The TAURAS Design Philosophy

Shin-ya AMANO Hideki HIRAKAWA Hiroyasu NOGAMI

R & D Center
Toshiba Corporation
1, Komukai Toshiba-cho, Saiwai-ku,
Kawasaki, 210 JAPAN

1. Introduction

TAURAS* is an experimental model representing our machine translation technology. Its operational model is AS-TRANSAC**, which can be implemented on TOSHIBA engineering work station AS3000 and 4000 series.

The main point in the TAURAS design philosophy is improvement in overall translation efficiency, from input of source text to output of target text. This means realizing automation for the translation process and the end of the traditional domestic manual industry.

2. Translation Process Automation Realization

For realizing translation process automation, the process must be divided into subprocesses. Figure 1 shows the total translation process and the subprocesses. This concept has previously been proposed, for example, by GETA. But it was proposed from the view point of the software engineering. TAURAS regards the subprocesses in a beltconveyor line as automation.

Each subprocess should pursue its own highest efficiency, while still maintaining a good interface between the neighboring subprocesses to make translation automation successful.

TAURAS has the following five subprocesses:

1) Source text input through an OCR
2) Correction and modification of input text, using a spelling checker and pre-editor
3) Translation in a batch mode

*TAURAS (Toshiba Automatic tRAnslation system reinforced by Semantics)
**AS-TRANSAC (ASseries TRANSlation Accelerator)

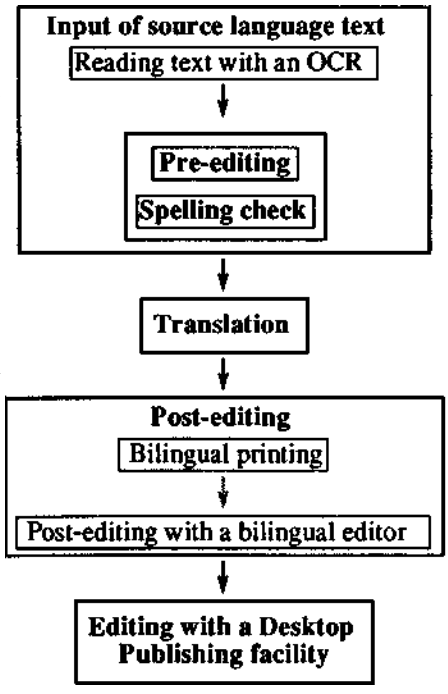

Figure 1. MT process

4) Post-editing in an interaction mode with bilingual editor which has the same screen image as the bilingual print for post-editing
5) Final editing with a desktop publishing facility

Figure 2 shows time factors and their improvement factors. To minimize each subprocess time, the following factors are the most important:

1) OCR
   accuracy
   function
2) Translation quality
   analysis grammar (parser)
   dictionaries
   generation grammar (generator)
3) Post-editing function
   screen configuration
   pointing tool

TAURAS is improving these factors.

TIME FACTOR	IMPROVEMENT FACTOR
Input source text Correction and modification	OCR accuracy spelling checker
Translation	translation speed
Post-editing	translation quality man-machine interface customizing tools
Final editing	DTP

Figure 2. Time factor and improvement factor

3. Improvement Factors

3.1 OCR

Any text usually includes complicated forms. It will have tables, figures, headers, footers, etc. besides the body. It will also employ multi-column format. These factors damage OCR accuracy fatally, if it does not have functions to treat them appropriately. In the worst case, manual input would be faster than OCR input.

Accuracy is a critical OCR condition. In practical use, the word recognition error rate does not go below 5%. A practical error rate will be 5-10%. This rather high error rate does not just result in word reading error only. Generally, texts have a lot of word coinage, especially in case of technical manuals. This raises virtual error rate, though OCRs read letters correctly. Spelling checkers are very important to reduce processing time for correcting errors.

3.2 Translation

Needless to say, translation quality is the most important factor. Translation speed is not critical, if hardware speed is a few MIPS. Translation quality heavily affects post-editing time, which occupies a large part of translation. Improvement of translation quality results in a parser, a generator, and dictionaries. These details, especially about parser including semantic analysis, will be described in Section 4.

3.3 Post-editor

Post-editing time is influenced by translation quality, man-machine interface, and human editor skill. Man-machine interface is composed of editing functions, screen configuration, keyboard configuration, pointing tool, etc. These factors are rather difficult for machine translation designers to design best, if they do not develop special hardware for machine translation systems.

Human editing skill is also important and is often neglected. For automation, human experts are necessary for each subprocess, especially in the post-editing.

4. Translation Method

4.1 Dictionaries and Morphological Analysis

TAURAS has three dictionaries;

1) Common word dictionary
   This dictionary has about 50,000 general-usage words and idioms.
2) Technical term dictionary
   This dictionary has a maximum of 250,000 technical terms.
3) User-defined word dictionary
   Words specific to users should be stored in this dictionary, even if their standard translations are stored in the general dictionary.

Words in English sentences appear morphologically complex. The morphological analyzer basically divides a word into morphemes and constructs a word structure, as shown in Figure. 3.

SW:	source word (infinitive)
POS:	category
NUM:	number
GEN:	gender
PSN:	person
TW:	target words (translations)
SM:	semantic markers
OTHERS:	tense, aspect, modality and so on
PLR:	pointer to the lexical rules

Figure 3. Word Structure

SW, POS, TW, SM, and PLR are necessarily provided by the dictionaries. NUM is also provided by the dictionaries, only if the word has an irregular form, such as "feet."

4.2 Syntactic Analysis

Syntactic analysis and semantic analysis do not function sequentially, but they proceed in an interactive manner. Their roles can be clearly divided as a module. The syntactic analyser employs ATN-like fashion. Figure 4 shows the flow of the syntactic and the semantic processing and their relation.

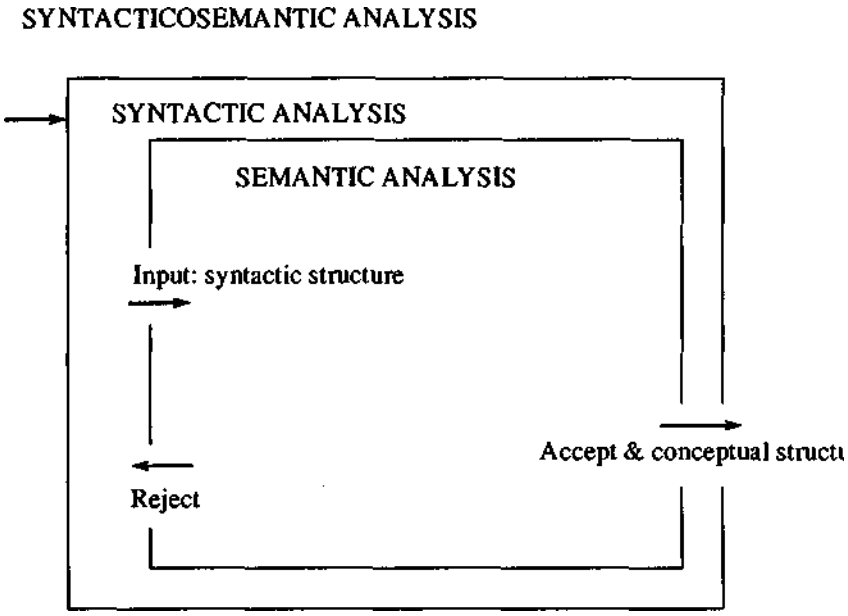

Figure 4. Syntacticosemantic Analysis

Syntactic analysis features are as follow:

1) The syntactic analyzer always derives only one syntactic structure for a string of categories in a sentence. Ambiguities arise from the fact that individual words can often be ambiguous, in regard to their category (lexical ambiguity), or because of different combinatorial possibilities for category strings (structural ambiguity). In the former case, ambiguity is resolved in the normal manner of syntactic parsing, that is by eliminating category values that do not permit coherent word category combinations. In the case of structural ambiguities, these are implicitly represented in the syntactic structure (henceforth called "implicit ambiguity").
Semantic analysis will enable constructing a plausible conceptual structure, resolving such implicit ambiguities.

2) The syntactic analysis is purely syntactic, and syntactic rules have no semantic conditions. A well-known example is the following:

1) He promised her to go.
2) He persuaded her to go.

These two sentences have the same surface syntactic structure, but they have different conceptual structures, corresponding to the different interpretations of the deep subject of "to go."
In the authors' method, conceptual interpretation is accomplished after syntactic analysis. The syntactic analyzer makes a unique syntactic structure for the same sequence of categories with different conceptual interpretations. Thus, syntactic rules do not need to have semantic conditions.

4.3 Semantic Analysis

The semantic analyzer constructs a semantic interpretation and simultaneously makes conceptual structures for the target language.

The proposed semantic analysis method is lexical-based. A typical example is shown in 4.2. Though both sentences have the same sequence of categories, they have different deep subjects for the infinitive "to go." This means these two sentences have different conceptual structures and this difference results from difference in the meanings of "promise" and "persuade." "Lexical-based" implies more than this fact. From the conceptual linguistic point of view, it insists that semantic rules should not be mixed with syntactic rules. According to this schema in the present system, semantic rules are attached to words as lexical rules. This makes syntactic rules simpler.

Sentences which appear in real documents are far more complex and show a wider variety of structure than sentences considered in what is called Chomsky's competence paradigm. Even purely syntactic rules for those sentences become very complicated. Introducing semantic elements into the syntactic rules results in syntactic rules which are much more complicated and makes the system development very difficult.
The proposed system adopts lexical rules as semantic rules to avoid these disadvantages.

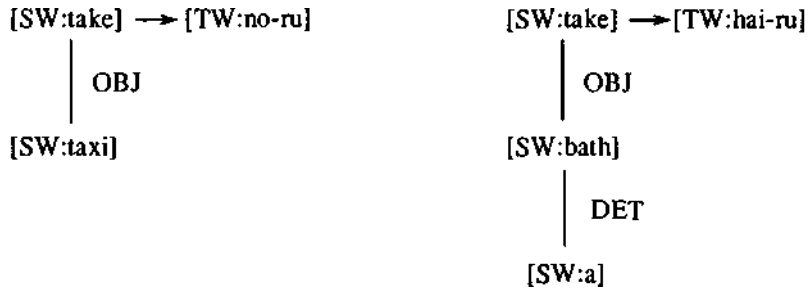

Figure 5. Lexical Rule Concept

5. System Configuration

The translation software is written in C and runs under UNIX* on AS 3000 and 4000 series. The total software configuration is shown in Figure 6. The translation unit and the bilingual editor run in parallel, the translation unit translating source text input via a keyboard or from a disc, an OCR, etc., while the bilingual editor is used by the operator to correct and edit both source and target texts, independent from the translation unit.

6. Conclusion

A brief explanation for the TAURAS design philosophy has been presented. To realize translation automation or to obtain high efficiency in translation, non-linguistic factors such as text input, a post-editor design, customizing tools whose explanation is omitted from this paper, and human user's skill are important, as well as linguistic factor. Rough machine translation estimation is shown in Figure 6. It indicates that the post-editor and human editor's skill is very critical for machine translation systems introduction.

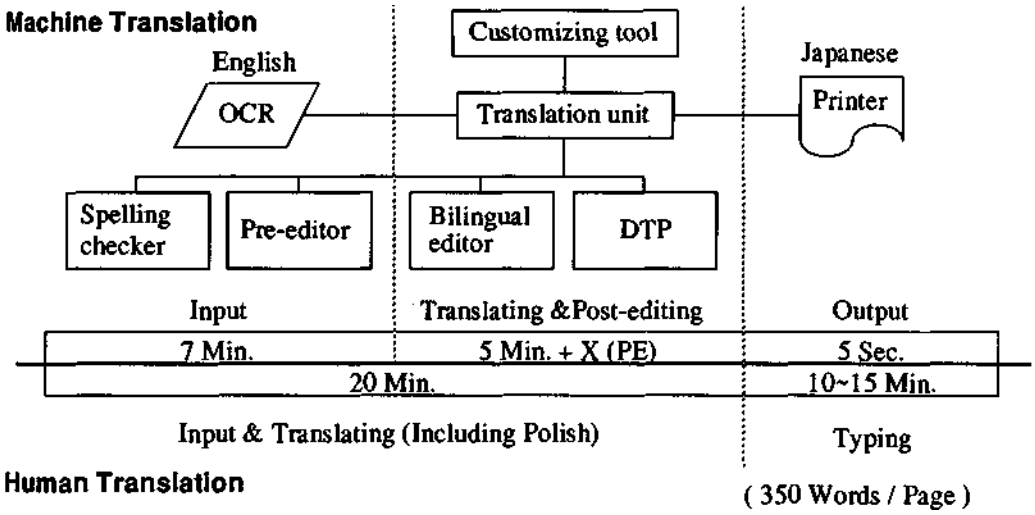

Figure 6. Software Configuration and Rough estimation of translation time

REFERENCES

W.A.Woods (1970). 'Transition Network Grammar for Natural Language Analysis', Communication of the ACM Vol.13, No.10

John Chandioux (1976). 'METEO: un systeme operationnel pour la traduction automatisee des bulletins meteorologiques', META, Vol 21, No.2, pp.127-133

Isabelle,P., Bourbeau.L., Chevalier,M., Lepage,S. (1978). 'Description d'un systeme de traduction automatisee des manuels d'entretien en aeronautique', TAUM, Universite de Montreal

Shin-ya Amano et al. (1989). 'What Should a Translation Work Bench Be Like?', Manuscripts of IFTT'89 (International Forum for Translation Technology)

* Unix is a Trademark of Bell Laboratories