[*Terminology Bulletin* 45, 1984, pp.25-37]

CHANGES AND IMPROVEMENTS TO THE EUROPEAN COMMISSION'S SYSTRAN MT SYSTEM 1976/84

When the Commission of the European Communities bought its first Systran system in 1976, in what might be called the bronze age of machine translation, it was a system of some 30 000 lines of programming and came with a dictionary of around 6 000 entries. Today, such figures appear laughably small, but for our purposes, Systran was the best there was. Faced with a growing mountain of documents and severe difficulties in obtaining enough qualified translators to produce them in 6 (now 7) working languages, the Commission had wondered about the possibilities of computerization of translation, had ordered a study of all systems available world wide, subsequently published as the 'Handbook of machine translation and machine aided translation,' Herbert Bruderer, Amsterdam 1977, and had come to the conclusion that of the half-dozen operational systems available at the time, Systran was the most suitable as the basis for an extended trial.

Eight years later, while machine translation may not have reached its nuclear age, we are certainly in its age of steam, and we are still with Systran. Systran '84 however, looks very different from Systran '76.

On the one hand this is because the solitary translator battling away at the problems in splendid isolation - Ian Pigott, who will be speaking tomorrow morning - has metamorphosed amoeba-like into a team of four full-time translators controlling about a dozen linguistic programmers and dictionary coders. This paper would not be complete without a tribute to this linguistic staff, who have stuck with us through thick and thin under a variety of management hats and have stoutly written the difficult bits of the papers whenever one of us wanted to go swanning off to a conference!

On the other hand, the animal itself is very different. Steam-age Systran now has about one hundred thousand lines of programming and the dictionaries in each of the three operational language pairs (English-French, French-English, English-Italian) amount to about 70 000 one-word entries and 35 000 multi-word expressions.

To a large extent, this increase in size has merely meant more of the same, both in the programs and in the dictionaries. On the dictionary side, facilities for single word entries, for contiguous multi-word expressions, as well as for non-continuous conditional expressions, were already in existence when we bought the system, and much of our early work consisted in coding up thousands and thousands of such entries. Our source for this coding was the fairly conventional one of a large data base, in our case the Food Science and Technology Abstracts data base, to which we happened to have access and which fitted in with our probable future use of the system. A KWIC index

was run on this data base, and every word or expression occurring with a frequency higher than 5 was entered in the dictionary.

We used the same base as the starting point for our work on improving the programs. A list of the more glaring errors in the English-French translation was drawn up, for attention by World Translation Centre of La Jolla, California, from whom we had originally bought the system.

After this initial stage, however, we went over to the practice of using as our test corpora real texts which we simply purloined from the translation departments' in-tray. Our procedure was to take such a production text, run it through Systran, compare it with the human version, and pick out the errors. Or rather, take out the very few correct items in a mess of incorrect verbiage! Missing coding, whether stems or expressions, was handled either by a team of multi-word free-lance coders, or by the two Commission translators by then working full-time on the project, and errors in the linguistic programs were noted down and passed to La Jolla for their attention. And it was while we were working on the sort of texts that our human translator colleagues were translating every day - texts drawn up with complete freedom, not constrained into an artificial format such as that of a data-base entry - that we found a need not only to continue with the existing features but to have new and additional ones developed.

One of the fundamental problems of any dictionary-based MT system such as Systran is that of Not Found Words. Originally, such words as were not found by Systran in its dictionary merely appeared in the target text in their source-language form. Indeed, in our early translations, a typical sentence would have more NFWs than words which were found! Once we had got the proportion down to an acceptable level, however, we took steps to improve the handling of the remaining ones.

Systran already incorporated morphological tables of endings, enabling this part of the program to make at least an informed guess as to the part of speech of the word and to use this tentative identification in its analysis of the rest of the sentence. Even though it cannot of course translate an unknown word it is often able to put it in the right grammatical place in the target sentence, to link up presumed nouns with adjectives of the same number and gender or vice versa, and so on.

The innovation introduced at this time was that the routine also added semantic information to the words it guessed were nouns. An unknown word ending in '-meter,' for example, would automatically be coded 'device,' one ending in '-ogie' or '-isme' would be coded as some branch of science, and so on. Such semantic coding could then be used subsequently by the more complex dictionary entries, or by the rearrangement programs at the Synthesis stage, with the result that an unknown word could travel right through the translation process, unidentified and untranslated, but with all the right things happening to it as it went.

Some years later, we developed further this process of handling NFWs. It had been noticed that several categories of endings in French and English were identical ('-ation' is one, '-ine' is another). To the English translator post-editing the raw output, therefore, it doesn't matter that what he has on his paper or his screen is actually the French word 'commutation,' which the system has failed to find. As far as he or she is concerned, what is there is the English word 'commutation.'

This led us to the realisation that other endings, while different between the two languages, are equally standard. In consequence, the system was refined to transform certain standard endings on NFWs into corresponding standard suffixes in the target language. Systran French-English, for example, not finding the word 'radiologue,' would spot the standard ending at the morphology stage, add a corresponding English ending at the transfer stage to give 'radiologist,' and hope for the best! Being an optimistic little creature, Systran would also add some semantic information - in the case of '-ogue' to '-ogist' it would be the codes for 'human, profession.'

In this particular case, the subroutine would have worked brilliantly, the post-editing translator wouldn't even notice the trick, and the linguists who devised the scheme would grin smugly.

There would be less enthusiasm if it had tried the same dodge on the word 'pirogue,' a canoe, and decided that it was the human profession 'pirogist!'

Similarly, NFWRTN which was already handling numbers, fractions, and the problem of the decimal point or decimal comma, was modified to detect NFWs starting with figures and ending with the sequence 'ième,' or even a single 'e,' recognise the word as an ordinal numeral and add the appropriate target-language ending.

At our present, level of development, individual NFWs are likely to fall only into two categories: either highly technical and specialised words, or else proper names, whether geographical or personal. By the time we bought Systran it already 'knew' that 'Everest' is a place, but it had to be taught, in 1980 for English source and 1981 for French, to recognise 'Mallory' as a person.

An aspect requiring further work on the part of our team, on the other hand, was the detection of '<u>Found</u>' words, starting with a capital letter and being used as proper nouns. One of the structural passes that follow dictionary lookup and homograph resolution has the job inter alia of spotting these, by testing whether the word is in apposition with 'Monsieur' or some other title; or with another word which has already been identified as a name; or in enumeration with another capitalized noun or proper noun, and keeping them in their source-language form. Before we had this subroutine working as well as it does now, the head of translation at the Council of Ministers, Mr Duck, of course used to be translated as 'M. Canard,' and the head

of the interpreting service for Commission and Parliament, Renée Van Hoof, used to be rendered as 'Madame Sabot de Camionnette.'

It also took us quite some time to work out who on earth was 'Dr. separated into various volumes,' until one of us spotted that the past historic of the obscure verb 'tomer' was identical with a famous name in machine translation - Toma! There are those who feel that Systran lost something when it stopped making such charming howlers.

A further step toward the resolution of the NFW problem came to light in particular as we started translating scientific texts. What was one to make of the following sentence in the middle of an otherwise reasonable English translation: 'Simple furnace drank efficacious methods tribunal osmosis of gold blood toilet, Wheeler, Cranfield, 1983?' At first sight, not much!

But the human reader, infinitely brighter than the cleverest computer, spots the clues in the format, namely a proper name, a place, and a date, all three together at the end of a sentence, and realizes that this must be the title of a publication. But what on earth can be the original French have been? This was a question our translators were asking themselves several times a week. The answer is that the original was not French at all. The text as a whole was in French, certainly, but right in the middle of it comes the title of a learned paper published in English. And this title was: 'Four simple but efficacious methods for osmosis of blood or water, Wheeler, Cranfield, 1983.' Simple, but do not forgot that Systran thinks that it is translating out of French. As far as it is aware, the words 'efficacious', 'methods' 'osmosis,' 'blood,' 'Wheeler,' and 'Cranfield' are not in its French dictionary, although it will guess that 'methods' is a plural noun and 'efficacious' is an adjective agreeing with it. That leaves the words which it thinks are French, namely 'four - furnace,' 'simple - simple,' 'but - drank,' 'for - tribunal,' 'or - gold' and 'water - a polite word for *a* toilet!' 'Simple furnace drank efficacious methods tribunal osmosis of gold blood toilet."

Our solution to this problem was the pragmatic one of incorporating a counter mechanism, so that if the proportion of not found words in a sentence exceeds 50%, the sentence as a whole is left untranslated, on the theory that at our present level of development, a sentence with that many NFWs in it cannot be in one of the languages we have covered.

Complementing this, another routine detects that any sentence in which all the words are capitalized is a title or a heading, and thus knows that it must ignore any inferences about proper names which it would otherwise draw from the capitalization.

Our next NFW problem was also one which occurred wherever the text was written in capital letters. Whereas most of these improvements applied to all systems, this is a specific French problem, namely that in capitals, accents are not shown. A phrase such as 'La Communauté

élargie', which would normally be perfectly translated as 'the enlarged Community' would remain in French if it had been appearing in capitals, because 'communaute' and 'elargie' are not in the dictionary. An intermediate solution was to enter such words as inflexions, 'communaute' being an inflected form of 'communauté,' but obviously, this could only be a stopgap solution applied to the words that we knew would most frequently appear in titles, such as, indeed, 'Communauté,' 'régime,' and 'Traité - Treaty.' As soon as we got to 'Traité', indeed, even if no earlier, it became evident what sort of maze we were getting into, because this artificial word 'traite' is of course a homograph with the third person singular of the verb 'traiter.'

A more radical solution was therefore sought, and a first step in 1978 was a subroutine to add a dummy acute accent before any initial 'e.' This helped, but *as* we got more experienced we also got more demanding, and two years later a lot of work was done on rules for the addition of dummy accents within the NFWs. It was originally thought that it would be sufficient to add this dummy to each and every letter 'e,' but this didn't work either - whereas 'Communaute' now became 'Communauté', 'elargie' would then become 'élargie' which is also a Not Found Word! Rules were therefore drawn up on a morphological basis to ascertain which cases of the letter 'e' should get a dummy accent and which should not. For example, the combination 'ént' is unknown in French, and therefore an 'e' in that position would be left unaccented.

One other minor improvement to the handling of not-found-words was to add to the morphological tables the demonstrative suffixes '-ci' and '-là' as permissible endings. Finding any word ending with either of these suffixes, (which had then made it into a NFW), Systran strips off the last three characters and then looks again in the Stem to see whether the shorter word thus created is in fact a genuine source-language word. This morphological phase subsequently links up with a lexical routine to translate 'ce something-ci' as 'this something' but 'ce something-là' as 'that something.'

Like the postman in the thrillers who is the perfect murderer just because he is so visible, other features of ordinary running text kept coming to light, things which are perfectly obvious once you have noticed them but invisible until you do. An example was the use of a bracketed plural ending: 'Fill in the relevant box(es).' For bronze-age Systran, 'box(es)' was simply a NFW, but with some work in SYN(thesis) it became possible to handle these and have the same bracketed plural in the target language.

Similarly, translating out of English, we began to come up against the problem of the '-'nt' ending, as in 'can't' or 'don't.' A very typical example of the sort of difficulty which emerges only in the real working environment, because in the development phase one tends not to write such casual English.

Our solution to this one consisted in taking an existing Systran feature, and using it backwards. We already had something known as Idiom Replaces, which handle certain multi-word expressions by treating them as one word. These are either complex idioms such as 'dans le même ordre d'idées,' which can be treated syntactically as a single adverb; 'dans.le.même.ordre.d'idées' or alternatively standard phrases - a lot of our bureaucratic work is standardised - where the source-language version is equally standard but syntactically quite different from the source side, such as; 'Les entreprises devront soumissioner conjointement ou solidairement au sein du groupement qu'elles pourront former,' which is transformed into the massive single word 'Les.entreprises.devront.soumissioner.conjointement.ou. solidairement.au.sein.du.groupement.qu'elles.pourront.former,' and translated, quite differently, by 'Consortia must have joint or several liability.'

It occurred to us that the feature could be used in the opposite way - to decompose words rather than create them. Thus a new type of Idiom Replace was developed which, finding the word 'don't' at dictionary lookup, splits it up into 'do' and 'not' and pops both of these words into the alphabetical list of words to be checked against the basic dictionary.

Once we had found that the idea would work, some poor coder had to grind through all the cases, making 'wouldn't' equal 'would not.,' 'he'd' equal the two possibilities 'he had' or 'he would,' and so on.

With the NFW problem at least under control, we had time to take a breather and look at what else was fundamentally wrong. When to our surprise we found that once out in the big bad world of real translation, Systran was frequently having trouble deciding where its sentences ended. "At the fullstop, dummy!" we would snarl in frustration, only to have our hearts melted by a poor confused little system whimpering that it didn't know which full-stop was which. Because out in that big bad world, sentences contain abbreviations, and abbreviations may, or may not, end with a stop, and this full-stop may, or may not be, at the same time the one which ends the sentence.

To crack this one we composed as long a list as we could dream up of abbreviations, divided into those which may end with a full-stop and those which never do. The tables then trigger a program, under which if an abbreviation from the second table is found to be followed by a full-stop, then that must be the end of the sentence. A stop after an abbreviation from list 1 may, or may not, be the end of the sentence, and so further tests and checks have to be set off, such as how many blanks follow the stop, whether the next word is a capitalized NFW (as the full-stop may merely be in the middle of 'M. Dupont') and so on.

We then turned our attention to the design of features to meet problems encountered in particular document types. One of the things we have found during our eight years of development work is that document type, as much as document content, can have a marked effect on the quality of the final output. This was something of a surprise,

although it should be clarified that documents of a new type do not exclusively imply new problems, they may also bring with them new clues leading to easier analysis.

The first specific document type we worked on was that of patents, but here the various steps we took consisted almost solely of work on existing dictionary and program features. The only innovation here was to increase the number of permitted words in a sentence. Up to this point, in 1980, Systran would arbitrarily break at the 105th word in a sentence and start a new one. Which hadn't been a problem, because we never encountered sentences that long! Until we started work in the field of patents, under the guidance of Veronica Lawson, when we found that mammoth sentences were actually quite common in patents. The limit was then increased to 255 words, and another problem was - more or less - resolved.

Our next specific type was minutes of meetings - a not uncommon type of document in a giant bureaucracy like the Commission. Here the problem is that minutes are written in the present tense in French or Italian and have to be translated into the past tense in English, and vice-versa. We thus found a need to develop a group of Typology Categories, or Typcats, complementing the existing system of Topical Glossaries. When a text is handled under Typcat MINUTES, therefore, the tense-change is carried out automatically. Not that that is the whole story: perfect tenses have to become pluperfect, futures have to become conditionals, words such as 'demain' have to be translated 'the day after' rather than 'tomorrow'. Additional work had also to be done on tenses in subordinate clauses, which depending on their meaning may not need to be changed after all.

With the bit now firmly between our teeth, we had a look at the question of imperatives. We had translated a large number of aeroplane and helicopter maintenance manuals for Aérospatiale in Paris, when it became apparent that whereas English imperatives in normal running text are to be translated by a French imperative ('Cease, fire!' - 'Cessez le feu!') it is correct practice in a maintenance manual or similar document to translate them by infinitives ('Fit the oil filter' - 'Monter le filtre d'huile') and vice versa. Here, too, a subroutine had to be implemented enabling this change of mode to be made or not made depending on the type of document. In normal texts, the 'imperative' mode is the default mode, in technical manuals it is the 'infinitive' mode, but in each case the opposite mode may be selected.

The difference may seem minor, but it is such differences which actually cause a translation to be produced, rather than a mere transposition of the words from one language to another.

By now we were in production, of course, and it was at this time that we introduced an innovation which in retrospect seems ludicrously obvious - the splitting up of Systran into a Test version and a Production version. During the development phase it didn't particularly matter that we were all tinkering away on the same

system, so that Tinkerer A's work on one program might cause a puzzling deterioration in Tinkerer B's work somewhere downstream. A couple of excited phone calls, and all was sweetness and light again. This ad hoc approach became quite impossible, however, once we were trying to interest translators, with their own special reservations and sensitivities, in using MT. We certainly could not expect them to post-edit output from a system which one week would translate 'il va de soi' beautifully as 'It goes without saying,' and the following week would produce 'He is going from itself' because one of us had been trying out a bright idea on the Idiom Replace routine.

From our entry into production in the spring of 1981, therefore, each of the operational language pairs has had a Production version, which is at the service of the Translation Divisions and which is fixed, and a Test version whose output quality see-saws from week to week, (although the overall trend is upwards!). Periodically, at the discretion of the linguists in charge of each system, the Test version is upgraded to become the Production version, with a collection of new and improved features, and a new Test version is created for the next phase of development work.

It would have been quite inadequate, of course, to dream up innovations like that one just to keep the customers happy. Of comparable importance were innovations to assist in the work of the linguists by whose efforts and skill the whole edifice ultimately stands or falls.

When we first bought the system, the programs were written entirely in Systran macro, and were opaque to all but the most experienced and dedicated reader. A major step forward was taken when we commissioned the Cambridge Language Research Unit headed by Margaret Masterman, to write a program for us that would automatically annotate the Systran programs into natural English.

A pilot project was run on annotating the homograph resolution routines, and once the feasibility of the project had thus been demonstrated, the entire English-French system was annotated.

This was a help to the programmers, and the dictionary coders were assisted in a similar way by our next innovation, a dictionary concordance.

If an erroneous translation has been caused by a wrong entry in either the single-entry or the noun-group dictionary, the source of the error is fairly easy to find. Once the dictionary of conditional expressions is involved, however, things become a lot more complicated. Picture the poor dictionary coder, puzzling over the sentence 'The pilot was eating his lunch as the aircraft changed its bowl of flight.' In desperation, he would look again and again at the entry 'assiette de vol' and would be absolutely convinced that it had been correctly coded as 'flight attitude.' In this case, the Principal Word of the expression is 'assiette,' and he would therefore find the expression under 'A' in the dictionary. What he would not

spot, however, is an entry in which one of his colleagues, all unknowing, had specified that when 'manger' appears in the sentence, 'assiette' is to be translated as 'bowl.' Here, the Principal Word is 'manger,' the expression comes therefore under 'M' in the dictionary, and our poor coder, scrabbling around under 'A for assiette,' never finds it. This problem was resolved by the creation of the concordance, in which all cases of a word used in expressions, whether as Secondary Word or Principal Word, are listed together.

For a time, this concordance proved very useful, and is indeed still used for certain specialized applications. For general dictionary work, however, its place has been taken by a more sophisticated device, the so-called SQ printout, in which each sentence of the translation is preceded by several lines of information, indicating which dictionary expression has in fact been taken as applicable. This device has proved one of the most useful for dictionary development work, saving hours and hours of searching and checking.

An added refinement of the SQ allows the coder to see not only which entries were selected, but which ones were tested and then rejected. This may be useful for resolving priority clashes - at CLS level, the longer entry takes precedence over the shorter one - or for detecting cases where an entry has blocked a subsequent entry from being used.

Once the dictionary coders had been provided with these tools, enabling them to see what had gone wrong, they immediately wanted new syntactic and semantic codes to make things go better in the future.

An example of such a new code was APPFIX, which causes words in apposition to remain in their source-language order, whereas normally REARR(angement) would have reversed them. 'A4 type paper' would be standardly reversed to give 'papier type A4,' but we can use the code APPFIX to ensure that an 'E type Jaguar' remains an 'E type Jaguar' even in French!

Another such example was the pair of codes REFDE and REFA, for words which take 'de' or 'à' only when they are reflexive, such as 'attendre.'

Subsequent to this, and in part because of these improvements, it became apparent that in certain cases the existing CLS expressions were not adequate to cover all the forms or structures which might occur in real text.

Thus were born two new types of expression, namely homograph resolution expressions and parsing resolution expressions. Whereas the existing expressions are concerned with obtaining the correct meaning on the target side, the new ones are designed to handle problems on the analysis side. As such, they deal with particularly difficult, particularly specific cases, often in fact the hundredth case where the source-language strays from the rules it follows in the other 99.

For example, when the homograph resolution program is testing a word that might be a noun, one of the tests it can apply is whether the word to its left is a verb. If it is, the word under test cannot be a noun. 'Je prends son' cannot possibly mean 'I take sound,' it has to be 'I take his. . .' A noun in French cannot be preceded directly by a verb. And then comes the hundredth case: 'Je prends note.' Rather than coding this one oddity into the homograph routines themselves, however, the new development made it possible to write an HLS expression on 'note,' which in effect will tell the system that just this once the rules don't apply. Observe, however, that no meaning has to be given in this expression to 'note' (although it can be). All that the expression is concerned to do is to get the part of speech right for subsequent analysis.

HLS's are called immediately before the homograph resolution programs proper, and override these programs. Parsing expressions, on the other hand, can be specified for calling at any one of five points of entry: after homographs but before the first structural pass through the sentence; before any of the following passes; or after the last one and before Synthesis.

They are designed to resolve syntactic difficulties such as the parsing problem with the construction 'éviter que l'argent soit dépensé': not 'that the money be spent,' but 'prevent the money being spent.' Such a radical restructuring could be handled by analysis in the traditional sense, but would be unnecessary clutter for such a specific case. It is preferable to write a dictionary expression, in which one single line of programming can check whether 'éviter' governs 'que,' delete the 'que' from the analysis, change the subjunctive subordinate verb to *a* gerund, and exit triumphantly. Once again, it should be emphasized that no meaning need be given, and indeed that the meaning (not the syntax) of the words in this expression are still open to modification if an expression called subsequently finds an appropriate match. (Which will in fact be the case - further down the line, while leaving the grammatical structure untouched, an entry will change 'argent,' governed by 'dépenser' from its Stem meaning of 'silver' to a less poetic but more correct 'money.'

At about the same time, various new coding macros were introduced, to make life more efficient for the dictionary coders. Thus the new macro 'PRPGOV' covers all cases of preposition government, avoiding the need to spell them out specifically by bits and bytes as we had to in the past; NOMGRP replaces three different types of noun-to-noun relationship; and ENUMER covered enumeration in both directions, at a stroke cutting the coding effort by half.

Perhaps the most significant innovation in this phase was the 'SCAN' function. Previously, to give a word a special meaning in a given context, it had been necessary to specify precisely the syntactic relationship between that word and the one indicating the context or subject field. To make 'puits' come out as 'shaft' in a mining context, for example, it was necessary to write dozens of expressions such as 'puits' governing 'de' governing 'mine;' 'puits' governing

'dans' governing 'mine;' 'mine' subject of 'avoir' with 'puits' as its object, and so on. And even then one was far from covering every conceivable case. With the introduction of the 'SCAN' function, the system can now be instructed by an expression on 'puits' to run up or down the sentence and match the expression if it finds any occurrence of the word 'mine,' regardless of its syntactic relationship.

The cases where Systran has to translate texts about pen refills dropped down wells are statistically insignificant!

Another innovation at the program level was to allow Systran to look back into the sentence before the one it is currently analysing. This takes place right at the end of the synthesis of the target language, although the process is not in fact part of the overall process of rearranging a sentence. Nevertheless, looking back at the preceding sentence has to fit in here, after everything else has been done, since only then can information on this completed sentence be stored for use in the next one.

On the one hand, any antecedents in the current sentence of an 'il/elle' in the next sentence are checked for animateness or inanimateness and this information is stored in a couple of bytes of word zero of the following sentence, to help with the resolution of the pronouns. If the antecedents of pronouns in Current Sentence + 1 are themselves pronouns, in Current Sentence, then their antecedents, in Current Sentence - 1 are examined and this information stored in Current Sentence + 1.

A similar procedure is followed for predicate complements. The information that the complement was a noun or an adjective respectively is also stored, as this may be needed to resolve a pronoun in the next sentence. As in:
'Son père était boulanger. Le fils l'est aussi - The son is also one' but 'Son père était horrible. Le fils l'est aussi - The son is too.'

So as not to neglect any aspect of the whole Systran procedure, innovations were also made on the input and output sides. Originally, input was carried out on punch cards, subsequently replaced by a direct tape machine, but both of these methods were adequate only for development work, in which time was not of the essence, proving too cumbersome for actual production work. The poor punch girls had to remember to type in the hieroglyph 'C$' before a letter, for example, to indicate that it was a capital, or '$$LN' to show a line change.

When we went into production, obviously we could not be seen to be struggling away at the level of a Hollerith card punch, and we tried to make the application as user-friendly as possible, by installing a Wang word-processing network. Our choice settled on Wang principally because at that time they seemed to be the best at communicating with the IBM mainframe on which Systran runs, and so far, at least from the technical point of view, our choice seems to have been a good one.

Now all the input typists have to do is copy the text. There are no format constraints, no special representation of characters, it is pure copy work. When the whole text has been copied into the word processor, the typist enters a small Basic program via her Wang applications menu, requests the translation by typing in the document's number and pressing one key for the desired target language, and as far as she is concerned the job is finished.

Similarly, for the production team actually batching up the texts and sending them down the line, much of the parametrisable work is hidden behind simple commands, set to default to the most common case.

Of course, this entailed a considerable amount of work to ensure that what Wang sent down the line was actually readable by the IBM and by Systran.

For historical reasons, two programs are used to produce a TDCS (Text Data Control System) input file from the raw text. One is the original Systran front-end program SETUP, which accepts the artificial format mentioned a moment ago and which was modified to produce TDCS files; the other, which forms the new front end of the system, is called NATTEX.

Within NATTEX, certain groups of characters, such as any sequence of numerals preceded and followed by hyphens and standing alone on a line, are recognised as introducing a new page, and having detected the limits of the page the program checks whether there are columns of text within it, by looking in each physical line for gaps of three blanks or more (trailing blanks being ignored). If such gaps are found, succeeding lines are checked for gaps of at least two blanks right-aligned with the original ones. Any such gaps are assumed to form column divisions, and the columnar structure is assumed to continue until one of the column divisions contains *a* non-blank character, or the end of the page. Each column is then processed in turn (the preceding single-column text having been previously processed) and on output is preceded by a control word identifying its horizontal position on the page. The column text is preceded as a whole by a "tab set control word" giving the actual start positions of the succeeding columns as character displacements from the left-hand margin.

The page is thus broken up into blocks of text ready for processing. The first word of each line of the block is subjected to a series of tests intended to determine whether it could be a paragraph identifier or not. A character group which passes all the tests is assumed to be a paragraph identifier and is preceded in the output by the control word indicating a line-change, so as to ensure its appearance at the start of a line in the translation. Originally, the number of tests carried out was rather time-consuming and frequently prone to inaccuracy - for instance, if the year part of a date appeared at the start of a line it was assumed to be a paragraph identifier and separated accordingly - but a major improvement was made once we were consistently inputting on word-processors and it could therefore be

assumed that the input would have had a standard Wang width of 80 characters. It thus became possible to detect a genuine line-change (such as would precede a paragraph identifer) by one simple test: would the first element on a line have fitted on the end of the previous line? If it would have done, then clearly the operator pressed the "return" key intentionally, and so a line-change ought to be inserted.

The "hit rate", in terms of unwanted sentence breaks not generated is now of the order of 95%, as opposed to something like 60% beforehand, and the mill time consumed by the input program was reduced by some 60% as a result of these changes and other similar improvements.

Currently, work is in hand to provide a means of detecting vertical columns of characters, separating the columns and, in conjunction with the new post-processor, reinserting them in the translation. In the present system, there is no information in the TDCS file as to the lateral alignment of columns, which can lead to misleading output: given a typical piece of layout such as a table of contents with page numbers at one side, if the translation of a text line is much longer than the original (as when going from English to French) the numbers may find themselves 'left behind' and vertically shifted from any meaningful alignment with the items to which they refer.

The new post-processor program, in conjunction with a new version of NATTEX, will be able to preserve the lateral alignment of columns.

I am indebted for this latter portion of my paper to Mr Alan Carlisle, who covers the Commission's ongoing Systran d-p work, and who produced a scholarly chapter on text-handling which I then butchered to my own sketchy and flippant style.

In conclusion, if our eight years of development have taught us anything about the design of MT systems, it is that many if not most of the trickiest problems come to light only when the system moves out of its development cocoon into the real world of genuine text, warts and all. The warts are often the most difficult and frustrating to cure, but they can also be the most fun!

Peter J. Wheeler

The Commission of the
European Communities,
Luxembourg