR ENGLISH OUTPUT *)
00080 (* 87.06.16 Y. H. *)
00090
00100 VAR
00110 LEN :INTEGER;
00120 FI,FO,FOE:TEXT;
00130 UPPER,LOWER,NUMBER:STRING(26);
00140 KIGOU1,KIGOU2,KIGOU3:STRING(32);
00150 C,C1,C2 :CHAR;
00160 NC1,NC2:INTEGER;
00170 KANJI,HANKAK,BUNKIN,SEIGYO:BOOLEAN;
00180 TAIYAKU,EIBUN,WABUN:BOOLEAN;
00190 BUF,JBUF:STRING(255);
00200 DUMMY:STRING(13);
00210 KI,KO:STRING(1);
00220 I,J,K:INTEGER;
00230
00240
00250 PROCEDURE INITIALIZE;
00260 BEGIN
00270 UPPER:='ABCDEFGHIJKLMNOPQRSTUVWXYZ';
00280 LOWER:='abcdefghijklmnopqrstuvwxyz';
00290 NUMBER:='0123456789';
00300 (* 0123456789ABCDEF0123456789ABCDEF *)
00310 KIGOU1:=' . . . ; ? ! ~ - / ; (* A1A0-A1BF *)
00320 KIGOU2:=' | " ( ) [ ] { } - + - ; (* A1C0-A1DF *)
00330 KIGOU3:=' = < > ¥ $ % # & * @ ' ; (* A1E0-A1FO *)
00340 KIGOU2(.8.):=''''';
00350 KI:=STR(CHR('28'X)); KO:=STR(CHR('29'X));
00360 SEIGYO:=FALSE;KANJI:=FALSE;HANKAK:=FALSE;BUNKIN:=FALSE;
00370 TAIYAKU:=FALSE;EIBUN:=FALSE;WABUN:=FALSE;
00380 END;
00390
00400 BEGIN
00410 INITIALIZE;
00420 OPEN(FI, 'DDNAME=ODMOUT,RESET,BLKSIZE=2048,LRECL=258,RECFM=VB');
00430 OPEN(FO, 'DDNAME=JOISOUT,REWRITE');
00440 OPEN(FOE, 'DDNAME=ANKOUT,REWRITE');
00450 WHILE NOT EOF(FI) DO BEGIN
00460 JBUF:=''; KANJI:=FALSE;BUF:='';
00470 READLN(FI,C,DUMMY,BUF);
00480 IF C=CHR('15'X) THEN BEGIN
00490 LEN:=LENGTH(BUF) ;LEN:=LEN DIV 2;
00500 IF LEN <>0 THEN BEGIN
00510 FOR I:=1 TO LEN DO BEGIN
00520 C1:=BUF(.2*I-1.); C2:=BUF(.2*I.);
00530
00540 CASE ORD(C1) OF
00550
00560 '30'X:CASE ORD(C2) OF
00570 'EA'X:SEIGYO:=TRUE;
00580 'EB'X:SEIGYO:=FALSE;
00590 'CO'X:BEGIN
00600 HANKAK:=TRUE;
00610 IF KANJI THEN BEGIN

```

```

00620                                JBUF:=JBUF !! KO;
00630                                KANJI:=FALSE;
00640                                END;
00650                                END;
00660                                'C1'X:HANKAK:=FALSE;
00670                                'D8'X:BUNKIN:=FALSE;
00680                                'D9'X:BUNKIN:=FALSE;
00690                                OTHERWISE BEGIN END;
00700                                END; (* CASE C2 *)
00710
00720                                '40'X:IF ORD(C2)='40'X THEN BEGIN
00730                                    IF HANKAK THEN JBUF:=JBUF !! ' '
00740                                        ELSE JBUF:=JBUF !! ' ' ;
00750                                END;
00760
00770                                '76'X:CASE ORD(C2) OF
00780                                    'B3'X:IF SEIGYO THEN TAIYAKU:=TRUE;
00790                                    'B2'X:IF SEIGYO THEN TAIYAKU:=FALSE;
00800                                    OTHERWISE BEGIN END;
00810                                END; (* CASE C2 *)
00820                                'A1'X:BEGIN
00830                                    IF HANKAK THEN BEGIN
00840                                        NC2:=ORD(C2);
00850                                        CASE NC2 OF
00860                                            'A1'X..'BF'X:
00870                                                JBUF:=JBUF !! STR(KIGOU1(.NC2-'9F'X.));
00880                                            'C3'X..'DD'X:
00890                                                JBUF:=JBUF !! STR(KIGOU2(.NC2-'BF'X.));
00900                                            'E1'X..'F7'X:
00910                                                JBUF:=JBUF !! STR(KIGOU3(.NC2-'DF'X.));
00920                                            OTHERWISE BEGIN END;
00930                                        END; (* CASE NC2 *)
00940                                    END ELSE BEGIN
00950                                        IF NOT KANJI THEN BEGIN
00960                                            JBUF:=JBUF !! KI;
00970                                            KANJI:=TRUE;
00980                                        END;
00990                                        JBUF:=JBUF !! STR(C1) !! STR(C2);
01000                                    END;
01010                                END;
01020
01030                                'A3'X:BEGIN
01040                                    IF HANKAK THEN BEGIN
01050                                        NC2:=ORD(C2);
01060                                        CASE NC2 OF
01070                                            'C1'X..'DA'X:
01080                                                JBUF:=JBUF !! STR(UPPER(.NC2-'C0'X.));
01090                                            'E1'X..'FA'X:
01100                                                JBUF:=JBUF !! STR(LOWER(.NC2-'E0'X.));
01110                                            'B0'X..'B9'X:
01120                                                JBUF:=JBUF !! STR(NUMBER(.NC2-'AF'X.));
01130                                            OTHERWISE BEGIN END;
01140                                        END; (* CASE NC2 *)
01150                                    END ELSE BEGIN
01160                                        IF NOT KANJI THEN BEGIN
01170                                            JBUF:=JBUF !! KI;
01180                                            KANJI:=TRUE;
01190                                        END;
01200                                        JBUF:=JBUF !! STR(C1) !! STR(C2);
01210                                    END; (* IF *)
01220                                END;
01230
01240                                'A4'X..'F3'X:
01250                                    IF NOT SEIGYO THEN BEGIN
01260                                        IF NOT HANKAK THEN BEGIN
01270                                            IF NOT KANJI THEN BEGIN

```

```

01280                                KANJI:=TRUE;
01290                                JBUF:=JBUF || KI;
01300                                END;
01310                                JBUF:=JBUF || STR(C1) || STR(C2);
01320                                END;
01330                                END ELSE BEGIN
01340                                CASE ORD(C1) OF
01350                                'C6'X:CASE ORD(C2) OF
01360                                'FC'X:BEGIN
01370                                    WABUN:=TRUE;
01380                                    EIBUN:=FALSE;
01390                                    END;
01400                                    OTHERWISE BEGIN END;
01410                                END>(* CASE C2*)
01420                                'B1'X:CASE ORD(C2) OF
01430                                'D1'X:BEGIN
01440                                    EIBUN:=TRUE;
01450                                    WABUN:=FALSE;
01460                                    END;
01470                                    OTHERWISE BEGIN END;
01480                                END>(* CASE C2 *)
01490                                OTHERWISE BEGIN END;
01500                                END>(* CASE C1 *);
01510                                END; (* IF NOT SEIGYO *)
01520                                OTHERWISE BEGIN END;
01530                                END; (* CASE C1 *)
01540                                END; (* FOR *)
01550
01560                                WRITELN(JBUF);
01570                                WRITELN(FO,JBUF);
01580                                IF EIBUN THEN IF TAIYAKU THEN WRITE(FOE,JBUF)
01590                                    ELSE WRITELN(FOE,JBUF);
01600                                END; (* IF LENGTH *)
01610                                END; (* IF C=CHR('15'X) *)
01620                                END; (* WHILE DO BEGIN *)
01630
01640                                CLOSE(FI);
01650                                CLOSE(FO);
01660                                CLOSE(FOE);
01670                                END.

```

図4. ODM文書から訳文を取り出すプログラム ODMJOIS

使用するDD名

- ・ ODMOUT : 取り出されるODM文書のデータセットを割り当てる。
(ODM. TEXT (ODMXXXXX))
- ・ JOISOUT : 対訳文書を書き出すデータセットを割り当てる。
(JOUT. TEXT)
- ・ ANKOUT : 英文を書き出すデータセットを割り当てる。
(EOUT. TEXT)

プログラムの説明

手続き INITIALIZE (行番号250-380) は変換のテーブルと変数の初期化を行なう。行番号480で読み込んだレコードが日本語情報部であるかをチェックし、日本語情報部については、変数BUFにテキストが読み込まれている。変数BUFを2バイトごとに取り出し、その第一バイトの値によって場合分けを行い、次のように処理をする。

- ・ 30 : 第2バイトによる制御コードを分類する。 (560-700)

- ・ 4 0 : 空白の処理 (7 2 0 - 7 5 0)
- ・ 7 6 : 対訳はじめ, 対訳おわりのチェック (7 7 0 - 8 1 0)
- ・ A 1 : 英文中の記号(全角)を半角に変換する. (8 2 0 - 1 0 1 0)
- ・ A 3 : 英文中の英大文字, 英小文字, 英数字を半角に変換する. (1 0 3 0 - 1 2 2 0)
- ・ A 4 - F 3 : 日本語の文字列にシフトコードを付ける. (1 2 4 0 - 1 5 1 0)

対訳文(JOUT.TEXT)に取り出す英文はレコード単位とし, 英文のみ(EOUT.TEXT)に取り出す英文は文単位とした. これらのデータセットには行番号を付けなかったので, LISTコマンドで見るときは次のようにNONUMを指定する.

```
LIST JOUT.TEXT NONUM
```

使用方法

実行形式をATLAS2.LOAD(ODMJ0IS)に作り, 以下のJOB文で実行する.

この例ではODM文書としてODM.TEXT(ODM00001)を指定している.

```
00840 //*****
00850 //*      ODM --> JOUT & EOUT      *
00860 //*****
00870 //ODMJ0IS EXEC PGM=ODMJ0IS,REGION=4096K
00880 //STEPLIB DD DSN=A79999A.ATLAS2.LOAD,DISP=SHR
00890 //ODMOUT DD DSN=A79999A.ODM.TEXT(ODM00001),DISP=SHR
00900 //JOISOUT DD DSN=A79999A.JOUT.TEXT,DISP=SHR
00910 //ANKOUT DD DSN=A79999A.EOUT.TEXT,DISP=SHR
00920 //OUTPUT DD DUMMY
00930 //SYSPRINT DD SYSOUT=*
00940 //
```

図5. ODMJ0ISを実行するジョブ制御文

5. プログラムの入手方法

ここに掲載したプログラムおよび入出力用のデータセットを筆者のデータセット(課題番号A71723A)に本年度中は格納しています. 興味のある方は自由にコピーしてご使用下さい.

公開されるデータセット名

- ・ ATLAS 2 . PASCAL : JOISODMとODMJ0ISのソースプログラム.
- ・ ATLAS 2 . LOAD : JOISODMとODMJ0ISの実行形式ファイル.
- ・ ATLAS 2 . CNTL : 3つの処理を1つのJOB文にまとめたファイル.
- ・ ODM . TEXT : ODM文書ライブラリ.
- ・ ODMHEAD . TEXT : 文書管理情報部を取り出すためのデータセット.
文書名「翻訳結果」を持つ.
- ・ ATLASDOC . TEXT : 最新の情報およびバグ情報を掲載した日本語テキストファイル.

JOIS.TEXT, JOUT.TEXTおよびEOUT.TEXTは各自がEDITコマンドで作成してください. (JOUT.TEXTとEOUT.TEXTは空でも良いが, 既存でなければならない.)

コピーの仕方は次のとおりです.

```
COPY 'A71723A.ATLAS2.PASCAL' ATLAS2.PASCAL
```

また、ATLAS 2. CNTL中の課題番号はA79999Aとなっていますので、コピー後にEDITモードで次のように各自の課題番号 (USERID) に変更して下さい。

```
CHANGE 10 999 /A79999A/USERID/ ALL
```

パスワードも正しく変更して下さい。

文書番号を変更したい時は、ジョブ文の全ての番号を変更しなければなりません。文書番号を00001から00101に変更するには次のようにして下さい。

```
CHANGE 10 999 /00001/00101/ ALL
```

以上の手続きが終了した後、JOIS. TEXTに原文を入力して

```
SUBMIT ATLAS2
```

を投入して下さい。結果はEOUT. TEXTとJOUT. TEXTに出力されているはずですが、STコマンドでジョブの終了を確認してから、LISTコマンドで出力して下さい。(NONUMの指定をお忘れなく)

6. 終わりに

ここに発表したプログラムは必要最低限のことを実行するものにとどめたが、本稿の解説である程度ODMの文書構造が理解されたと思う。必要な方はその他の制御コードをも取扱い、もっときめ細かな処理プログラムを書いて下さい。

近々、日本語PFのTTYTYPEがサポートされると聞きましたが、その際は是非ODMのTTYTYPEもサポートしていただきたい。

最後に、センターのライブラリ室の森田洋子さんには内部資料の貸与や、富士通への問い合わせなどでお世話になった。感謝いたします。

参考文献

- [1] 平良 豊：「PC9801用日本語対応TSSグラフィック端末プログラム」 九大大型計算機センター広報 Vol. 20 No. 3 1987, p. 183
- [2] ATLAS II 使用手引書
- [3] 森田洋子, 平野広幸, 菅崎直弘, 柳池 定, 垣内良子, 矢次恵美子, 武富 敬：「自動翻訳システムATLASの使用について」 九大大型計算機センター広報 Vol. 20 No. 3 1987, p. 143
- [4] 九大大型計算機センターニュース No. 317 (1985.8.2)