

## 英日機械翻訳システムSHALT における英日トランスファー

堤 泰治郎 西嶋 智恵子 笈 義郎

日本アイ・ビー・エム株式会社 サイエンス・インスティテュート

### 1. はじめに

機械翻訳では、対象とする文書分野を限定することによって構文および意味的に処理すべき範囲が狭くなり品質の良い翻訳が期待できる。また、翻訳の言語対と方向を固定することにより効率の良い機械翻訳システムが比較的容易に実現できると考えられる。

本稿では現在、日本アイ・ビー・エム株式会社、サイエンス・インスティテュートで研究を行っている英日機械翻訳システムSHALT (A System for Human-assisted Language Translation) [1, 2, 3, 4] の概要と英日トランスファーについて述べる。

### 2. SHALTの概要

SHALTは(1)英語から日本語、(2)IBM計算機マニュアル対象という条件のもとで、(a)日本語らしい翻訳、(b)比較的小型の用語辞書、(c)英日トランスファー部分の維持・管理の容易性を目標としている。図1に英文解析、英日トランスファー、日本語生成から成るトランスファー方式にもとづくSHALTの翻訳プロセスを示す。

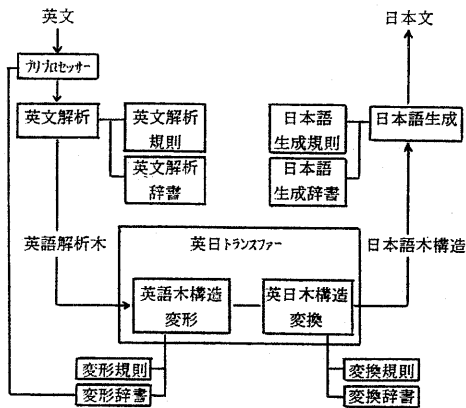


図1 翻訳のプロセス

翻訳者による技術翻訳の場合、特に動詞を中心とした表現や慣用句などについて英日翻訳の定石や翻訳者の知識・経験が活用されて、最終的に読みやすい日本語が得られていると考えられる。その翻訳時の主要な手掛りの一つは、原文の統語構造と特定の動詞や名詞、さらにその分野における意味であるとされる。SHALTでは、このような情報を最も直接的に記述できて、しか

も維持・管理が容易な方法としてトランスファー方式を採用し、さらに英語が日本語よりも統語的に制約が強いことを考慮して統語解析により得られる英語解析木を中間表現としている。

図1において、まず、入力された英文は英文解析辞書を参照しながら英文解析規則によって統語的に解析が行なわれる。この出力は一つまたは二つ以上の英文解析木である。次の英日トランスファー部分は、英語木構造変換と英日木構造変換の二つの独立した処理に分かれている。ここでは、解析木レベルのトランスファー方式で一般に複雑になりがちなトランスファー処理の部分の二つに分けることによって、トランスファー全体の処理の流れを明確にし、その部分の維持・管理を容易にしている。その結果、より多くのトランスファー規則および辞書の情報を蓄えることによって、翻訳文の品質を向上させることができる。

英語木構造変換部分では、入力された英語解析木を日本語に変換しやすい英語木構造へ変換する。一方、英日木構造変換部分では、英語の統語構造と名詞の意味マーカーを手掛りにして英語木構造を対応する日本語木構造へ変換する。この処理では、基本的な日本語構文、動詞や名詞の訳語選択を行ない、さらに英語木構造での統語的あいまい性を解消した一つの日本語木構造を出力する。

最後の日本語生成では、主として、助動詞の生成や格助詞あるいは格助詞相当語句の変換などを行うが、翻訳文の品質を高めるために日本語木構造の構造変換処理も行えるようになっている。SHALTでの日本語中間表現、すなわち日本語木構造はこれらの処理が容易に行えるような構造になっている。

次に、英日トランスファー処理の説明の都合上、英文解析について簡単に述べておく。

### 3. 英文解析

英文解析は、IBMワトソン研究所で開発されたパーサー、英語解析辞書、および英文解析規則[5]を使ってボトムアップ、パラレルにおこなわれる。主な特徴としては、(1)統語情報による解析 (2)文として最終的に解析できない場合に尤もらしい構造を出力することなどが挙げられる。(1)の特徴により、アルゴリズムの単純化、処理の高速化が可能であり、また意味情報を使わないという意味で分野依存性が少なくなる。(2)の特徴により、翻訳作業の後編集時に部分的に成功した翻訳結果を有効利用することができる。図2に英文解析の結果の例を示す。

```

(DECL (NP (NOUN* "Messages" ("message" PL))
  (PTPRICL(VERB* "received" ("receive" (ED EN)))
    (PP (PREP "from")
      (AJP (ADJ* "remote" ("remote" BS)))
      (NOUN* "terminal" ("terminal" SG))))))
(VERB "are" ("be" PS))
(VERB* "processed" ("process" (ED EN)))
(PP (PREP "by")
  (DET (ADJ* "an" ("a" BS)))
  (NP (NOUN* "application" ("application" SG)))
  (NOUN* "program" ("program" SG))
  (PRPRICL(VERB* "running" ("run" FG))
    (PP (PREP "in")
      (DET (ADJ* "the" ("the" BS)))
      (AJP (ADJ* "central" ("central" BS)))
      (NOUN* "processor" ("processor" SG))))))
(PUNC ".")

```

図2 英文解析の結果の例

図2は、多品詞語や係り受けによる統語的なあいまい性がない場合であり、ただ一つの解析木がLISP形式で得られている。図2において、DECL、NP、VERB\*などは、文法カテゴリー名や品詞名を表している、“Messages”、“received”などは、末端記号、すなわち英単語を表している。また、各単語の形態素情報は、(カテゴリー名 “出現形” (“原形” 形態変化情報))の形式で示されていて、たとえば、(NOUN\* “Messages” (“message” PL))の場合、カテゴリー名がNOUNで、その原形が“message”また、形態変化が複数形という意味である。表1に、形態変化情報を表すのに使用する記号を示す。

カテゴリー名	記号
NOUN	SG PL
VERB	PS ED EN
ADJ	BS ER EST
ADV	BS ER EST

表1 形態変化を表す記号

表1において、SG、PL、PS、ED、EN、BS、ER、およびESTは、それぞれ、単数形、複数形、現在形、過去形、過去分詞形、形容詞および副詞の基本形、比較級の形式、および最上級の形式を表す。

多品詞語によるあいまい性がある場合で統語的に可能な複数個の解析結果が得られる場合には、各解析木にそれぞれの木の尤もらしさを示す値が同時に出力されるようになっていて[6]、英日トランスファーの段階で意味的に排除されなかったもののうち、この値が最小のものを日本語中間表現として選択するようになってきている。

前置詞句についての係り受けについては現在、前置詞がofの場合にかぎり直前の名詞に係るようにし、その他の前置詞の場合には動詞に係るようにしている。一般に、これらの係り受けに関するあいまい性は、深い意味処理を行わないと解決できない問題である。SHALTの英文解析の部分は、現在、効率と維持・管理の観点か

ら統語情報のみによる処理を行っており、それ以降のステップでの意味処理のためにとりあえず必要な材料を提供するという方針をとっている。現在、英日トランスファーの部分で、係り受けの他の可能性も調べ、IBM計算機マニュアルの分野で最も確からしいものを選択するような機能を研究・開発中である。

#### 4. 英日トランスファー

英日トランスファー部分は、英語木構造の変形と英語木構造から日本語木構造への変換の二つのステップから成る。最初の変形の部分では、あらかじめ英語木構造を日本語木構造へ変換しやすい形、すなわち日本語構造により近い形に構造変形する。

本来、英日翻訳の場合、原文の表現に最もよく対応する日本語の表現を作り出すことを目指しているようであるが、技術翻訳の場合には、原文が複雑で高級な言回しであるときに、それを単純で判りやすい表現にかえる方が望ましい場合が多いようである。

SHALTでは、異なる英語の文体であっても可能なかぎり単純な同じ文体の日本語に翻訳することを特徴の一つとしている。ここでは、英日トランスファー過程を二つのステップに分け、二段階トランスファー方式としている。すなわち、英語木構造変形では、いずれ、同じ日本語木構造となる英語木構造をまとめる処理を行ない、また、英日木構造変換では、ほぼ1対1に対応した日本語木構造へ変換する処理を行う。図3に、この処理の概念図を示す。

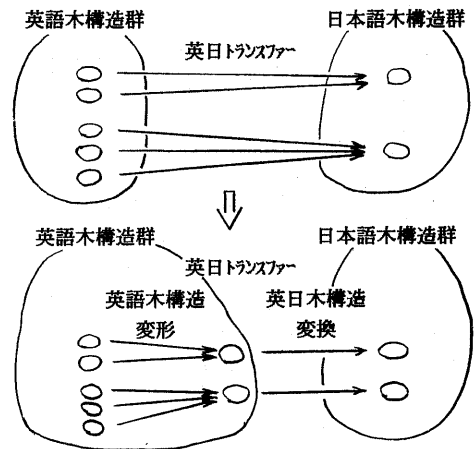


図3 二段階トランスファー方式の概念図

この方式によれば、複雑になりがちなトランスファー部分を二つに分けることにより、処理の分担および明確

化を行ない、変形・変換規則や辞書の維持・管理および精密化を容易にする。従って、技術文翻訳の定石や英語の慣用表現を適切に翻訳するための規則を具体化しやすくになり、翻訳文の品質が向上し、また簡明な日本語を生成できる。

#### 4.1 英語木構造変形

英語木構造変形は、変形規則と変形辞書を使って行なわれる。変形処理の主目的はすでに述べたとおりであるが、英文解析結果とのインタフェースとしての役割や、英文から時制、アスペクト、モーダル情報を抽出し、その英文を単純化するというような機能も持っている。変形規則は、マッチング・パターンとターゲット・パターンの組で構成されていて、そこでは、入力木がマッチング・パターンに合致した場合に、どのような木構造を生成するかをターゲット・パターンとして記述する。マッチング・パターンは、次のような形式で表される。

(部分木記述部 - マッチング条件記述部<sup>\*</sup>)<sup>\*</sup>  
 ここで、  
 部分木記述部 : (n<sup>\*</sup>) または、  
                   マッチング変数  
 マッチング条件記述部 :  
                   マッチング変数 または、  
                   (マッチング変数 マッチング条件<sup>\*</sup>)  
 マッチング条件 : (述語関数名 引数<sup>\*</sup>)

また、\*は、一つ以上の要素の並びを表す。部分木記述部では、二つの方法で入力木の任意の部分木を指定することができる。たとえば、(1 3 2)は、入力木の第一番目の子ノードを根ノードとする部分木の第三番目の子ノードの部分木、さらに、第二番目の子ノードの部分木を指定する。これは、絶対的な指定の方法である。単に0を指定した場合には、入力木全体を対象とすることを示す。また、後で説明するマッチング変数を指定することによって、そこにすでに代入されている部分木をマッチングのチェックの対象とすることもできる。

マッチング条件記述部は、部分木記述部で記述された部分木の一段下位の部分木の並びについてマッチングのチェックを行う部分であり、記述された条件に合致した場合に指定されたマッチング変数にその部分木が代入される。また、マッチング条件を指定せずにマッチング変数だけを指定することもでき、任意の部分木または、空が代入される。

マッチング条件部では、各種の述語関数とそれらの引数を指定して、マッチングの具体的な条件を記述する。述語関数としては、その部分木の根ノードの文法カテゴリー名を調べたり、末端記号、すなわち英単語を調べたり、また、その根ノードに付加されている属性情報を調べたりするものなどが用意されている。マッチングの条件としては、述語関数の引数のリストが、OR条件であ

る以外は、すべてAND条件である。

一方、ターゲット・パターンでは、すでに部分木または空が代入されているマッチング変数や追加したい木構造の要素、さらに新に作る木構造形式を指定する。

図4に、変形規則とその適用例を示す。図4(a)は、英文解析部の出力、すなわち英語木構造変形部の入力を、図4(b)は、適用する変形規則を、図4(c)は、適用の結果を示す。

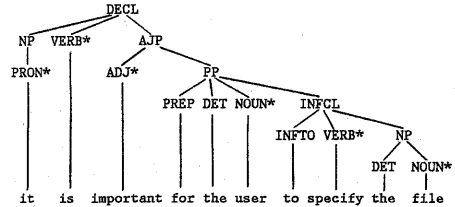


図4 (a) 入力英語木構造

A  
 <0 - (\*1 (\*2 (T "it") \*3 (\*4 (T "is") \*5 (\*6 (P AJP) \*7))  
 (\*6 - (\*8 (\*9 (P ADJ\*) \*10 (P PP) \*11))  
 (\*10 - (\*12 (\*13 (P PREP)(T "for") \*14 (\*15 (P INFCL) \*16))  
 (\*15 - (\*17 (\*18 (P INFTO) \*19))>  
 →  
 B  
 <\*1 (VP (COMPL "that") ((ISN NP) \*14) \*19) \*3 \*5  
 ((ISN VERB\*)(IAPP-TERM "BJ-" \*9)) \*7>

図4 (b) 変形規則

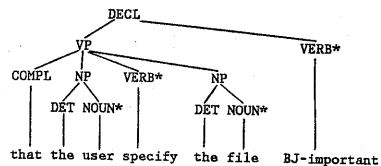


図4 (c) 出力英語木構造

図4(b)において、A部はマッチング・パターン、B部はターゲット・パターンを示す。\*1、\*2、\*6、\*10、\*15、\*19は、マッチング変数を表す。A部の記述では、(0 - (\*1...))、(\*6 - (\*8...))、(\*10 - (\*12...)) および (\*15 - (\*17...)) という四つの (部分木記述部-マッチング条件記述部) の形式が指定されている。最初の行は、入力英語木に対する処理を表し、二番目の行は、最初の行での処理で、マッチング変数\*6に代入された部分木に対する処理を表す。\*6、\*10、\*15のように、部分木記述として指定されるマッチング変数は、それ以前の任意の個所で部分木が代入されるものであればよい。このような記述方式は、特に、複雑な木構造に対するマッチング・パターンを書きやすく、また見やすくする。

A部でのマッチング条件 (T "it") は、その部分木の末端記号、すなわち英単語が "it" であるという条件を表し、(P AJP) は、その部分木の根ノードの文法カテゴリー名が AJP (形容詞句) であるという条件を表している。ここで、TおよびPは、それぞれ、末端記号を調べたり、文法カテゴリー名を調べたりする述語関数名である。

B部では、マッチング変数 \*1、\*2、・・・、\*19 に代入された部分木を使って出力木構造を作ることを指定するものである。ここで記述されている ("や") は、基本的にそのまま構造の指定に使われ、マッチング変数は、その代入値が使用される。また、別の構造要素を作るための処理プログラムの名前の指定も行われ、そのプログラム名の直前に ! 記号を付けて区別する。B部の (!SN NP) では、SN が処理プログラム名で NP がその引数であり、出力として、NPXXX(XXXはユニークな三桁の数字) という記号を得る。また、(!APP-TERM "BJ-" \*9) では、"BJ-" と \*9 の代入値を結合して新しい記号を作る処理を行う。

図4 (c) にこの規則の適用結果を示すように、結局、It is important for the user to specify the file. という英文が、  
That the user specify the file is BJ-important.  
という形式に変形されたことを示している。ここで、BJ-important は、この important が叙述的用法の形容詞であるということを明示的に表したものである。

規則の実際のインプリメンテーションでは、マッチングの効率を上げるために、マッチング部分の条件の概略を記述したインデックスを設けている。インデックスの記述例を次に示す。

<(VERB\* (PREP ADV ADV\*) INFCL)  
("is" "be" "are") "about")>

ここで、リストの第一要素は、入力木に対して第一レベルのカテゴリー名の種類と並びを調べることを指定し、第二要素は、末端の英単語とその並びを調べることを指定する。図4 (b) のA部の場合には、インデックスは、

<(ADJ) ("it" "is")>

となる。A部の具体的なチェックに進む前にこのインデックスを見て、入力木構造にこの規則がマッチするかどうかを大まかに調べる。

変形辞書は、変形処理にともなって必要となる品詞変換処理のためのデータを蓄えており、表2に例を示すように、これには現在、名詞-動詞変換、形容詞-副詞変換などのためのものがある。

名詞	動詞	形容詞	副詞
addition	add	particular	particularly
use	use	primary	primarily
deletion	delete	careful	carefully
destruction	destroy	continuous	continuously
execution	execute	efficient	efficiently

表2 変形辞書の内容の例

表2のデータの利用例を次に示す。ここでは、名詞 "deletion" が動詞 "delete" に変換される。

The program allows deletion of the records.  
The program allows deleting the records.

変形規則は、修正を容易にし、効率を上げるために適用する部分木の種類ごとにグループ化されている。たとえば、文レベル、名詞句レベル、また前置詞句レベルの部分木に適用される変形規則のグループなどがある。

まず、文レベルの変形処理の若干の例を次に示す。

例1:

It is required that you specify the assignment.  
That you specify the assignment is required.

例2:

There are several records in the file.  
Several records exist in the file.

例3:

System operation is so impaired that the IPL procedure has to be repeated.  
Because system operation is very impaired, the IPL procedure has to be repeated.

例4:

The routine has a relatively low usage rate.  
Usage rate of the routine is relatively low.

上記の例で、例1および例2は、英語での It-that 構文、および There be-構文をより日本語へ変換しやすい形に変形する例である。また、例3は、言い換えを行った例で、例4は、日本語的発想の表現に近づけるための変形の例である。

次に、名詞句レベルの変形処理のいくつかの例を示す。

例1:

execution of the program  
executing the program

例2:

a disk available with ~  
a disk which is available with ~

例3:

one type of ~  
one-type-of ~

上記の例で、例1は、英文にしばしば出現する"動作名詞+of+名詞"表現を直訳的に「そのプログラムの実行」と訳すのではなく、「そのプログラムを実行する」のように日本語としてより自然な叙述的表現を得るための変形例である。例2は、英文での簡略表現を基本的な表現へもどし、それ以後の処理をできるだけ共通化する例である。例3は、"one type of"の表現を"one-type-of"という形容詞相当語句に変える変形の例である。

最後に、前置詞句レベルの変形の例を示す。

例1:

in turn ~  
in-turn ~

例2:

in the field of ~  
in ~

例3:

for the purpose of ~  
for-the-purpose-of ~

例1は、前置詞句を副詞句に、例2は、別の前置詞に、また例3は、前置詞相当語句に変形する例である。

以上述べた例のうち、名詞句レベルの変形の例3や前置詞句レベルの変形の例1~3のようなかなりジッドな慣用表現で、統語的あいまい性がほとんどない場合には、英文解析に先立って、プリプロセッサで、あらかじめ処理することもできる。

以上のように、変形規則は、文レベル用、名詞句レベル用、前置詞句レベル用などのようにグループ分けされているが、各グループは、さらに複数のサブ・グループに分かれている。たとえば、文レベルの変形規則グループは、広域構造変形規則、to不定詞処理規則など現在、19のサブ・グループに分かれている。

図5に、これらの変形規則群の適用順序を示す。

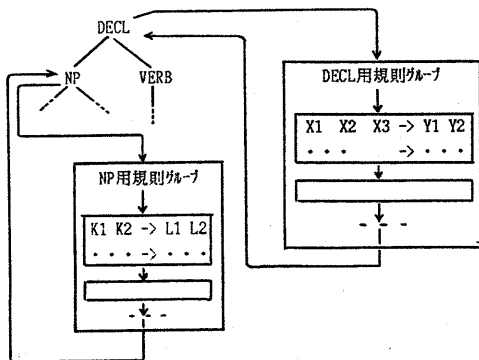


図5 変形規則群の適用順序

変形規則の適用は、入力木に対してトップ・ダウンに

行なわれる。各部分木において、その根ノードの文法カテゴリーを調べて、それぞれに専用に用意された変形規則グループの適用が行なわれる。各変形規則グループのサブ・グループは、図5に示すように、あらかじめ決められた順番に従って適用されていく。各サブ・グループのなかで一つの規則がマッチして入力木が変形を受けた場合は、そのサブ・グループの残りの規則については調べられずに次のサブ・グループの適用へと進む。このようにして、すべてのサブ・グループの適用が一度終了すると、その文法カテゴリー別の規則グループについての処理を終了する。次に、入力木構造の下位のレベルの部分木に対して同様に規則グループの適用が行なわれる。最後に、入力木の末端ノードまで処理が終了すると変形処理全体の処理が終わる。

トップダウンの規則の適用では、入力木に対して広い範囲から狭い範囲の方向へ処理していくので、後で使うかもしれないデータを保持したりすることもほとんどなく、一度の処理パスで効率的に変形処理が行える。

4.2 英日本語構造変換

すでに変形処理をうけた英語木構造を変換辞書と変換規則または変換ルーチンを使って、対応する日本語中間表現(日本語木構造)へ変換する。主な機能は、(1)英文での係り受けのあいまいさを解決する(2)訳語を選択する(3)日本語構文を選択することである。これらの処理のために、計算機マニュアル分野専用の名詞の意味マーカーを設定し、それらを使って単純で効果的な英日変換辞書を記述している。

(1) 計算機マニュアル分野での名詞の意味マーカー

文や名詞句の訳し分けという観点から現在24種類の意味マーカーを設定している。表3に意味マーカーの一覧を示す。

意味マーカー	説明	意味マーカー	説明
LC	Logical Container	WK	Work/Action
LE	Logical Entry	PS	Predicate
LP	Logical Path	AP	Attr. of PS
DM	Document	SL	Supply
ST	State	PT	Part
TH	Technique/Theory	DT	DM Term
FA	Feature/Ability	ML	Material
IF	Information	TM	Time
AT	Attribute	PL	Place
VA	Value of AT	PN	Name
HM	Human	PO	Point
UD	Unit/Device	OG	Organization

表3 名詞の意味マーカー

計算機マニュアル文に出現する名詞はすべて一つあるいはそれ以上の意味マーカーをもち、これらは、名詞変換辞書に記述される。たとえば、programはLE、fileはLEとLC、operatorはHMとLEという意味マーカーをもつ。名詞の意味マーカーの体系には各種のものが提案されているが、単純すぎると効果が小さく、逆に複雑で詳細すぎると維持・管理が大変になると思われる。ここでの分類は、計算機マニュアル分野では現在のところ妥当なものであると考えている。

(2) 英日変換辞書

英日変換辞書では、英文の統語構造と日本語名詞自身や主名詞の意味マーカーとの組合せによる英日変換の条件と目標とする日本語構造を記述しており、動詞、名詞、形容詞、前置詞などについて現在16種類の辞書が用意されている。図6に動詞"provide"についての変換辞書の記述内容を簡略化して示す。

```

("provide"
A { ((SB (S ((LE UD) Y1 "が")))
      (DO (S ((FA AT) Y1 "を")))
      (P "備え" PY1 (V SHIMO1 NIL JYOOTAI TRANS)))
B { ((SB (S ((DM LE UD) Y1 "が")))
      (DO (S ((HM) Y1 "に")))
      ("with" (S ((IF FA AT) Y1 "を")))
      (P "提供" PY1 (V SAHEN NIL KEIZOKU TRANS))) )

```

図6 "provide"についての記述例

図6において、SBは、英文の主語に対して、またDOは、第一目的語に対してこれらの記述を適用することを表す。また、ここでは示されていないが、IOという記号で、英文の第二目的語に対して適用することを指定する。SBで始まるリストでSは、意味マーカーによる条件を表し、(SB (S ((LE UD) Y1 "が")))では、英文の主語の意味マーカーがLEまたはUDであれば、それに対応する日本語名詞句には、"が"という格助詞を使うことを指示している。ここでのSによる意味マーカーの指定の前に、Hという記号を使って日本語の主名詞自身を条件として記述することもできる。これは、意味マーカーによる条件があらずして、特定の日本語主名詞を直接に記述したい場合に有効である。

SB、DO、IOの記号の位置に直接、前置詞が記述された場合には、英文でこの位置にその前置詞がくることを表し、さらにその中の名詞句について、SB、DO、IOの場合と同様な条件およびその前置詞の訳、すなわち日本語名詞句の格助詞を指定する。また、Pの記号で始まるリストは、それ以前に順序だてて指定された条件がすべて満足されたときに、その英語動詞を日本語でど

のように訳すかを指定するための記述である。具体的には、その日本語動詞の語基と、活用形、性質などの付属情報を与える。

また、SB、DO、IOなどの記号の位置に、PPという記号を使って、オプションであるがその動詞とともに頻繁に使用される前置詞句のリストを指定することもできる。たとえば、

```

(PP
  ("in" (S ((TH DM) Y1 "で")))
  ("with" (S ((LE VA) Y1 "を使って"))))

```

などになる。

図6において、Y1、PY1は、作成する日本語中間表現の部分木の構造のタイプを指定する記号である。

次に、図6の記述内容による英日変換について具体的に説明する。まず、図6のA部では、もしも主語の意味マーカーがLEまたはUDで、第一目的語の意味マーカーがFAまたはATであれば、日本語の最初および二番目の名詞句の格助詞としてそれぞれ、"が"、"を"を選択し、動詞の訳として"備え"を選ぶことを表している。一方、図6のB部の記述では、SBとDOのリスト部分はA部と同じような指定であるが、これにくわえて、"with"の前置詞句が第一目的語の直後にきて、その名詞の意味マーカーがIF、FA、またはATのときに、その前置詞の訳として"を"を選ぶことを指定している。またこの場合、動詞の訳として"提供"を選ぶことを表している。図7(a)および図7(b)に、図6の記述内容を図式で示す。ここで、図6のA部は図7(a)に、B部は図7(b)に、それぞれ対応する。

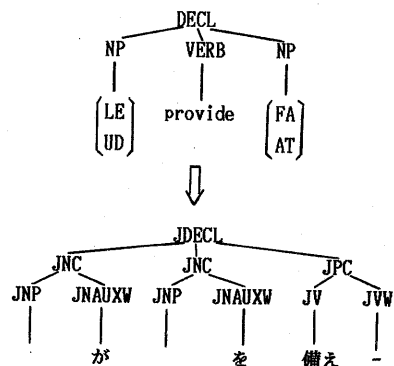


図7 (a) A部の記述内容

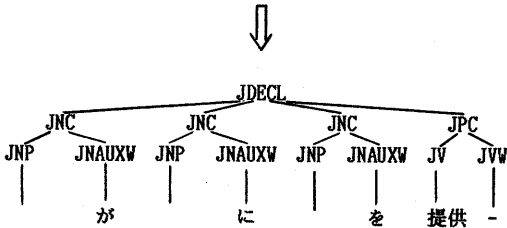
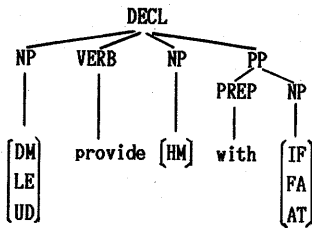


図7 (b) B部の記述内容

このように、英語の構文と主名詞の意味マーカとの組合せによって動詞、名詞、前置詞の訳語選択と日本語での名詞句の順序、格助詞などの構文選択が行われる。また、辞書記述のいずれに対しても英語木構造が満足されない場合で、他に受理できる英語木構造が存在する場合には、計算機マニュアル分野で適切ではないとしてこの木を捨てることもできる。

図8に、“with”についての前置詞変換辞書のうちNP-PP REP-NPのタイプの場合について、記述内容を単純化した例を示す。

(“with”  
 (S ((LE) (FA) “を備えた”)  
 ((LE) (LC) (AT) “の”)  
 ((WK) (LE) “による” “を使った”)  
 (\* \* “をもつ”))

図8 “with”についての前置詞変換辞書の記述例

図8において、Sは意味マーカによる条件を表す。たとえば、((LE) (FA) “を備えた”)では、第一番目および第二番目のNPの意味マーカが、それぞれLEおよびFAであれば、“with”の訳として“を備えた”を選択することを指定している。最終行の(\* \* “をもつ”)では、これよりも前に指定してある記述がすべて満足されない場合にデフォルトとして“をもつ”という訳にすることを表している。次に、この変換辞書を使った訳し分けの例を示す。

ACF/VTAM with the multisystem networking feature

[LE] [FA]

マルチシステム・ネットワーク機能 を備えた ACF/VTAM

Jobs with mixed characteristics

[LE] [AT]

いろいろな種類の ジョブ

また、図9に、名詞変換辞書の記述例を示す。

(program (J (LE “プログラム”)))

(operator (J (HM “操作員”) (LE “演算子” “作用素”)  
(K “オペレーター”))

(function (J (FA “機能”) (LE “関数”)))

図9 名詞変換辞書の記述例

図9に示すように、各々の英語名詞に対して意味マーカごとに頻度順に訳語が並べられている。このような意味マーカについての多義性は、単純名詞句の範囲でのしほりこみや、すでに説明した動詞変換辞書を使ったしほりこみによって、最終的に解消されるものもある。

### (3) 英日変換処理

英日変換は、英語木構造変形により得られた英語木構造にそってボトムアップに、英日変換辞書を参照しながらこれを順々に対応する日本語木構造へ変換していく。この部分は現在、一部規則化され、残りはLISPで直接に処理ルーチンを作成している。この理由は、英日変換処理部分のうち英日変換辞書の追加・修正は頻繁に行なわれるが、変換処理部分自身はあまり頻繁に修正がないものと考えており、また、ここでの処理では、意味処理のために柔軟な記述が必要だと考えたからである。

図10 (a) に示す入力木を使って処理手順を説明する。まず、1、2、3、4の部分について、各単語に応じて英日変換辞書の検索を行なう。次に1、3、4の部分については単純名詞句の英日翻訳プログラム [2] によって日本語部分木を作成する。これらの結果を図10 (b) に簡略化して示す。

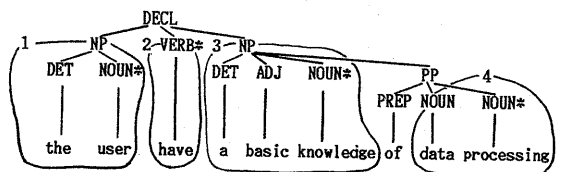


図10 (a) 入力英語木構造

- [5] G.Heidorn et al: The EPISTLE Text-Critiquing System, IBM Systems Journal 21 (3), 1982
- [6] G.Heidorn: Experience with an Easily Computed Metric for Ranking Alternative Parses, Proc. 20th Annual Meeting of the ACL, Toronto, Canada, 82-84, 1982

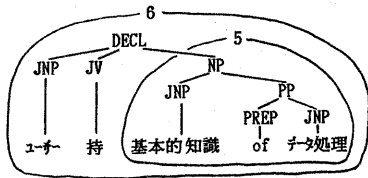


図10 (b) 変換の途中結果

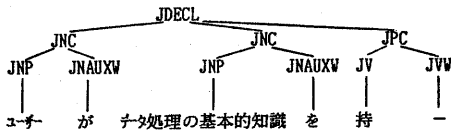


図10 (c) 日本語木構造

次に、図10 (b) の5の部分木について、NP-PREP-NPの変換辞書を検索して5に対応する日本語部分木を求める。最後に、6の木全体について、図6で説明したような形式の"have"についての動詞変換辞書データを使用してこの木構造を変換し、図10 (c) のような日本語木構造を得る。

このとき、英語木構造変形ですでに抽出され、DECLノードの属性として保持されている英文の時制・アスペクト・モーダルなどの情報も変換テーブルを使って英日変換され、図10 (c) のJDECLのノードの属性として保持される。この情報は、このあとの日本語生成の段階で日本語の助動詞などを生成するのに使用される。

## 5. まとめ

IBM計算機マニュアル文を対象とした英日機械翻訳システムSHALTの概要と英日トランスファーについて述べた。現在、社内での実験的使用へ向けて規則や辞書の追加・修正、および翻訳システムの改良・拡充を継続的に行っている。

## 6. 参考文献

- [1] 堤、英日機械翻訳システムSHALTの概要および英文解析と英日トランスファー、情報処理学会第31回大会予稿集1H-3, 1985
- [2] 堤、英日機械翻訳システムSHALTにおける単純名詞句の翻訳、情報処理学会第31回大会予稿集1H-4, 1985
- [3] 原田、英日機械翻訳システムSHALTにおける日本語生成、情報処理学会第31回大会予稿集1H-5, 1985
- [4] 西嶋、英日機械翻訳システムSHALTにおける翻訳文の評価、情報処理学会第31回大会予稿集1H-6, 1985