Knowledge Representation and Machine Translation

By Susumu Sawai, Masakatsu Sugimoto and Naoya Ukai

(Manuscript received September 26, 1981)

This paper describes a new knowledge representation called "frame knowledge represeentation-0" (FKR-0), and an experimental machine translation system named ATLAS/I which uses FKR-0.

The purpose of FKR-0 is to store information required for machine translation processing as flexibly as possible, and to make the translation system as expandable as possible.

1. Introduction

Preliminary research on machine translation (MT) started soon after computers became available. Early MT systems were only able to produce low-quality translations because of the speed and memory limitations of the machines. Translation programs were coded in low-level programming languages and, as a result, they could not be easily extended.

MT research was prevalent in the USA during the early 1960s. However, the conclusions of the ALPAC report published in 1966 opposed funding for MT research and resulted in the general discontinuance of MT research in the USA[1)].

The effort is more concerted in countries where MT systems are more necessary than in the USA. For example, the use of both French and English in

*Ca*nada and the multilingual use of formal documents in the EEC present pressing demands for practical MT systems.

The SYSTRAN system (produced by Latsec Incorporated) has been applied in the following areas:

1) Private companies in Canada use it for translating engineering documents from English into French.
2) NASA used it to communicate with the crews of the Apollo and Soyuz spaceships, translating between Russian and English.
3) The EEC uses it for examining the feasibility of other MT systems[2)]

Other systems currently being used include the METEO system which translates English weather reports into French in Canada, and the WEIDNER and LOGOS systems produced by private firms in the USA. Recently, there has been a revival of interest in MT systems in the USA, partly because of significant advances being made in artificial intelligence (A I) research.

The future development of MT systems is ensured by the total integration of high-performance computers, new man-machine interface designs, new software methodologies, and progress in knowledge engineering.[3)]

The language barrier in Japan is far greater than in the EEC or Canada, because Japanese is an isolated language. There is a large demand for document translation in Japan. For example, Japanese computer firms produce Japanese documents and manuals for export products which must be translated into English and other languages. The automobile, aircraft, and ship-building industries also have pressing need for MT systems.

2. Problems and solution

21.1 Methodological problems in a machine translation system

Basically, a machine translation system consists of three components: a dictionary (lexicon), grammar (translation rules), and the translation program (algorithm).

The major methodological problem in machine translation systems is how to separate the translation program from the grammatical rules. The advantage of this separation is that the program can be used for various languages and grammars without modification; that is. it is language-independent. However.

there are practical problems in separating the grammar from the program, including difficulties in formulating complex rules for linguistic data and avoiding large storage requirements or heavy computation loads[4)]

2.2 Solution by knowledge representation methodology

Artificial intelligence research on natural languages and knowledge representation progressed rapidly during the 1970s. In AI, "knowledge representation" is a combination of data structures and interpretive procedures that leads to '"knowledgeable" behavior.

A new type of of machine translation system conceived by Drs. Y. Wilks and R. Schank appeared in the early 1970s. This type of system translates input text into the knowledge representation of semantic primitives intended to be language-independent[1).]

At present, the major knowledge representation techniques are predicate logic, procedural representations, semantic networks, production systems (PS), and frames. In procedural representations, knowledge is contained in procedures (programs). The basic idea of production systems is a database consisting of rules, called production rules, in the form of condition-action pairs; i.e. "if this condition occurs, then do this action." A frame is a predefined internal relation. For example, a generic frame for a dog might have knowledge hooks, or slots, for facts that are typically known about dogs, like the BREED, OWNER, and NAME, and "attached procedures" for finding out who the owner is, if that is not known .

This paper proposes an efficient knowledge representation method using frame techniques to solve the above-described problems in machine translation systems.

3. FKR-0

Figure 1 shows the framework of frame knowledge representation-0 (FKR-0)[6)].

In the FKR-0 knowledge representation method, a production system is combined with a procedural representation and is systematized into a state

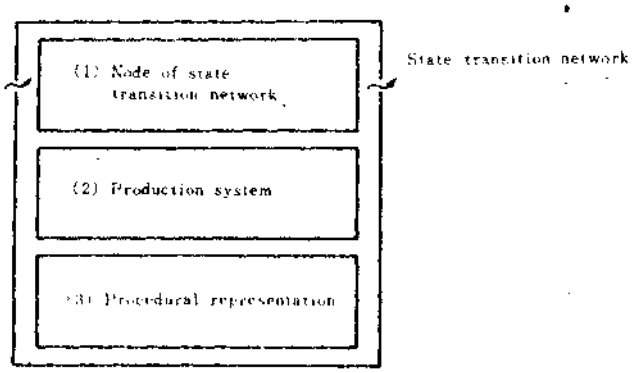

Fig. 1 - Framework of knowledge representation FKR-0.

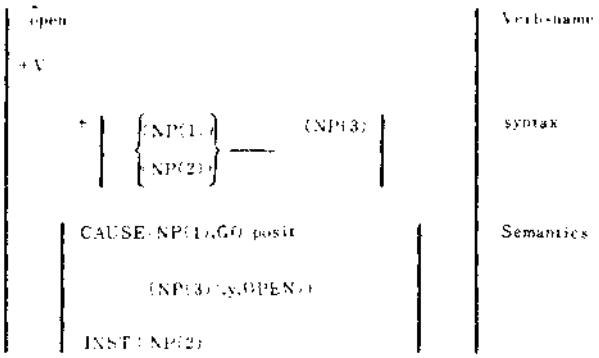

Fig. 2 -Jackendoff's semantic representation.

transition network. Rule representation frames and control frames are provided for the efficiency of system operation.

3.1 Rule representation frame

Figure 2 shows Jackendoff's semantic representation of verbs. Because Jackendoff is a linguist, he did not propose any machine translation system, but his semantic representation provides a good frame work with a clear indication of the relationship between the actor and the action. It indicates that the verb

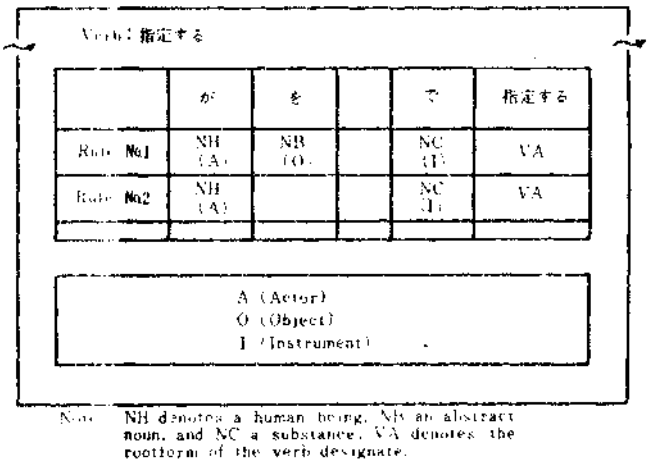

Fig. 3—Example of rule representation frame.

"OPEN" can take two noun phrases; that the subject can be cither of two noun phrases, NP (1) or NP (3); and that NP (2) is an instrument, "INST"[7].

Figure 3 is an example of the rule representation frame used in FKR-0 for the Japanese verb "指定する" (to specify). The frame shows:

1) the verb name, which is the name of a node in the state transition network;
2) the relationship between the verb and one or more noun phrases;
3) the conditional process to be performed after the rules are applied.

This conditional process includes the judgement of the conditions required for calling other frames.

Current FKR-0 specifications do riot have the "cause" concept included in Jackendoff's semantic representation, however, procedural representation is planned for future FKR-0 editions.

3.2 Control frame

The FKR-0 system has control frames which supervise the rule representation frames discussed in Sec. 3.1. Each pair of adjacent frames communicates by a control parameter as shown in Fig. 4.

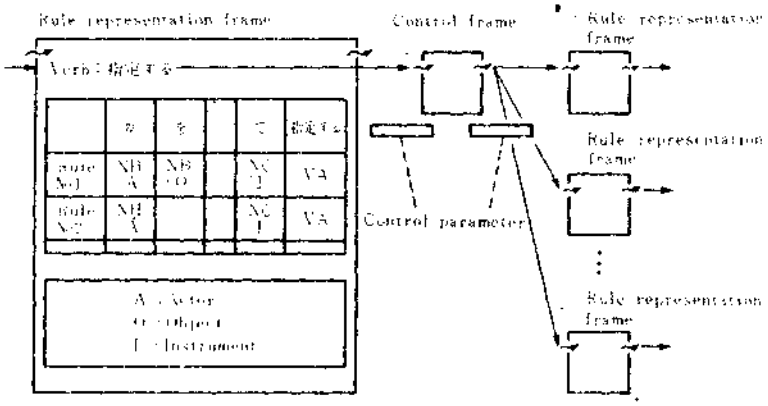

Fig. 4 - Communication of frames.

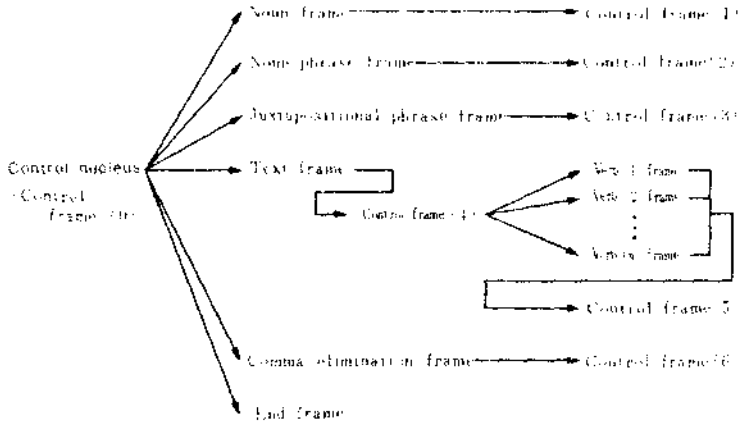

Fig. 5—Overview of frame structures of grammatical rules.

The roles of the control frame are to resolve one rule frame into several subrule frames and to control the calling sequence of these frames. This is one solution which overcomes the disadvantages inherent in production system (PS) methodology; i.e. inefficiency of program execution and opaqueness of the control flow. A control frame addresses the rule frame by means of the contents

of the control parameter. If the next frame is not specified, control returns to the top-level control frame called the "control nucleus".

Figure 5 is an overview of the frame structures of grammatical rules.

3.3 Grammatical rules

ATLAS/I is an experimental machine translation system in which grammatical rules are specified in FKR-0 representation. ATLAS/I currently has seven control frames and the following seven types of rule representation frames. Figure 6 shows the detailed frame structure of grammatical rules in FKR-0.

1) Noun frame

Example: ((STATE(NOUN)—-> STATE(CONTROL1)))
(R(COND(NHX4) H4) (PARM(0 W1)))

A noun phrase "H4" is formed by combining a noun denoting a human being "NH" (太郎) and postposition "X4" (が). The function "PARM" is the mapping function from graph to graph which can be used by the analysis, translation, and synthesis processes.

2) Noun phrase frame

Example: ((STATE(NOUN_OF_NOUN) —-> STATE(CONTROL2)))
(R(COND(NHX2NC) ND) (PARM(0 W321)))

A noun phrase "ND" is formed by combining a noun "NH" (太郎). the postposition "X2" (の), and a concrete noun "NC" (カギ).

3) Juxtapositional phrase frame

Example: ((STATE(NOUN_AND_NOUN) —-> STATE(CONTROL3)))
(R(COND(ND X9 ND)ND) (PARM(18 W1#3)))

A noun phrase "ND" is formed by combining a noun phrase "ND", the postposition "X9" (と), and a noun phrase "ND".

4) Test frame

Example: ((STATE(TEXT) —-> STATE(CONTROL4)))
(R(COND(SS CN SS)SS) (PARM(0 W123)))

A sentence "SS" is formed by combining a sentence "SS", a conjunction "CN" (及び), and a sentence "SS".

5) Verb frame

Example: ((STATE(OPEN) —-> STATE(CONTROL5)) DEMON)

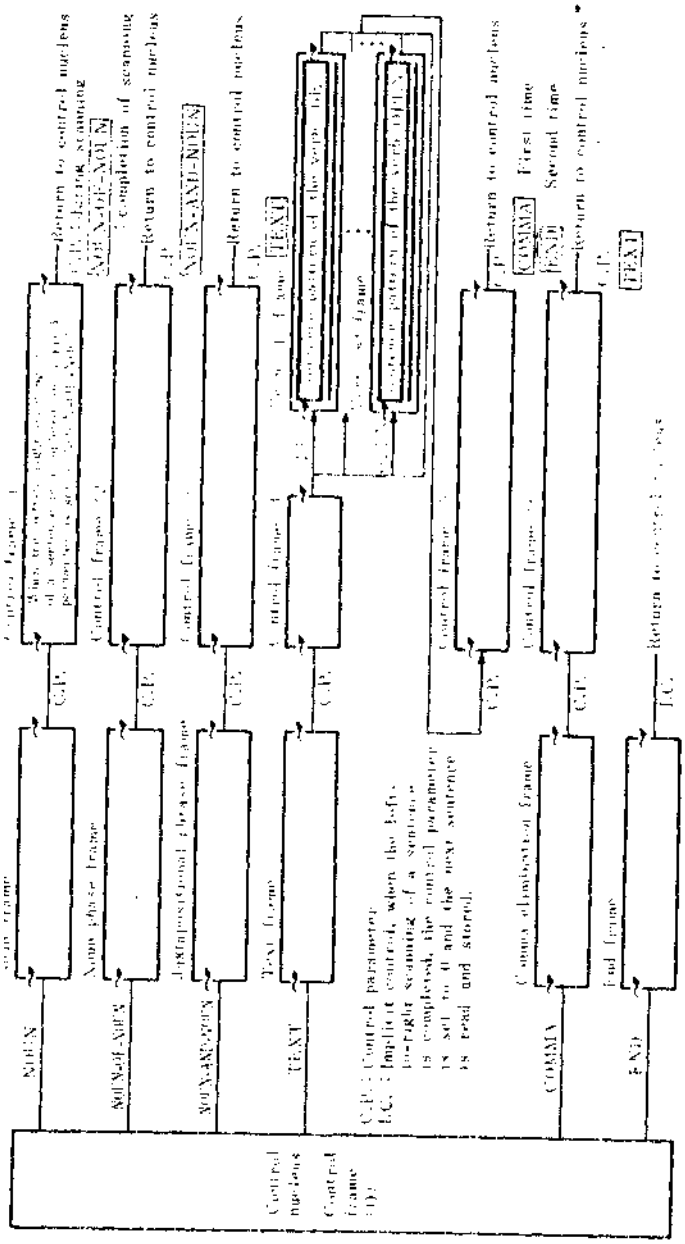

Fig. 6—Grammatical rules specified in FKR-0.

```
(R(COND(H4 C7 V0) A')                    (PARM(I'_V0  W@1@2@3)))
 DEMON:PROC;  /*  ACTOR  AND  INST  ARE  SLOTS.  */
 ACTOR=WORD(2);/*  A  WORD(*)  IS  THE  CONTENTS.  */
 INST=WORD(3); /* OF THE STACKS.                    */•
 END;
```

The surface case structure "H4 + C7 + V0" is changed to a deep case structure "A' + I' + V0". DEMON is a procedure. A noun phrase "C7" is a combination of the concrete noun "NC" (カギ), and postposition "X7" (で). If the surface case structure is "H4 + C7 + V0", a verb "V0" (open) has an agent case "A'" and an instrument case "I'". A variable "WORD" designates a slot in the stack.

6) Comma elimination frame

   Example: ((STATE(COMMA) —-> STATE(CONTROL6)))
   (R(COND(XA@_) XA) (PAEM(0 W1)))

   A comma"" ."(,) is eliminated. A postposition "XA" (に) is unchanged.

7) End frame

   All translation processes end.

3.4 Model of ATLAS/I

Figure 7 is a simplified model of ATLAS/I which includes an input tape, an output tape, a stack, a control section, a dictionary, a register, and grammatical rules (rule representation frames and control frames). When scanning an input tape, the stack is used as a table for temporary storage; at reduction, it is used as a table with attributes and equivalents. The dictionary is a table with words, attributes, and equivalents and is used as a table for lexical rules. The word "太郎", for example, is stored as (太郎, noun, Taro) in the dictionaries. The character strings "太郎がカギで開ける.…", for example, are stored in the input tape. The control parameter is set in the register.

3.5 Initial state of ATLAS/I model

"NOUN" is set in the register as an initial value. Grammatical rules have predefined values. The input head points to the leftmost position of the input tape. The output tape is blank. The output head points to the leftmost position of the

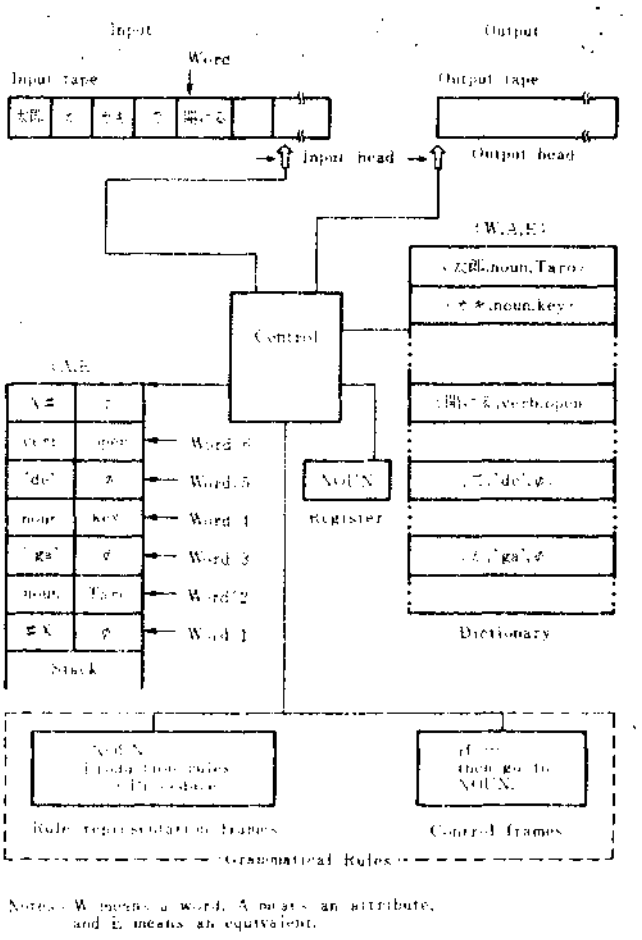

Fig. 7-Model of ATLAS/I.

output tape. The initial value of the slots in the stack is (#X, $), meaning the top of the sentence and null string (ϕ).

3.6 Flow of ATLAS/I model

3.6.1 Phase A: sentence processing

In phase A, one sentence of a text is translated.

1) State (1): scan

The words in the input tape are scanned by the input head and the dictionary is accessed to determine the attributes and equivalents. When found, these attribute? and equivalents are stored in the stack, and then the input head advances one word to the right. When the input tape is scanned, the stack is used as a table by the control section.

The control section scans the words of the input tape. If a period " . " is encountered, it stops scanning and stores (X#, ϕ0) in the stack.

The rule representation frame that is pointed to by the control parameter is referenced, and the control causes a translation from state (1) to state (2).

2) State (2): reduction and code generation

The control section checks if the slots in the stack are (#X, ϕ), (SS, a character string), and (X#, ϕ). If so, there is a transition from state (2) to state (4); if not, the control section checks whether the attributes in the stack match those in the input pattern of the production rule (P). If not, the control causes a transition to the state specified by the rule representation frame.

If matched rule (P) does not exist and if the rule representation frame does not specify the new state, there is a transition to the default state specified by the top-level control frame called the "control nucleus". If matched rule (P) exists, the equivalents (SE) in the stack whose attributes (SA) match the attributes of the rule (P) input pattern are used as parameters of the rule (P) action function. Figure 8 is a general diagram of the organization of the grammar. The input and output patterns are organized according to the sequence of attributes. The action function of rule (P) pops the equivalents (SE) and attributes (SA) from the stack, and pushes the. attributes of the rule (P) output pattern and the character strings created by this action function as the new slots (SA', SE'), whose number equals that of the rule (P) output pattern. The control returns to state (2).

3) State (3): Frame transition

The control checks the control frame, determines the name of the next

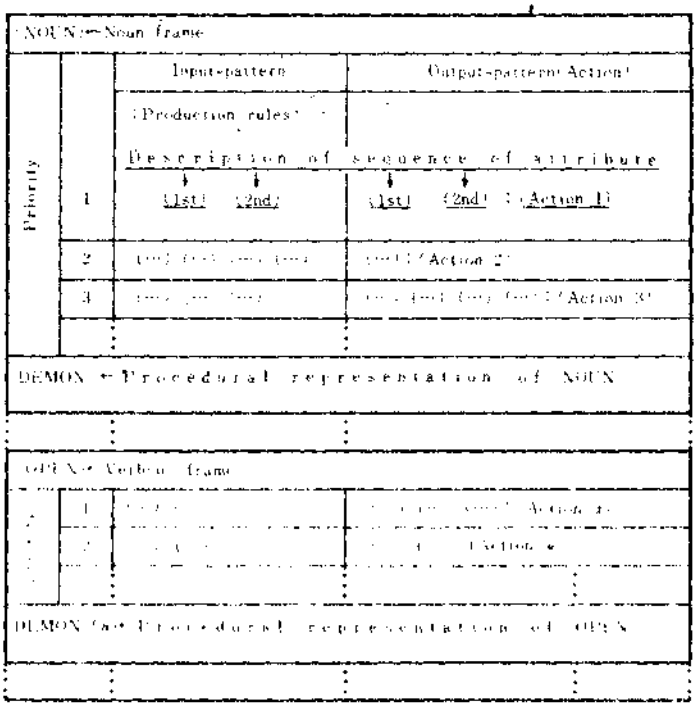

Fig. 8-Structure of the grammar.

rule representation frame, and stores this name in the register. After selection of this rule representation frame control passes to state (2).
Remark: The control frame determines the termination of phase A. At termination, the control causes a transition to phase B.

4) State (4): accept

Control pops the character string of the slot (SS, a character string) from the stack and writes this string into the output tape.

3.6.2 Phase B: text processing

The text is translated in phase B. Control continues phase (A) until the input head arrives at the rightmost position of the input tape.

All text translation processes end when the input head arrives at the right-

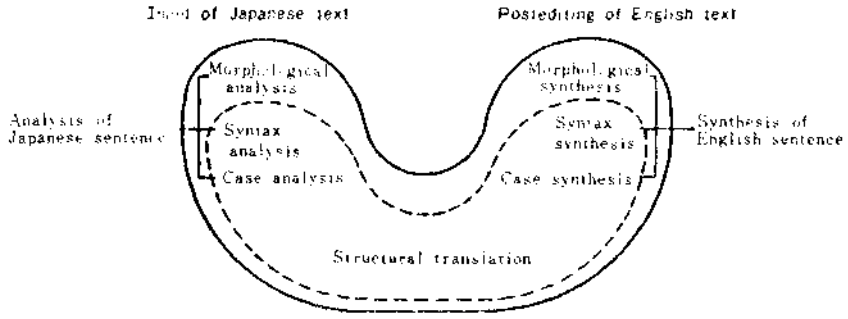

Fig. 9-Flow of ATLAS/I,

most position of the input tape.

4. ATLAS/I machine translation system

ATLAS/I is currently used in the limited application of translating software-related reports (mm Japanese into English. The number of sentence translation patterns is gradually increasing through the addition of grammatical rules and vocabulary.

4.1 Japanese to English machine translation

Machine translation involves three stages: input of the original Japanese text, translation, and postediting of the translated English text (see Fig. 9).

ATLAS/I currently integrates three processes: analysis of an original Japanese sentence, structural translation from Japanese into English, and synthesis of the English sentence. The "case" of noun phrases, which is the relation of noun phrases to verbs, is checked while the syntax is analyzed. This case analysis is performed with reference to the production rules. When the matching rule is found, case and syntactic synthesis of the English are performed, and the synthesized English sentence is printed. These production rules are defined in FKR-0.

The machine translation processes achieved through the use of FKR-0 are delineated in Fig. 9 by dotted lines. Figure 10 shows an example of Japanese to English machine translation processes which follow the flow shown in Fig. 9.

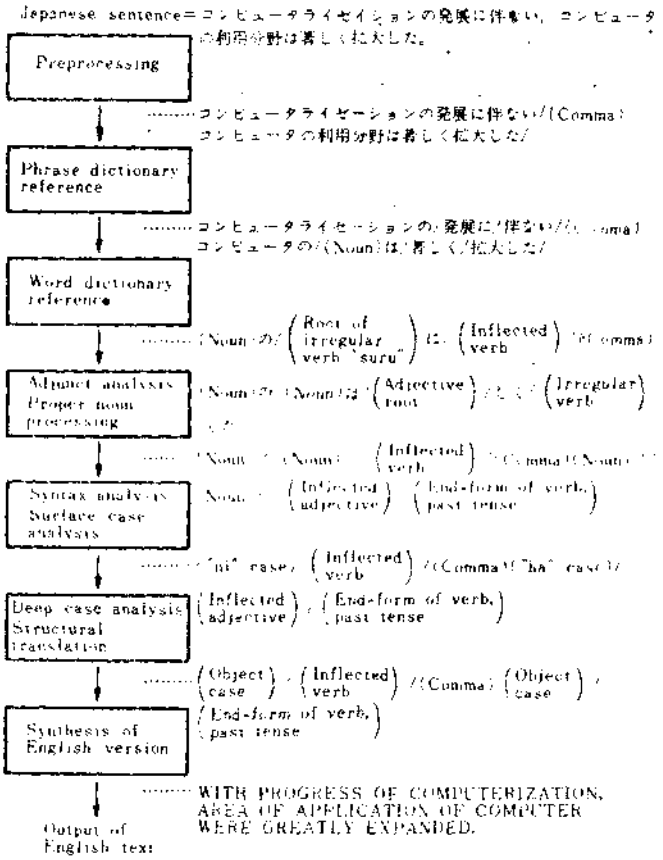

Fig. 10—Example of Japanese-English machine translation.

When the Japanese sentence is typed in, morphological analysis is performed first. This is followed by syntax analysis, ease analysis, structural translation to English, synthesis of the English sentence and, finally, output of the corresponding English sentence.

4.2 Morphological analysis

Morphological analysis consists of preprocessing, phrase dictionary refer-

ence, word dictionary reference, adjunct analysis, and proper noun processing.

1) Preprocessing

The Japanese sentence is segmented by a period (.), question mark (?), exclamation mark (!), commas (,) and parentheses.

2) Phrase dictionary reference

Idioms, compound words, and word phrases written in hiragana (cursive form of the Japanese alphabet) and kanji (Chinese characters) are assigned attributes and an English equivalent.

3) Word dictionary reference

Each kanji or katakana (blocked form of the Japanese alphabet) word is assigned attributes and an English equivalent.

4) Adjunct analysis

After the phrase and word dictionaries are referenced, the character string is checked to extract adjuncts.

5) Proper noun processing

Any word or phrase entry that is not found in the above-mentioned dictionaries is given the semantic marker of a proper noun. Numbers are treated as proper nouns.

4.3 Syntactic analysis and case analysis

1) Syntactic analysis

Noun relations, noun phrase relations, and complex sentence relations are rearranged according to the production rules of the FKR-0 specifications.

Case analysis includes both case analysis and deep case analysis.

2) Surface case analysis

The relation between a noun and a postposition is analyzed to check for agreement with the production rules of the FKR-0 specifications. If it matches any of the production rules, the surface case is determined.

3) Deep case analysis

The relations between a verb and surface cases are analyzed. This pattern matching is intended to analyze surface cases in order to extract their deep cases. The deep case extracted by pattern matching means "when", "where", and "who does what how". It is possible to determine time (when), place (where),

actor (who), and object (what) by referencing the grammatical dictionary of FKR-0.

4.4 Structural translation into English and synthesis of English text

1) Structural translation into English

The production rules of the FKR-0 descriptions contain a Japanese to English pattern translation table. Pattern matching is done by selecting a structural translation pattern. The English synthesis program is invoked on the basis of this selected pattern.

2) Synthesis of English text

The corresponding English text is synthesized by a process which includes assumption of omitted words and plural forms. Omitted words include implicit prepositions and articles.

5. Conclusion

The major methodological problem in a machine translation system is how to separate the translation program from the grammatical rules. In ATLAS/I, grammatical rules are stored in the new form of knowledge representation called FKR-0. In FKR-0, a production system is combined with a procedural representation and systematized into a state transition network. Rule representation frames and control frames arc provided for efficient system operation.

ATLAS/I is currently operational with about 2500 words and 400 grammatical rules for translating software-related reports from Japanese into English.

FKR-0 allows the system to gradually increase the number of sentence patterns, and this expansion is currently underway in the ATLAS/I system.

6. Acknowledgement

The authors wish to express their sincere gratitude to all those who gave continued and valuable guidance in the field of machine translation. In particular the authors wish to thank the members of the Development Division in Computer Systems of FUJITSU.

References

1) A Barr and E.A. Feigenbaum: The *Handbook of Artificial Intelligence,* Vol. 1, William Kaufmann. Inc., Los Altos. Calif., pp. 231-38(1981).
2) W.J. Hutchins: "Progress in Documentation-Machine Translation and Machine-Aided Translation," *J. Document.*. 34, *2* (June 1978).
3) N. Nagao: "Machine Translation." (in Japanese), J. *Inform. Process.,* 20, 10, pp. 896-902 (Oct. 1979).
4) H.E. Bruderer: *Handbook of Machine Translation and Machine-aided Translation,* North-Holland Publishing Company, Amsterdam, (Aug. 1977).
5) A. Barr and E.A. Feigenbaum: op. cit., pp. 114-222.
6) S. Sawai. M. Sugimoto and N. Ukai: "Knowledge Representation FK.R-0 and its Application to Machine Translation," (in Japanese), WGAI Preprint of IPFJ, AI 19-3, (June 1981).
7) R. Jackendoff: "Toward an Explanatory Semantic Representation," *Ling, Inq.,* 7. pp. 89-150(1976).
8) C. Fillmore: "The Case for Case," Eds. Bach and Harms, in *Universals in Linguistic Theory,* Rinehart and Winston, (1968).