On the machine translation of string indexing languages between English and French using PRECIS as an example

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:


Metadata Record: [https://dspace.lboro.ac.uk/2134/28115](https://dspace.lboro.ac.uk/2134/28115)

Publisher: © Peter John Hancox

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 2.5 Generic (CC BY-NC-ND 2.5) licence. Full details of this licence are available at: [http://creativecommons.org/licenses/by-nc-nd/2.5/](http://creativecommons.org/licenses/by-nc-nd/2.5/)

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
ON THE MACHINE TRANSLATION OF STRING INDEXING LANGUAGES
BETWEEN ENGLISH AND FRENCH
USING PRECIS AS AN EXAMPLE

Volume one - text

by

Peter Hancox

A Doctoral Thesis
Submitted in partial fulfilment of the requirements
for the award of
Doctor of Philosophy
of the
Loughborough University of Technology
November 1983

© by Peter John Hancox 1983
To my parents
and Kathryn
ACKNOWLEDGEMENTS

I would like to thank Mr F. Smith for his supervision, and his family for extending their friendship and interest to me.

Professor P. Havard-Williams was my Director of Research, and I am grateful for his direction and for providing the facilities for completing this research.

I received assistance and encouragement from a number of people within the DLIS, and in particular from Gwyneth Tseng and John White (who between them originally taught me programming), Hilarie Bateman and Cynthia Robinson.

During the preparation of this thesis, I met people from diverse places and backgrounds. Dr Derek Austin, Mrs Jutta Austin, Ms Veronica Verdier and Mr H.L. Somers in particular gave time, helpful insights and guidance.

I wish to acknowledge the assistance given by the late Catherine Denereaz in the construction of the English-French dictionary.

Thanks are due to the British Library (Research and Development Department) for supplying a full copy of the final report of the PRECIS Translingual Project, and to Aslib for a pre-publication copy of an article about the TITUS IV system.

The work was largely funded by a research studentship from Loughborough University of Technology, and the version of the generation module described herein by a Department of Education and Science Information Science Research Fellowship.

Finally, thanks for the encouragement received from all my family, friends and relations, and in particular to my Mother and Father (who proof-read the final text), and to Kathryn Hobbs.
CONTENTS

ACKNOWLEDGEMENTS iii
LIST OF TABLES vi
LIST OF FIGURES vii

Chapter

I INTRODUCTION 1

II A BRIEF INTRODUCTION TO MACHINE TRANSLATION 3
   Direct systems 5
   Interlingual indirect systems 11
   Transfer indirect systems 19
   In conclusion 24

III TRAITEMENT DE L'INFORMATION TEXTILE UNIVERSELLE ET SELECTIVE (TITUS) 32
   Note on J.W. Perry's suggested use of translated abstracts 47

IV AN INTRODUCTION TO PRECIS 51

V APPLICATIONS OF PRECIS IN LANGUAGES OTHER THAN ENGLISH AND THE PRECIS TRANSLINGUAL PROJECT 73

VI THE OBJECTIVES AND DESIGN OF A TRANSFER TRANSLATION SYSTEM AND A METHODOLOGY FOR ITS TESTING 103
   General system design 107
   Methodology 110

VII SYNTACTIC ANALYSIS 115
   Text preparation (Program MT11C1) 116
   Transition network analysers 118
   The dictionary 127
   The grammars 130
   Syntactic analysis (Program MT21C1) 134

VIII SEMANTIC ANALYSIS 142
   The dictionary 146
   The programs 150
   Text preparation (Program MT25C1) 150
   Semantic analysis (Program MT27C1) 154
   Re-unification of parses (Program MT29C1) 162
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIII</td>
<td>SEMANTIC ANALYSIS (continued)</td>
</tr>
<tr>
<td>IX</td>
<td>TRANSFER</td>
</tr>
<tr>
<td></td>
<td>The dictionary</td>
</tr>
<tr>
<td></td>
<td>The grammar</td>
</tr>
<tr>
<td></td>
<td>Transfer (Program MT31C1)</td>
</tr>
<tr>
<td>X</td>
<td>GENERATION</td>
</tr>
<tr>
<td></td>
<td>French generation 1 (Program MT41C1)</td>
</tr>
<tr>
<td></td>
<td>Conditions</td>
</tr>
<tr>
<td></td>
<td>Actions</td>
</tr>
<tr>
<td></td>
<td>French generation 2 (Program MT43C1)</td>
</tr>
<tr>
<td></td>
<td>Conditions</td>
</tr>
<tr>
<td></td>
<td>Actions</td>
</tr>
<tr>
<td></td>
<td>Grammar MTFSG1</td>
</tr>
<tr>
<td></td>
<td>Grammar MTFSG2</td>
</tr>
<tr>
<td></td>
<td>Grammar MTFSG3</td>
</tr>
<tr>
<td></td>
<td>Grammar MTFSG4</td>
</tr>
<tr>
<td></td>
<td>Grammar MTFSG5</td>
</tr>
<tr>
<td></td>
<td>Grammar MTFSG6</td>
</tr>
<tr>
<td></td>
<td>Grammar MTFSG7</td>
</tr>
<tr>
<td></td>
<td>A comment on the grammars</td>
</tr>
<tr>
<td></td>
<td>Creating the target language strings</td>
</tr>
<tr>
<td></td>
<td>(Program MT45C1)</td>
</tr>
<tr>
<td>XI</td>
<td>THE OBJECTIVES REVISITED AND CONCLUSIONS</td>
</tr>
<tr>
<td></td>
<td>Future directions</td>
</tr>
<tr>
<td></td>
<td>BIBLIOGRAPHY</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Machine translation projects and systems (adapted from Hutchins, 1982)</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Schema of PRECIS role operators</td>
<td>71</td>
</tr>
<tr>
<td>8.1</td>
<td>Collected results of semantic analysis</td>
<td>167</td>
</tr>
<tr>
<td>8.2</td>
<td>Table of labels used in semantic analysis</td>
<td>169</td>
</tr>
<tr>
<td>10.1</td>
<td>Analysis of the causes of failure</td>
<td>215</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Basic structure of a direct machine translation system</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Basic flowchart of the SYSTRAN system</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Basic structure of an interlingual machine translation system</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>Diagram of Wilks's English to French translator</td>
<td>18</td>
</tr>
<tr>
<td>2.5</td>
<td>Basic structure of a transfer machine translation system</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Sentence types allowed in TITUS II</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Structure of a TITUS IV proposition</td>
<td>42</td>
</tr>
<tr>
<td>3.3</td>
<td>Structure of the TITUS IV sub-noun group</td>
<td>43</td>
</tr>
<tr>
<td>3.4</td>
<td>Structure of the TITUS IV verb group</td>
<td>44</td>
</tr>
<tr>
<td>3.5</td>
<td>Output from TITUS IV</td>
<td>46</td>
</tr>
<tr>
<td>3.6</td>
<td>Extract from subject index to annual volume of BNB (1980)</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Manual translations by Lamy-Rousseau</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>Extract from the Rouen index</td>
<td>77</td>
</tr>
<tr>
<td>5.3</td>
<td>Extracts from the EUDISED index</td>
<td>81</td>
</tr>
<tr>
<td>5.4</td>
<td>Sample PRECIS Translingual Project lexical record</td>
<td>85</td>
</tr>
<tr>
<td>5.5</td>
<td>Switching procedure of PRECIS Translingual Project</td>
<td>88</td>
</tr>
<tr>
<td>5.6</td>
<td>PRECIS Translingual Project algorithm for &quot;of&quot; and alternative translations</td>
<td>90</td>
</tr>
<tr>
<td>5.7</td>
<td>PRECIS Translingual Project article insertion algorithm</td>
<td>92</td>
</tr>
<tr>
<td>6.1</td>
<td>System flowchart of the transfer translation system</td>
<td>109</td>
</tr>
<tr>
<td>7.1</td>
<td>Output from the text preparation module (MT11Cl)</td>
<td>119</td>
</tr>
<tr>
<td>7.2</td>
<td>Syntactic analysis grammar 1</td>
<td>132</td>
</tr>
<tr>
<td>7.3</td>
<td>Syntactic analysis grammar 2</td>
<td>133</td>
</tr>
<tr>
<td>7.4</td>
<td>Syntactic analysis of string 69</td>
<td>139</td>
</tr>
<tr>
<td>8.1</td>
<td>Semantic analysis grammar 1</td>
<td>156</td>
</tr>
<tr>
<td>8.2</td>
<td>Semantic analysis grammar 2</td>
<td>159</td>
</tr>
<tr>
<td>8.3</td>
<td>A constructed grammar for &quot;calculation&quot;</td>
<td>160</td>
</tr>
<tr>
<td>8.4</td>
<td>Semantic analysis of downward reading of string</td>
<td>303</td>
</tr>
<tr>
<td></td>
<td></td>
<td>163</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>8.5</td>
<td>164</td>
<td></td>
</tr>
<tr>
<td>8.6</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td>9.1</td>
<td>176</td>
<td></td>
</tr>
<tr>
<td>9.2</td>
<td>180</td>
<td></td>
</tr>
<tr>
<td>10.1</td>
<td>197</td>
<td></td>
</tr>
<tr>
<td>10.2</td>
<td>198</td>
<td></td>
</tr>
<tr>
<td>10.3</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>10.4</td>
<td>202</td>
<td></td>
</tr>
<tr>
<td>10.5</td>
<td>204</td>
<td></td>
</tr>
<tr>
<td>10.6</td>
<td>206</td>
<td></td>
</tr>
<tr>
<td>10.7</td>
<td>208</td>
<td></td>
</tr>
<tr>
<td>10.8</td>
<td>210</td>
<td></td>
</tr>
</tbody>
</table>
This project would not have been possible twenty years ago. At that time the first steps were being taken to introduce the computer into library and information retrieval processes. At first, manual operations were mechanized. Then came new thought, and methods of fulfilling the old functions in ways more suited to the computer were devised. The most difficult of these old functions has always been the provision of subject access to information.

As a generalization, the user wants either a document on a very precise subject, or a number of items on a broader subject. It is relatively easy to produce a list of subject headings (an information retrieval thesaurus) which is essentially a list of phrases under which one may collocate bibliographic references. Documents on "information retrieval" would be collected under that heading which would probably be adequate until either a large number of items had been amassed or a user wanted to retrieve references on a more precise subject, for instance, "Information retrieval in Scandinavia". To obtain the relevant references, he must scan through all the citations offered under "Information retrieval" to decide if any include the concept of location in Scandinavia.

This simple listing under one term is clearly unhelpful to the user. The combination of several terms such as "Information retrieval. Scandinavia" is of some use, but liable to mislead. Does this heading refer to the retrieval of information in Scandinavia, or about Scandinavia? The response to this problem has been the production in the last ten to fifteen years of a family of "string indexing languages". These consist of index terms (drawn from natural language) which have been constructed
and linked using regular and explicit syntactical rules to form the strings. Once written by the human indexer, these strings are manipulated by computer to produce a set of entries. The improvement of string indexing languages over other methods of producing a printed index lie in their ability to indicate the relationships between the terms in the string, and thus to reduce the possibility of ambiguity: one of the classic functions of an artificial language.

It is the use of prepositions and (in some languages) case endings to indicate the relationships between terms that makes the project described here more than a simple exercise in looking-up terms in a lexicon and transferring their equivalents. In this sense the project is more akin to the translation of natural language; but it avoids some of the more difficult problems of that exercise, such as the range of discourse, anaphora, and (to a very great extent) homonyms.
CHAPTER II
A BRIEF INTRODUCTION TO MACHINE TRANSLATION

The quest for machine translation (MT) seems to be the story of the pursuit of an illusory goal. The task was first seen as a dictionary problem, with aspects of language such as word order being relatively unimportant. "The premise on which this [paper] is based is that syntax is of quite minor importance in understanding a language.... Generally ... the mere sequence of words, without any knowledge of syntax at all, is sufficiently revealing" (Richens and Booth, 1955, p27).

The results of word-for-word translations were soon found to be less than easy to comprehend (even with methods for processing idioms), because frequently there are multiple equivalences in a target language, even if the source language words each have only one meaning. Some syntactic processing was therefore introduced to distinguish different functions of one word form, such as "space vehicles" as opposed to "outer space". The system demonstrated by the Georgetown group on 7 January 1954 used a few syntactic "codes" to move lexical units so as to create a more readable English translation from the Russian. These codes (and other similar methods) were later to be considered less than was needed for an adequate translation. The elaboration of the original Georgetown system was criticized by Martin Kay for although it "purported to be concerned largely with syntax, it incorporated neither the notion of a grammatical rule nor the notion of a syntactic structure" (Kay, 1973, p219).

The next point of rest in the quest was at "syntactic translation", a method much influenced by the theories of Noam Chomsky, and taken up by groups from the University of Texas at Austin, Montreal, Grenoble and further afield. Arguably, most MT
projects have been content to stay here, venturing perhaps a little away from the spot by including some limited semantic processing.

Some have ventured further, claiming that the determination of syntactic structure is not enough to give high-quality machine translation. Bar-Hillel (1960) argued that the correct rendering of a seemingly simple sentence like "the box was in the pen" can only be obtained by recourse to knowledge about the relative sizes of writing pens and play-pens, and of boxes. Such knowledge used to understand a text is available to humans but was not, in 1960, to machines. The responses to this problem have been threefold. Some such as Wilks, have sought to build intelligence and knowledge into their systems. Others, such as the designers of CULT and Weidner, have chosen to move toward a partnership of man and machine, in effect using the computer as a powerful dictionary, and the human as an arbiter and guide. The third approach has been to limit the form and subject matter of the text, as with the TITUS and TAUM-Meteo systems.

Those who come into contact with MT systems have two pressures upon them. On the one hand is the wish to research for the illusive fully-automatic, high-quality translator; while on the other is the demand for practical working systems. After the optimism of the nineteen-fifties (the decade of the large research projects), and the disappointment of the nineteen-sixties (with the presageful black cover of the Automated Language Processing Committee's report that all but stopped research in the United States), MT is undergoing a revival. Whether it is destined to stumble along the way, as it did in the nineteen-sixties, can only be a matter for prophecy and of time. Its basis is now more sound, for it is built on the diffuse experiences of many people, working in a number of research groups, or using one of a number of systems for day-to-day translations. At this moment the EUROTRA project of the CEC seems to represent the nearest that any compromise between development
and research will get to the goal of the quest. It is a measure of how far MT has come that about this project its director has written: "Clearly the result of analysis must be in some way a representation of the text, but giving more detail about its structure (unless, of course, we are concerned with word-to-word translation, where we shall produce perhaps a number of good jokes, but not an adequate translation) (King, 1982, p142 - italics added).

The remainder of this Chapter is given over to a description of the two basic strategies for MT, that is to say the direct and indirect strategies. The latter is divided into the interlingual approach (where two systems will be discussed) and the transfer approach (which will concentrate on the EUROTRA system).

**Direct systems**

The earliest word-for-word systems were naturally restricted to a specific language pair, for instance Russian to English, and were not reversible. Sophistication did not at first imply a change of approach, and nearly all projects until 1966 kept to the strategy of a direct translation from a source to a target language. In conception the direct strategy is the simplest of all (Figure 2.1), although a single flowchart symbol may hide many programs. Size was the characteristic of the Georgetown system, the foremost of the projects examined by the ALPAC report. "Such information about the structure of Russian and English as the program used was built into the very fabric of the program so that each attempt to modify or enhance the capabilities of the system was more difficult and more treacherous than the last" (Kay, 1973, p219).

Although after 1966 attention shifted from the direct systems, a few notable examples have become operational. The Georgetown system was never taken up by its sponsors, although installed by the Atomic Energy Commission at Oak Ridge National
Figure 2.1

Basic structure of a direct machine translation system
Laboratory, and by EURATOM at Ispra. Its influence is to be seen today in the SYSTRAN system, which was originally developed by Peter Toma (a former liaison officer at Georgetown) to translate Russian into English. Because of SYSTRAN's modular system design, it has proved relatively easy to adapt it to other language pairs, most notably English to French, and French to English.

In whatever configuration, SYSTRAN has all the characteristics and drawbacks of the direct systems. It is designed in all details for the translation of just one pair of languages, and all information and processing is tailored to this end. There is no overall linguistic theory, and no more work is done than is necessary for the languages in hand. To put it more firmly, it uses "brute force" rather than theoretical subtlety, relying on ad hoc methods to surmount any difficulties that arise.

Russian texts have to be transliterated at data entry, which may be via one of a number of devices (Figure 2.2). The words of the text are allocated a running number and searched for in a dictionary of frequently occurring words, which also contains the first word of idioms. The unfound words are sorted into alphabetic order, and retrieved from the main stem dictionary. Endings are tested for acceptability and, if successful, the dictionary information (which includes the target language equivalent or equivalents) is attached to the source language words. At the completion of this stage, the texts are sorted back to their original order.

The third stage is analysis, which comprises seven parts. The first of these resolves source language homographs by examining the grammatical categories of adjacent words; so in English one would expect "light" to appear after an adjective or article if it was being used as a noun, or before a noun if it was being used as an adjective. The second part looks for compound nouns (such as "blast furnace") in a "limited semantics
dictionary". Of the remaining parts, all are concerned with recognizing higher syntactic relations such as government and apposition (Toma, 1977, p575; Pigott, 1979, p241).

Transfer is concerned with choosing the correct equivalents in the target language. Where analysis has not rejected all but one alternative, three procedures are available. First, where words may have idiomatic translations under certain conditions, tests are carried out to determine which translation to include. There is much recourse here to the semantic categories of surrounding words. These categories are a number of ad hoc markers that denote certain attributes that have been found useful in translation: there is no attempt to produce a theory of meaning or understanding. So in translating the English "employ" with the CEC version of SYSTRAN, the French "employer" is chosen if the object is marked with "PROF" (ie a profession such as engineer or secretary), otherwise "utiliser" is used. Prepositions are also translated using this semantic information which has been assigned either to words which govern them, or which they govern. So again for English to French, "in" with the name of a subject field such as chemistry (denoted by the marker "SCINO") is translated by "en" rather than "dans". Here in particular, it is easy to see the ad hoc nature of these methods, for the adoption of these markers to say English to German SYSTRAN would not produce comparable results to the English to French version. The third procedure is to use other information of a purely ad hoc nature, centred on testing words in the immediate context. The final stage is synthesis, which is concerned mainly with morphological generation and rearranging word order.

SYSTRAN is currently being used in several versions. The United States Air Force adopted it as a replacement for the FTD Mark II translator, which was one of those that performed poorly for ALPAC. Here the texts are Russian scientific and technical papers, whereas the CEC uses its versions (English to French, and
Figure 2.2
Basic flowchart of the SYSTRAN system
to Italian; and French to English) for a very wide range of subject fields, with post-editing of the texts by translators to overcome the many shortcomings of the system. The Xerox corporation take an opposing position, in that they limit the style of their texts to produce "multinational customized English" (Ruffino, 1982).

A number of evaluations of SYSTRAN have been carried out, and in particular, the CEC commissioned one project in 1976, and a second in 1978. The problem with evaluation is subjectivity, but lack of excellent tools doesn't necessarily mean that a job is not worth doing, or its results are meaningless. In the second evaluation, against an intelligibility of ninety-nine per cent for the original text, the MT system achieved seventy-eight per cent, using a main dictionary of some forty-four thousand items, and more than ten thousand expressions. With post-editing it was possible to raise this intelligibility to the same as for human translation, ninety-eight per cent. Accuracy of MT was seventy-three per cent and style was assessed at seventy-six per cent. The post-editing rate was calculated using three CEC translation services, and it ranged from thirty-one to forty-eight per cent. Finally, it took fifteen minutes to create a dictionary entry for English to French SYSTRAN, including terminological research, linguistic coding and data capture (Van Slype, 1978).

SYSTRAN is the most widely used MT system, but it is not the only operational direct system. Recently the Weidner system has become available; this being interactive and having a number of language pairs, including English to French, to Spanish, and to German (Wyckoff, 1979; Hundt, 1982). The LOGOS system was sponsored by the United States Air Force for the translation of English into Vietnamese, but is now concentrating on the politically more secure target language of French. Other operational systems have included the FTD Mark II translator (referred to above), a very unsophisticated system that relied, for the ability that it had, on large dictionaries mounted on the
"photoscopic store" which was an early method of fast and large storage; and the system for translating United States Patent Office applications into Russian implemented by the Central Research Institute for Patent Information, in Moscow. Work has also proceeded on several systems that never got beyond the experimental stage. As an approach however "the 'brute force' trial-and-error approach of most direct MT systems has been rejected in favour of thorough analysis of linguistic processes and careful design of appropriate and efficient computational procedures" (Hutchins, 1978, p130).

**Interlingual indirect systems**

The idea of an interlingua or universal language has a longer history than has MT. In the memorandum that first breathed life into the field, Warren Weaver (1949) speculated that use could be made of elements that are surely present in all languages; "to descend, from each language, down to the common base of human communication... and then re-emerge by whatever particular route is convenient". Practice has served to prove just how difficult it is to construct a third neutral language that will express all the structures and facilities of two or more languages.

The attraction of the interlingual strategy for an MT project that wishes to translate amongst several pairs of languages is that the number of programs, and therefore work, is significantly reduced. If there are three languages in the system, then six direct systems would be needed. In an interlingual system, three programs would be needed to translate the languages into the interlingua, and another three to translate from the interlingua into the target languages. If a fourth language is added, then another six direct systems would be needed, while in the interlingual system another two programs (one source language and one target language) would be needed (Figure 2.3).
Basic structure of an interligual indirect machine translation system
The difficulties of constructing an interlingua have come to be known mainly through the efforts of two prominent groups of the nineteen-sixties. The Centre d'Etudes pour la Traduction Automatique (CETA) at the University of Grenoble, and the Linguistics Research Center of the University of Texas at Austin both used intermediate languages capable of representing general syntactic structures, but whose lexicon consisted of conjoined lexical units from source and target languages. To put it more simply, sentences with identical meaning but different words did not produce an identical result in the interlingua.

Both of these projects started in 1961, and lasted about ten years. Since then there has been no large scale attempt to produce fully-automatic interlingual MT. The economic attraction still holds, and at about the same time as these projects were running down, the Church of Jesus Christ of Latter-day Saints was starting research at Brigham Young University on an interactive system (now called "ITS") to translate from English into five languages. ITS has been tested, and while it is claimed to produce encouraging results, it does need rather more human intervention than was wished (Melby, Smith and Peterson, 1980).

The other project to start in the early nineteen seventies was the TITUS system for translating abstracts (written according to well-defined rules) about textiles. This is reviewed in detail in Chapter III.

The inspiration for this reaction to the brute-force direct system owes much to Chomsky's theories of transformational grammar, and especially to the view that all languages share common base structures, from which the various surface structures are transformed. Thus in the METALS project of the Texas group, German surface forms were progressively analysed down to one or more acceptable base forms. The process started with morphological analysis, and the retrieval of lexical records from the dictionary. Using this information, together with the
"surface grammar", one or more tentative standard strings were produced. So "Er nahm von diesem Plan Abstand" would have its discontinuous elements brought together to become "Er nahm Abstand von diesem Plan". These standard strings were then tested for well-formedness by the application of the "standard grammar", and for each accepted, a phrase structure was built. Finally in the analysis, the "normal form grammar" filtered each standard tree to produce "semantically well-formed" normal forms. "Semantically well-formed" refers not to a thorough going semantic analysis, but to particular predicates requiring certain semantic features to be present in specified arguments. Transfer consisted of switching the target language lexical units for source language units, and from there, the surface structure was created by a number of transformations, essentially in reverse order to the analysis of the source language (Hutchins, 1982; Locke, 1975).

The problems with METALS seem to have stemmed from two underlying limitations. The first was the inadequacy of the context-free parser that was used. While a parser which can produce all possible structures for an input must be a reassuring thing to have, its drawbacks lie in being capable of producing many false or unlikely readings that have to be rejected by other, later stages in the analysis. While a single surface form could produce several normal forms; equally, because synthesis was the reverse of analysis, one normal form could produce a number of surface forms. This might be an advantage for automatic indexing or abstracting, but for MT it serves to confuse the user. The second deficiency of METALS would have affected the first problem. Because no account was taken of the meaning of a text, rather than of the sentence currently being processed, multiple readings were left in that could have been filtered out by intersentential or discourse semantics. This lack of semantic processing has been characteristic of almost all MT projects to a greater or slightly less-than-great extent.
A lack of semantics is not a criticism that could be fairly made of the English to French translator of Yorick Wilks. The research behind this system did not have the objective of producing an operational system, but to investigate one artificial intelligence approach to natural language understanding. The function of the system is less important than the principles that it embodies, but having said that, the translation into French provides a good test of the success of the system, for the generated language either is or is not correct. The account here presented is of the system as it seems to have stood in about 1975 (Wilks, 1975a; 1975b; 1976). It has undergone some theoretical development since then (Wilks, 1979) but this is not crucial to a general account of the system.

An artificial intelligence approach to any form of natural language processing is distinguished by an attempt to "understand" the data being processed. Furthermore "what almost all [artificial intelligence] programs have in common ... is strong emphasis on the role of knowledge in understanding, and on the presentation of a theory as a possible process" (Wilks, 1977, p698). In the case of Wilks's translator, the theory is that of "preference semantics" (Wilks, 1973), which essentially states that when interpreting natural language, the most suitable available interpretation should be taken. This is set against the view that deduction can be used in such processing to determine whether or not a certain sequence of words is allowable and therefore meaningful.

The system eschewed overt syntax, relying instead on semantic representations of word senses. The smallest units were the semantic primitives, being a list of about seventy basic concepts. These were combined to produce formulas, one for each sense of a word. The primitives thus formed into formulas constitute, in effect, an interlingua, and that the primitives are based on English is inconsequential. "Crook" in the sense of someone who does criminal acts was represented as:
While the sense of a shepherd's crook for manipulating animals was given the formula:

$$(((\text{NOTGOOD ACT}) \text{OBJE} \text{DO})(\text{SUBJ MAN}))$$

Actions had formulas of a slightly different kind. Apart from the primitive that described the "type" of act, there was a case frame. Case grammar started as an extension of transformational grammar and at its simplest, describes a small number of roles (cases) that arguments can play in association with verbal elements. A case frame for an individual verb or action describes the allowable cases together usually with some information about their semantic nature. So the formula for "interrogates" was:

$$(((\text{THIS BEAST}) \text{OBJE} \text{FORCE})(\text{SUBJ MAN}) \text{POSS})(\text{LINE THING}))$$

which may be paraphrased as; one person forcing another to communicate information.

Texts of small paragraph length were accepted (Wilks, 1976), with the first procedure being the retrieval of the word senses. The text was then fragmented into manageable units according to a list of key words, and to some extent, the formulas (Figure 2.4). In the first of the matching processes, each of these sub-units were assigned one or more "templates", being short agent-action-object structures, representing a basic message. For the sentence, "The policeman interrogated the crook", two underlying templates would be matched:

$$\text{MAN FORCE MAN}$$

for "crook" as a criminal, and for the shepherd's crook:

$$\text{MAN FORCE THING}$$

It was the function of the "preferential expansion" to attempt to reject all but one interpretation for each sub-unit. The process involved examining the parts of the formulas to evaluate how well preferences were matched. The formula for "interrogates" preferred a human actor ("SUBJ") and a human patient ("OBJE"), thus giving the preferred reading of "crook" as a criminal a greater semantic density. It is this template that
is carried over to the next stage of processing.

A semantic block to represent the whole of the text had to be created before the target language text could be generated. The sentence used so far is artificially simple, and a more likely sentence would be: "He left Loughborough by motorway", which would have been allocated two sets of templates and which would have been joined by a "paraplate". A paraplate consisted of six slots (three for each template), each having to be satisfied. Essentially each paraplate represented a case grammar relationship, and was therefore stored under the key of an English preposition, and sub-arranged in a pre-defined order so as to give the preferred interpretation first.

Following on from paraplate matching it was often possible to resolve remaining pronouns. In the sentence: "John bought some wine, sat on a rock, and drank it", a search was made across the semantic structure so far built up, to ascertain that "drink" preferred a liquid ("FLOW STUFF"), and thus the interpretation that John drank the wine had a greater semantic density than that of John drinking the rock.

In a number of cases this amount of processing would not have been sufficient and the extended mode of pronoun resolution would have to be invoked. Amongst the best examples is: "The soldiers fired at the women and we saw several of them fall". The difficulty in the interpretation of "them" stems from the necessary information not being present in the sentence, but being supplied by the hearer. This is the same problem that caused Bar-Hillel to doubt the feasibility of fully automatic, high-quality translation, as was noted above. Wilks resolved this type of problem sentence in two stages. The first was in effect to supply some knowledge of the world by extracting from a store of template-like structures, some new information judged to be relevant from a comparison with the information already in the semantic block. Then a set of common-sense rules could be
Figure 2.4
Diagram of Wilks's English to French translator
invoked, which took the information in the block and tried to infer a solution to the anaphora. For the current example, the appropriate rule would have stated that when a first person strikes a second, then the second person falls. After this process, the remaining ambiguities had been resolved, and the French text could be generated (Herskovits, 1973).

The main drawback of this system lay in the uneconomy of using primitives. Apart from the difficulties and vagueness in writing a formula, there was much duplicated information in the semantic blocks. The extension of the dictionaries to a more operational size (from about six hundred entries) might produce unlooked for problems. With a larger vocabulary would have to go more common-sense rules, and it might prove difficult to control the practicality of their application.

Transfer indirect systems

The second indirect type of system represents a feasible middle way between the economies of the interlingual approach and the uneconomies of the direct strategy. It separates the processing into three distinct areas; analysis, transfer, and synthesis (Figure 2.5). It can be argued that so do most modern direct systems, if only because of modularity in their programming. The difference between the two strategies is that in a transfer system the amount of processing at any one stage is not restricted to just enough to get acceptable results in a given target language, but is taken to a near abstractly defined level, irrespective of a particular target language needing that amount of analysis. Most transfer MT projects have as their aim the translation into more than one language, and so a rough and ready rule is that analysis should be as deep as is necessary for the most difficult language pair. The economy of the transfer over the direct approach lies in fewer programs having to be written. For each language in the system, there has to be an analysis and a synthesis module, and for each source to target
Figure 2.5

Basic structure of a transfer machine translation system
pair, there has to be a transfer module. So a system with three languages would have six language pairs and six transfer modules.

The idea of the transfer approach can be traced back to a paper by Yngve (1957) which proposed essentially syntactic translation. Since then a number of projects have adopted the approach, with varying degrees of success. Amongst those that have never got beyond one language pair have been the POLA Chinese to English project at the University of California at Berkeley; and the TAUM (Traduction Automatique de l'Université de Montréal) project, which has worked with English to French from 1962. Its work bore fruit on the production of the TAUM-Meteo system for translating weather reports for practical use. With a success rate of about eighty per cent, it indicates the results to be gained from using a limited subject field and type of text. This project has also produced TAUM-Aviation, a system designed for the translation of American aeronautics manuals. In Europe the most impressive project has been the continuation of CETA who, on abandoning the interlingual approach, renamed themselves GETA (Groupe d'Etudes pour la Traduction Automatique). They have concentrated on producing a few powerful algorithms which are capable of calling and using a number of sub-grammars to control translation from Russian to English (Boitet and Nedobejkine, 1981). This project has been willing to share its experience very freely and has notably influenced the SUSY (Saarbrücken Übersetzungs System) project at the University of Saarbrücken who have worked on Russian to German translation, with some more limited work on other language pairs (Maas, 1977).

This considerable experience provides the background to the CEC's EUROTRA project. The need for an MT system arises from the number of official languages in the European Community (all of which are accorded equal status) and from the volume of routine documents that therefore have to be translated. The Commission acquired SYSTRAN, but this was viewed as an interim measure, for in February 1978, work started on specifications for EUROTRA. The
objectives are to produce a prototype by the beginning of 1987, capable of translating limited categories of documents, in a limited subject field (Council of the CEC, 1981). The transfer approach has been adopted, which has enabled the work on monolingual modules (ie analysis and synthesis) to be distributed to individual centres in Community countries. The transfer modules will be created by the two appropriate national centres working together, and the whole project is subject to centralized control.

As far as possible, flexibility has been built into the specifications. Each centre may pursue its own approach to both analysis or synthesis, providing that the input to and output from the modules is in accord with the "interface structure" laid down as the project's communication standard. Essentially, this structure is a tree on which information about the text can be held (Maegaard and Ruus, 1980). There is a minimum amount of information that must be provided on the tree. After the lexical units themselves, their morphological characteristics are recorded, together with the morpho-syntactic class (eg finite verb); the surface syntactic function (eg subject); and valency information. This last is akin to case grammar in that it describes the relationships which predicates have with their arguments and how "close" those relationships are (King, 1982, p143). The representation of so much information is rarely as straightforward as this account might suggest (King and Perschke, 1982, p29-30).

A necessary part of any system is of course the dictionaries. Little information has been generally released on EUROTRA as yet, and in this respect they have received the least attention. It seems that they are to be held as one large database, with software written for control and monitoring (Knowles, 1982). Clearly there would have to be some information appropriate to creating the interface structures, and there must be some additional information for the transfer modules to allow
tests to be performed on parts of the current tree to decide which of the alternatives should be chosen.

In connection with EUROTRA there has been no reference to programs, but instead to modules. It is a measure of how far MT has progressed that the programs for EUROTRA are to be written by software houses, and the national centres will be provided with what is, in effect, a very high level language in which to describe the processes to be performed. The statements of this language will be cast in the form of "production rules" (first made famous in the MYCIN system for computerized intelligent medical diagnosis) which state on the left-hand side one or more conditions and on the right-hand side of the rule the action to be performed if the condition side fits. The various modules will therefore consist of sets of production rules which will be interpreted by the programs against the texts that are entered (King and Perschke, 1982).

The success of EUROTRA will probably rest on two factors. The first is just how much post-editing will be necessary, and the second is how long it will take to enter all the necessary information into the dictionaries. As regards the first, there is some difference of opinion as to whether or not it has been designed as a machine-aided system. The proposal to the Council of the Commission states that it is, and this is presumably what it will be. Certainly, efforts are being made to integrate the MT system into more generalized text processing systems. SYSTRAN, with all its failings, seems not to have been popular as a pre-translator with the CEC translators: EUROTRA may have its work cut out to convert more than the naturally enthusiastic. As for dictionary creation, a study of the TAUM-Aviation system (that only has English to French) showed that an entry took three and three quarter hours to produce, at a cost of about twenty-three pounds (Van Slype, 1982). Economic judgements must be relative, and a cost like this may be a better price to pay than the salary of a translator from, for instance, Danish to Greek.
In conclusion

MT seems to be flourishing, but is it in danger of another set-back such as the ALPAC report? The answer to that as far as the European Community countries are concerned is that it probably depends on EUROTRA. The outlook is different than it was in 1966, for automation, including office automation, is gathering pace, and MT is only a part of this. Ideas and techniques have flowed in from other disciplines. Today it is virtually unthinkable that anyone should write a computational linguistics program that completely includes its rules as parts of the program, rather than as data. There are more powerful techniques, such as Woods's searching procedure, the augmented transition network. The theories of case and valency grammar have both come from descriptive linguistics. Most importantly, the demand is there, for SYSTRAN and Weidner have sold, and TAUM-Meteo is working to its masters' satisfaction. "There is now a mood of quiet optimism in MT research; it is a mood which should not be lightly dismissed" (Hutchins, 1978, p150).
<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Project leader</th>
<th>Languages</th>
<th>Strategy</th>
<th>Research dates</th>
<th>Operational</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Georgetown</td>
<td>Georgetown U</td>
<td>Dostert</td>
<td>Rus-Eng</td>
<td>Direct</td>
<td>1952-63</td>
<td>1962</td>
<td>AEC (1962-)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Euratom (1965-75)</td>
</tr>
<tr>
<td></td>
<td>(1968-)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EURATOM (1976-)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Eng-Fre</td>
<td>Direct</td>
<td></td>
<td></td>
<td>CEC (1976-)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fre-Eng</td>
<td>Direct</td>
<td></td>
<td></td>
<td>Xerox (?)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Eng-Ital</td>
<td>Direct</td>
<td></td>
<td></td>
<td>CEC (1978-)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Eng-Span</td>
<td>Direct</td>
<td></td>
<td></td>
<td>Xerox (?)</td>
</tr>
<tr>
<td>FTD II</td>
<td>IBM Yorktown</td>
<td></td>
<td>Eng-Fre</td>
<td>Direct</td>
<td>1978</td>
<td>1978</td>
<td>Mitel (1980-)</td>
</tr>
<tr>
<td></td>
<td>Heights</td>
<td></td>
<td>Eng-Span</td>
<td>(inter-</td>
<td></td>
<td></td>
<td>Mitel (1980-)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Eng-Ger</td>
<td>active</td>
<td></td>
<td></td>
<td>Mitel (1981-)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Chin-Eng</td>
<td>Direct (int-</td>
<td>1975</td>
<td></td>
<td>Acta Mathematica</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1968-</td>
<td></td>
<td></td>
<td>Sinaica (1975-)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>eractive</td>
<td></td>
<td></td>
<td>USAF (1971-73)</td>
</tr>
<tr>
<td>Weidner</td>
<td>Weidner Corp</td>
<td></td>
<td>Eng-Viet</td>
<td>Direct</td>
<td>1964-9</td>
<td>1971</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Eng-Fre</td>
<td>Direct</td>
<td>1975-9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Eng-Fre</td>
<td>Direct</td>
<td>1964</td>
<td>1965-75</td>
<td>Central Research</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rus-Eng</td>
<td>Direct</td>
<td>1964</td>
<td></td>
<td>Institute for Patent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Information</td>
</tr>
<tr>
<td>CULT</td>
<td>Hong Kong</td>
<td>Loh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chinese U</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Logos Corp</td>
<td>Scott</td>
<td>Eng-Viet</td>
<td>Direct</td>
<td>1964-9</td>
<td>1971</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1970-)</td>
<td></td>
<td>Eng-Fre</td>
<td>Direct</td>
<td>1975-9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moscow Patent</td>
<td></td>
<td></td>
<td>Eng-Fre</td>
<td>Direct</td>
<td>1964</td>
<td>1965-75</td>
<td>Central Research</td>
</tr>
<tr>
<td>Office</td>
<td></td>
<td></td>
<td>Rus-Eng</td>
<td>Direct</td>
<td>1964</td>
<td></td>
<td>Institute for Patent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Information</td>
</tr>
</tbody>
</table>

Table 2.1

Machine translation projects and systems (adapted from Hutchins, 1982)
<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Project leader</th>
<th>Languages</th>
<th>Strategy</th>
<th>Research dates</th>
<th>Operational dates</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>CETA</td>
<td>Grenoble U</td>
<td>Vauquois</td>
<td>Rus-Fre</td>
<td>Interlingual</td>
<td>1961-71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITS</td>
<td>Brigham Young U</td>
<td>Lytle</td>
<td>Eng-Span,Port, Ger, Fre, Chin</td>
<td>Interlingual</td>
<td>1971-80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wilks</td>
<td>Stanford U</td>
<td>Wilks</td>
<td>Eng-Fre</td>
<td>Interlingual</td>
<td>1971-75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POLA</td>
<td>California U</td>
<td>Wang</td>
<td>Chin-Eng</td>
<td>Transfer</td>
<td>1967-75</td>
<td>1975-76</td>
<td>Bureau des Traducteurs</td>
</tr>
<tr>
<td>TAUM-Meteo</td>
<td>Montreal U</td>
<td>Chandioux</td>
<td>Eng-Fre</td>
<td>Transfer (limited)</td>
<td>1972-76</td>
<td>1976</td>
<td>Canadian Air Force</td>
</tr>
<tr>
<td>TAUM-Aviation</td>
<td>Montreal U</td>
<td>Boitét</td>
<td>Eng-Fre</td>
<td>Transfer</td>
<td>1976-79</td>
<td>1979?</td>
<td></td>
</tr>
<tr>
<td>GETA</td>
<td>Grenoble U</td>
<td>Boitét</td>
<td>Rus-Fre</td>
<td>Transfer</td>
<td>1971-72</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSY</td>
<td>Saarbrücken U</td>
<td>Eggers,Maas</td>
<td>Rus-Ger</td>
<td>Transfer</td>
<td>1967-78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EUROTRA</td>
<td>CEC</td>
<td>King</td>
<td>Eng,Fre,Ger, Ital, Dan, Dutch Greek</td>
<td>Transfer</td>
<td>1978-87</td>
<td>[1987]</td>
<td>CEC</td>
</tr>
</tbody>
</table>

Table 2.1 (continued)

Machine translation projects and systems (adapted from Hutchins, 1982)
References


In


In


In


In


In


In Coling 80: proceedings of the 8th International Conference on Computational Linguistics, Tokyo, 30 September–4 October 1980. [s.l]: [s.n]. p424-429.


In

Toma, P.P. (1977) SYSTRAN as a multilingual machine translation system.


Wilks, Y. (1975) A preferential, pattern-seeking, semantics for

Wilks, Y. (1976) Parsing English II.


CHAPTER III
TRAITEMENT DE L'INFORMATION TEXTILE UNIVERSELLE ET SELECTIVE
(TITUS)

Since 1945, there has been a growth in the amount of literature published, especially in the scientific and technical fields. The range of languages used in this literature explosion means that a worker in a field has to face a larger number of documents in a language other than his native one. (This does not imply that the proportion of documents published that are in the non-native language has necessarily increased). For the organization of a monolingual document retrieval service, a proliferation of languages necessitates the employment of subject area specialists with some ability in a foreign language to analyse material, which may, because of its provenance, be quite difficult to acquire.

One solution to these difficulties of acquisition and analysis is manifested in a number of international co-operative networks, such as the International Road Research Documentation (IRRD) network. Here a number of OECD member countries have developed a multilingual information retrieval thesaurus, which takes the form of English, French and German thesauri, from one of which terms are drawn by the indexer to describe the subject matter of the document. When this bibliographic data is entered into the data base, the thesaurus descriptor terms are translated into a pivot language. The essential element in this processing of the subject information is that the constructors of the thesaurus have ensured that there is a direct equivalence of terms between all three languages; hence the pivot file records are able to stand for the descriptors. At the stage of searching, the descriptors that the searcher uses are translated into their pivot language equivalents, and these are used to search the
inverted files to gain access to the bibliographic descriptions. From the point of view of acquiring the material for inclusion in the retrieval system, each member country of such a network is responsible for the acquisition and processing of its own literature (Mongar, 1968). A similar situation holds for the International Labour Organization.

Some on-line data base providers (as opposed to their producers) have incorporated multilingual software into their retrieval systems, these routines working in a similar way to the procedure previously described. This approach has grown up because the large bibliographic data bases such as Medline are in English, but are of interest to, for instance German speakers, who can access it in their native language through DIMDI; or its offshoot, Cancernet, of interest to the French speakers who have access via the SABIR system.

The same problems of language and acquisition were felt by the textile industries, and indeed, due to the competitive nature of the industry and the growth of competition outside the Western world, their problems seem to have been more pressing. The "Traitement de l'Information Textile Universelle et Selective" (TITUS) was born out of this need, and owes much to the initiative of one of the foremost textile information centres in Europe, the Institut Textile de France. It has gone through several stages, starting with an original co-operative venture of the type described with reference the IRRD network. At this stage, which lasted from 1969 to 1973, the textile information centres of Britain, France, Germany, Italy and Spain contributed, although finance caused Italy's later withdrawal. It was during this period that the American textile thesaurus of S. Backer was translated into the network's members' languages, with a final total of about ten thousand key words per language. (Backer and Valko, 1969; Ducrot, 1973b).

As it stood, TITUS I translated only the descriptors, and
the user was presented with a foreign language abstract, unless he was either French (for all abstracts were translated into French) or the document was contributed by his language centre. Experience proved the foreign language abstracts provided were a barrier to the user. Added to this was the problem of different levels of analysis that prevailed in the different contributing centres. What seemed to be needed was a system that would compel all the indexers to apply descriptors with a comparable consistency, and also a method for overcoming the barrier of the foreign language abstracts.

The solution proposed and implemented in TITUS II had intellectual predecessors in several respects. The idea of an interlingua has been traced to some extent in the previous Chapter, together with pre-editing. The use of translated abstracts for documentation purposes had been suggested in a related form by J.W. Perry in 1959 and is reviewed in the note to this chapter. At the time of design of TITUS II, the problems of translating even such a limited text as a natural language abstract or title were considered too many to be attempted. Therefore, titles of documents were translated manually, while the abstract was coded in a "Common Documentary Language".

The structure of a TITUS II abstract seems to be more difficult to describe than it would be to encode. At the heart of the system was the thesaurus previously built. To this was added grammatical information, forming four language lexicons, and a pivot language. To these were added "tool words", which were simply those words or phrases which are necessary to create well-formed phrases, but which have no value in the retrieval of documents. The latter group included transitive verbs and the auxiliary verbs "to be" and "to have", adjectives, some verbal nouns, and others. The third group of words were coded words, which included articles, prepositions and prepositional locutions, conjunctions, and word endings. They were called "coded" because the abstractor did not write these units as such,
but represented them as codes.

This then represented the lexicon of TITUS II. To this restricted vocabulary was added a syntagmatic element that allowed the combination of lexical units. This syntagmatic organization was on two levels, the first being concerned with the linking of phrases, and the second with the structure within phrases.

At the first level, three main types of sentence were allowed: noun phrases connected by one or more prepositions; noun phrases connected by one verb and one or more prepositions; and an enumerative phrase followed by numerous single-descriptor phrases (Figure 3.1). (At this point it should be noted that in TITUS II parlance, a phrase is referred to as a syntagmatic group). When an abstract was written, the sentences were cast in a coded form, which consisted of a "syntagmatic code" which indicated the type of sentence; an "actant" which indicated the relationship (if appropriate) between the last two syntagmatic groups; and the sentence proper.

The second level of syntagmatic organization was coded by inserting codes around lexical units to indicate their relation to the previous and following units, and their morphological behaviour. Thus "wool fibres" was written as "WOOL FIBER+"; "the felting" as "*FELTING"; and the noun complement phrase "the felting minimization" as "*FELTING/*MINIMIZATION". The organization of the lexical unit can be summarized as:

1 Article code (taking definite or indefinite article, or no article).
2 Lexical unit.
3 Number code (plural or singular).
4 Liaison code (how the syntagmatic group is related to the next).
Noun phrases connected by one or more prepositions.

GRAFTING OF WOOL FIBER
DYEING PROCESS
CHEMICAL REACTION OF COPOLYMERIZATION IN SOLVENT MEDIUM
OPTIMAL REACTION FOR THE GRAFTING
GRAFTING OF WOOL FIBER WITH THE METHYL METHACRYLATE FOR THE
PURPOSE OF THE FELTING MINIMIZATION
GRAFTING INITIATION BY MEANS OF A MIXTURE OF BENZYL CHLORIDE
AND ANILINE METHYL COMPOUND

Noun phrases connected by one verb and one or more prepositions

BONDED FIBER FABRIC POSSESSES A HIGH VALUE OF ELASTICITY AND
CRUSH RESISTANCE

Enumerative phrase

ARE CITED: } followed by
descriptors separated
ARE PARTICULARLY MENTIONED: by commas.
ARE PARTICULARLY EXAMINED:

Figure 3.1
Sentence types allowed in TITUS II
Drawing together the levels of syntagmatic organization, a complete TITUS II sentence as coded for input looked as follows:

<table>
<thead>
<tr>
<th>Code ant</th>
<th>SG1</th>
<th>SG2</th>
<th>SG3</th>
<th>SG4 (cont)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-0</td>
<td>4</td>
<td>GRAFTING, WOOL FIBER+, METHYL METHACRYLATE, *FELTING/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SG4 (cont.)</td>
<td></td>
<td>*MINIMIZATION</td>
</tr>
</tbody>
</table>

("SG" stands for "syntagmatic group").

At output, this would be translated from the pivot language into: "GRAFTING OF WOOL FIBERS WITH THE METHYL METHACRYLATE FOR THE PURPOSE OF THE FELTING MINIMIZATION"

Once the abstract had been recorded together with the bibliographic information, the completed input sheets were entered into the data base at either Paris or the German centre at Düsseldorf. If the abstracts were allowed through the verification procedures, they were converted into the interlingua, which was (and still is) in number form, and thereby very economical of storage space. By a process of translation from the pivot language, the four target languages could be created. Few details of the language generation routines are available. It may be reasonably supposed that it was centred around the lexicon entries of the nouns in the system. The record for each noun or noun phrase noted various characteristics, such as the prepositions that normally served to convey relationships, the singular and plural forms, and whether or not a definite article was customary before the unit under consideration. From this information, what was virtually a code-matching was undertaken to produce the target text. (Ducrot, 1973a).

There has been no thorough investigation of the performance
of the TITUS II translation modules reported, but A.M.N. Barnes of the Centre for Computational Linguistics, UMIST, is investigating the syntactic structures used by abstractors, and the relation of these findings to the constraints of the TITUS system in order to establish a freer grammar for use in the context of machine translation. Bruderer commented that the quality was relatively good, but that the word order occasionally left something to be desired ("die Wortstellung lässt manchmal zu wünschen übrig" (1978, p141)). As regards cost, it seems that income from the sale of the documentation service covered only half of the expenditure on the development of programs and lexicons and the running of the system (Ducrot, 1974). The use of such a constrained syntax damaged abstracting costs in two ways. First, suitable staff had to be recruited, trained and maintained. After a period of six months, by which time a plateau of productivity was reached (Ducrot, 1974), the abstractors took some one and a half times as long to write a TITUS II abstract as they would an ordinary "natural language" one (Dubois, 1979). Unsurprisingly, the system was never taken up by other agencies.

The inefficiencies in coding of abstracts was realized at an early stage, and plans were laid for a more economic method of getting similar results. There was to have been a TITUS III, but of this no details were published and it seems not to have been implemented. In 1979, a paper was published which gave the bare outlines of the TITUS IV system (Streiff, 1979), and fuller details were reported in May 1982 (Ducrot, forthcoming). From this published information it is possible to piece together an account of the purpose and working of this less constrained version of the system.

As many of the cost problems of TITUS II were traceable directly to the coding needed for the abstracts to be translatable, the aim was to produce a version that would accept uncoded text. This is not to say that the aim was to be able to accept any text, but those in a particular subject field, and
written according to rigorous rules of syntax. This is the reverse of the Meteo system, which aims to translate texts that conform to standards devised for purposes other than machine translation. Computing costs and technological developments have allowed the creation of a system that will accept data added to the data base on line, and processing speeds are fast enough to allow acceptable speeds of syntactic analysis and verification.

The system is still interlingual, and data for all four language lexicons are created at the same time. The first part of the input deals with information applicable to all languages, which includes amongst other things, the type of item, and the inherent and implied semantic features. The rest of the record consists of separate areas for the individual languages. For a French substantive, the information required is at least:

Surface gender
Existence of singular and plural forms
Singular form (if appropriate)
Plural form (if appropriate)
Elision of the definite article
Scope note
Possible polysemy

This is of course, surface syntactic information, and one is justified in assuming the information for English, Spanish and German is no different.

As with TITUS II there are two levels of syntagmatic organization. The first level is again the combination of groups into a sentence or "proposition" in TITUS IV terminology. The structure is shown as a network in Figure 3.2. It should be noted that there is no possibility of infinite looping, for when a group is repeatable, a maximum of three repetitions are allowed, the groups being labelled as first, second and third.
There are two main types of group allowed; the nominal group and the verbal group. The organization within these represents the second level of syntagmatic organization, and will be presented below. For ease of comparison with published accounts, the French abbreviations have been retained.

GNS. Subject noun group. (Obligatory). Contains at its heart either one substantive lexical unit or a personal pronoun.

GNC1, GNC2, GNC3. Circumstantial noun groups. (Optional). These always start with a preposition or prepositional locution.

GV. Verb group. (Optional). Contains a verb with a simple or complex construction, in a variety of persons, tenses, and voices.

GVC. Verb group complement. (Optional). Contains at least one verb in the infinitive which complements the verb of the previous verb group.

GNV1. Complementive noun group 1. (Optional). Contains at least one substantive lexical unit or an attributive adjective. The group represents one or more complements of the verb.

GNV2. Complementive noun group 2. (Optional). Formed as GNV1. It represents the complement(s) of attribution (or the agent of the passive verb) of the verb of the proposition.

GNP1, GNP2, GNP3. Prepositional noun groups. (Optional). Follows the same rules as GNC1, GNC2, and GNC3. Contains the prepositional complements of the verb.

The way in which a sentence is split according to these rules is as follows:
The recent progress of the software suitable for the online treatment of information systems has favored the development of data bases in all the fields of science and techniques.

At the second level of organization, the nominal groups are made up of one or more sub-noun groups (SN) (Figure 3.3). This contains:

- A preposition
- A determiner
- One or two simple adjectives placed before or after the noun, depending on language usage
- A conjunction placed between two consecutive adjectives
- One or two appositions
- One adjective with complement (introducing another SN group which complements the adjective)

All nominal groups may contain from one to fifteen of these sub-nominal groups, bound together by the grammatical relations of: subject, noun complement, adjective complement, and complement of comparison.

The other group at this level, the verb group is shown in Figure 3.4. The verb may take all the normal conjugations, but in the working version it is restricted to the third person singular or plural.

It seems clear that TITUS IV is almost, if not completely, syntax-oriented. The inherent and implied semantic features of the substantive lexical record have not been explained in the literature available. No reference has been found to a level of semantic processing at either analysis or synthesis, although the
Figure 3.2

Structure of a TITUS IV proposition
Figure 3.3
Structure of the TITUS IV sub-noun group
Simple verb or auxiliary

Preceding negation

Simple verb or auxiliary

Following negation

Adverb

Past participle

Figure 3.4
Structure of the TITUS IV verb group
features might be used in some code-matching type of processing.

It is said that TITUS IV, version A is operational. It allows the abstractor to enter his text, and to edit it should it fail to satisfy the rules of syntax. It is at this stage that polysemic words may be resolved, in the fashion of a machine-aided translation system. Once accepted, the abstract is converted from its natural language form into the pivot language for storage. It should be assumed that the interlingua is very much the same as that of TITUS II, and thus the generation routines are much the same as in that system.

Ducrot (forthcoming) claims for the system a shorter training period (some two or three days), and a twenty per cent increase in time to prepare a TITUS IV abstract over the time taken to prepare a "natural language" one. The time taken for verification, acceptance and translation into the pivot language is two and a half seconds of CPU time, using a time-shared IBM 4331-2 machine. Another second of CPU time is needed to find and translate an entry into one of the four languages. As for quality of translation, this probably remains much the same as for TITUS II (Figure 3.5). As yet, no organization has seen fit to use it for any field other than textiles.
The recent progress of the software suitable for the on-line treatment of information systems has favored the development of data bases in all the fields of sciences and techniques.

Next Step Command =>

Figure 3.5

Output from TITUS IV (from Ducrot (forthcoming))
Note on J.W. Perry's suggested use of translated abstracts

During the nineteen-fifties, a system of "semantic factoring" was developed at Case Western Reserve University. The aim was to break down the subject of a document into its fundamental concepts (the semantic factors), of which there were to be only a limited number. Concepts, such as "thermometer" could be specified by coding them to show their semantic factors together with a specific concept. Using a complex set of roles and links, the concepts so coded could be strung together to form, in effect, an abstract. As a technique, it had several difficulties, including where to draw the analysis to an end (heat is equivalent to the movement of molecules); what to do about concepts that could be specified by use of not all of the applicable factors; and is it really possible to list all the attributes of a concept so as to aid retrieval. To enlarge on the last objection: mercury in a thermometer has associations with mercury in a barometer; should the code for "barometer" be added to a document dealing with mercury thermometers, given that it may be of interest to seekers of information on barometers? (Foskett, 1977, p60-61).

Perry's suggestion took for a starting point the amount of foreign language material that might be of interest to the researcher. Not much of this would have been worth translating unless actually required. On the other hand, a need was unlikely to be expressed unless the subject matter could be brought to the attention of the researcher in a suitable form. The suggestion was to use machine translation techniques to convert the abstracts into a semantic-factored form, and at retrieval to use thesauri which converted concepts in a given natural language into the factored form with which to search the data bases so created (Perry, 1961).

Perry's suggestion was never implemented. The system of semantic factoring has remained in textbooks of information
retrieval as a curious warning to enthusiasts. It is a measure of both the progress in the field, and of the relative ambition of the designers of TITUS that in the period of twenty years, they could have designed a successful system of somewhat more complexity.
References


CHAPTER IV
AN INTRODUCTION TO PRECIS

The remainder of this thesis is concerned with the translation of string indexing languages and more particularly the PRECIS language. This Chapter is intended as an introduction and because it presents neither new matter nor an original view of the language, it may safely be passed over by the cognoscenti.

PRECIS grew out of two trends in information retrieval in the nineteen-sixties: classification research, and computerization. As regards the former, the view had long been held that library classifications were (and still are) inadequate for retrieval as opposed to ordering books on shelves. The trend of research and in particular the Classification Research Group, was toward systems comparable to a child's Meccano set. The classification took the form of the nuts and bolts (the "role operators") and the parts (the "concepts" or "terms", such as "redness" and "stars"). The terms could be joined together using the role operators which were symbols which were interposed between two concepts and indicated the nature of their relationship.

As regards the second trend, libraries at first computerized the traditional manual routines and methods. So a computerized university library catalogue of the early nineteen-sixties held no more nor less information than the typewritten equivalent, and had the same entry points. The Library of Congress investigated the procedures for the storage of bibliographic data on computer, and produced the "Machine Readable Catalog" format, usually referred to as "MARC". Various national bibliographic centres have implemented similar formats, so there exists a family of MARC formats, of which UNIMARC represents an international
standard from which national formats may be derived. Because of
the degree of standardization it is possible for one national
centre to take MARC tapes of another centre's records and "peel-
off" those individual records it wishes to use for its own
purposes. The MARC format is therefore a tool for the
international exchange of bibliographic information.

The United Kingdom's national bibliographic centre has the
task of producing a weekly list of all new books produced in this
country, and it is published as the British National Bibliography
(BNB). At one time the BNB was printed from typewritten copy,
while other cataloguers of the same institution were creating
data for recording in MARC format for exchange with the Library
of Congress. It was only a matter of time before the computer-
held data was used in the production of the printed list.

The BNB has a classified sequence and an author and title,
and subject indexes. Before computerization the latter was a
chain index, but it proved difficult to automate this method and
this, added to technical disadvantages, led to the formation of a
project to devise a new indexing system that would meet the needs
of a printed publication produced from computer-held data. The
main research worker was Derek Austin, and the result was of
course PRECIS.

Viewed with the advantages of hindsight, the objectives of
this project were sixfold (Austin, 1982, p8; 1976, p4):
1) The indexer should have to write only a basic statement of the
subject, from which the computer could form one or more entries
which would be automatically filed and printed.
2) The string should be co-extensive (ie a complete statement of
the subject of the document), and all entries manipulated from it
should be similarly co-extensive.
3) The entries should be natural to the user, which in practice
means that the language used is close to natural language
(instead of librarians' language such as "vehicles, space") and
relationships that aren't explicit being made so by the use of natural language devices such as prepositions rather than a neutral set of symbols.

4) To ensure consistency amongst indexers and thereby that entries under a particular heading should be sub-filed in a consistent way, there should be a set of rules for organizing a string into a set order. These rules should be applicable to all subject-fields and media.

5) The concepts in the entries should be represented by words drawn from natural language, which is controlled in the sense of preferring particular forms of the word and preferring one term to synonymous ones. The vocabulary should be open-ended in that new concepts that are bound to arise are added to the thesaurus as and when needed.

6) The terms that are chosen as entry points should be supported by a thesaurus which would provide see and see also references in the printed index.

The initial product of the research was PRECIS I. This is no longer used and, as this is not an historical account, will be passed over. Further work and the practical experience of indexers led to the second version, which is now generally referred to just as PRECIS. It seems likely that a reprogramming of the British Library's computer facilities will be the catalyst for a third version. PRECIS has remained stable since the publication of the authoritative PRECIS: a manual of concept analysis and subject indexing (Austin, 1974) (hereafter referred to as the Manual), but several improvements have been drafted and published. The implementation of PRECIS used for this research includes some of these, in particular the new "differencing codes" (Austin, 1982, p216). For the sake of consistency, where examples drawn from other authorities would have meant a clash, they have been adapted to the account presented in this Chapter.

For present purposes, this account is more concerned with the entries produced from the original strings than the creation
of the strings themselves. Essential to the system is the basic layout of an entry, and this is best illustrated by using a simple subject statement:

planets - orbits - calculation

The layout consists of two areas: the heading (which is divided into the lead and the qualifier), and the display, in the format:

Lead    Qualifier

Display

Strings similar to the ones above are "shunted" around this format to ensure that there will be an entry filed under each candidate search term and that at the same time the context (consisting of the remaining terms) retains the same set of one-to-one relationships. For this example there would be three entries:

Planets
Orbits. Calculation
Orbits. Planets
Calculation
Calculation. Orbits. Planets

The process of shunting is not as arbitrary as it may seem from this example. For the sake of clarity, the role operators were omitted, and the "context-dependent" relationship between the terms relied on to suspend questioning. There are a number of operators and these will be introduced in batches. Those referred to are summarized in table 4.1 at the end of this Chapter. The most frequently used are "1" and "2". "2" is placed before an action, while "1" indicates a "key system", which may be the object of a transitive action or the agent of an intransitive action. Those operators that are numerals are called the "main line operators", and may have other operators interposed between them. Of these the most frequently used is "p" which represents a part or a property. These three operators would be those applied
to the example given above:

1) planets
   p) orbits
2) calculation

These operators are part of the rules that ensure consistency in the writing of strings and control to some extent the manipulation of the string to form entries.

There are several other operators which are not so frequently used. From the main line group, "0" indicates a location. Some of the interposed operators are: the co-ordinate concept "g", and "q" which indicates "membership of a quasi-generic group" (Austin, 1974, p423). This is best explained by usages, of which there are two. Firstly it is used to show that the relationship between two terms is a posteriori rather than a priori. In the following example, "space flight" is not inherently a subject taught in, and therefore a part of, schools:

1) secondary schools
   p) curriculum subjects
   q) space flight

This example shows quite well the facility that PRECIS has to introduce extra context-setting terms (in this case "curriculum subjects"). The other use of "q" is to introduce a "class-of-one", or unique concept. The following expansion of the original string will illustrate this and other operators:

0) East Sussex
   p) Herstmonceaux
1) planets
   q) Venus
   g) Mercury
   p) orbits
2) calculation
The shunting would produce a set of entries that would be essentially the same as those above.

There are three other main line operators, each of which introduces some form of "pragmatic" (ie non-subject) information about the document. "6" is the most common, for it represents the format of the item such as "teaching kit" or "bibliographies", as well as the audience the item is directed toward (eg "for children"; "for land surveying"). "5" introduces either the examples used for a study:

5) study examples
q) Crab Nebula

or the region or particular regions chosen:

5) study regions
q) universe visible from the southern hemisphere

This operator is used when a particular instance is used to illustrate a more general account. The last of the group is "4" which indicates the author's viewpoint, should it represent a distinctive school of thought. These three operators introduce a distinctive format: the inverted format. This is best illustrated by an example:

1) sun
2) eclipses
3) anthroposophical viewpoints

The three entries would be:

Sun
   Eclipses -- Anthroposophical viewpoints
Eclipses. Sun
There is a third format which is the most elegant of the three, but before an explication, the last of the main line operators and some more interposed operators must be introduced. The operator "3" introduces either the agent of a transitive action:

1) telescopes
2) construction
3) Galileo

or an aspect, factor, or instrument:

1) universe
3) mathematical models

The interposed operators are "$v$" and "$w$". The dollar sign is used in the MARC format to indicate a "tag code" that denotes the beginning of a field or sub-field in a record. They both are placed after a term to show that it should be connected to another term. "$v$" is used to link downwards (the point of the letter is downwards), while "$w$" is for the upward link (again there is mnemonic in the shape of the letter). It is usual, although not obligatory, for these to introduce some text, usually prepositions. To illustrate further, connectives may be added to one of the strings given immediately above:

1) telescopes
2) construction $v$ of $w$ by
3) Galileo

When "telescopes" is in the lead, the rest of the entry will be: "Construction by Galileo". When "Galileo" is in the lead, the
rest of the entry will be: "Construction of telescopes". These operators are devices for introducing natural language features for the purpose of making the message explicit. In fact, this string would be unambiguous because telescopes can't construct Galileo in any obvious sense. If the core of the subject was represented by the terms: "research students - attitudes - examiners", it would be ambiguous, although one or other party may instinctively prefer one interpretation.

To illustrate the "predicate transformation" the following string will be used:

1) solar system
2) planets
3) close-range gravitational interactions $v$ with $w$ with
1) astronomical bodies

If the standard format was used, the entries would be:

**Solar system**

   Planets. Close-range gravitational interactions with astronomical bodies

**Planets. Solar system**

   Close-range gravitational interactions with astronomical bodies

**Close-range gravitational interactions. Planets. Solar system**

**Astronomical bodies. Close-range gravitational interactions with planets. Solar system**

Suppose that the same printed index had a related string:

1) solar system
p) astronomical bodies
which would give the entries;

**Solar system**

Astronomical bodies

**Astronomical bodies. Solar system**

Compare the entries created from both strings with the led-term "astronomical bodies". Both are about astronomical bodies in the solar system. There is a context-dependent relationship between both terms. Yet in the first string, the terms are separated by the phrase: "Close-range gravitational interactions with planets". If the index has several entries under "astronomical bodies", then these two related entries will be physically separated on the printed page. The predicate transformation is a device to achieve a measure of collocation and surmount the problem outlined. In a string where there is either an operator "3" immediately preceeded by an action, or the sequence "1-2-1", then when the term introduced by the operator immediately after the action is in the lead, then the agent (and any other terms it is appropriately connected to) appear in the display area. To illustrate; the predicate transformation would have given this entry from the first of the two strings above:

**Astronomical bodies. Solar system**

Close-range gravitational interactions with planets

If this compared with the entries from the second of the strings, it will be seen that collocation on the printed page will be achieved.

So far the manipulation has been treated as if all terms except connectives were led, and the operators alone controlled the manipulation. As an interlude before introducing the last set of operators, the full manipulation codes will be briefly introduced. As this code is of fixed length, each of its characters will be introduced by a number:
1) Always a dollar sign, which (as with the connectives) signals the start of a new sub-field

2) This is usually "z". Not all documents are about a single subject, and to avoid having to write a separate string for each theme, the indexer can mark parts of the string as belonging to one theme or another by using "x" as the first item, and "y" for the following parts of the theme. Anything marked with "z" is interpreted as being common to all themes. Again an example will clarify:

```
$z1 man
$y2 sleeping
$y2 eating $v &
$yg drinking
```

This would produce the following set of entries:

```
Man
Sleeping
```

and

```
Man
Eating and drinking
Eating. Man
Drinking. Man
```

3) This position holds the role operator.

4) In the examples given up to now, all terms have been shunted into the lead position. The indexer actually has the choice, and all leads are coded "I" and non-leads "0" at this point.

5) The "substitute number" will be explained later.

6) This allows the indexer to chose where a term will or will not be printed as the string is manipulated. Usually terms are coded "3" which means they are printed wherever possible. For a term to appear only in the lead, the code is "0"; "1" is used to suppress
a term when a following one is in the lead; and "2" is used to suppress a term when an earlier one is in the lead.
7) This character has not yet been assigned a use.
8) This is another sub-field marker.
9) PRECIS was designed to operate in conjunction with a descriptive cataloguing standard, and therefore uses the same typographic conventions. This character contains a code to control the typesetting.

These codes have been presented because they are used in the first of the programs of the system presented below to aid the preparation of the strings for analysis. The particular system that was available to this project did not have characters seven to nine of the coding, although they have since become available in a more advanced version of the system. Therefore all strings presented in this thesis will be assigned the six figure manipulation coding.

In the account of the predicate transformation it was stated that a "3" had to be preceded by an action rather than an operator "2". This was because there are two other types of action in PRECIS. The more common is represented by "s" and indicates that a term is acting as a "role definer". Those with a grounding in case grammar would probably better understand this as introducing the action of either an instrumental phrase:

\$z2103 \text{calculations}
\$zs003 \text{use}\ \$v\ \text{of}
\$z3103 \text{computers}

or the action and agentive of a beneficiary:

\$z2103 \text{astronomy}
\$zs003 \text{influence}\ \$v\ \text{of}
\$z3103 \text{astrology}
The second of these actions is the operator "t" which is used to indicate a relationship between two concepts that the author has imposed, rather than is naturally existing:

$z1103 \text{Mars}$
$zp103 \text{astronomical data}$
$zt003 \text{related to}$
$z3003 \text{dimensions of Great Pyramid}$

One entry from this string will serve to illustrate the format:

Mars
Astronomical data related to dimensions of Great Pyramid

This operator is rarely used, as is "r" which represents an aggregate. Its use in the British Library's files seems to be for the representation of two or more concepts which each deserve individual leads:

$z1103 \text{Mars}$
$zp103 \text{surface features}$
$z3100 \text{photographs taken from Mariner 6}$
$zg100 \text{photographs taken from Mariner 7}$
$zr100 \text{photographs taken from Mariners 6 & 7}$

This string would allow entries to be made for the individual Mariner missions which would therefore file together with other entries describing just single missions. The operator is therefore being used as a device to obtain a good printed index.

This has exhausted the list of operators that take the full manipulation coding. There are some other operators, the "differencing operators", each of which takes the same form as the connectives (ie a dollar sign followed by one character). "$d$" allows the indexer to assign a date to a string:
Space flight to 1969

This is probably the easiest of the operators to apply, and fairly frequently used, even in the science sample chosen for this research. The parenthetical difference operators "$o$" and "$n$" are extremely rare. They both have the same purpose which is to indicate a particular method used where it is a significant factor in reaching the author's results or conclusions. As an example, there are several methods of calculating planetary orbits. A document that is about the use of a particular method in itself could be coded with the operators "$s$" and "$3$":

$zs003$ use $v$ of $z3103$ Kepler's laws

To write a string in this way for an item that used a method only as an illustration for something else would be misleading, and here "$o$" or "$n$" could be used:

$zs003$ use $v$ of $z3103$ Kepler's laws
$z1103$ planets $zp103$ orbits $zs2103$ calculation $so$ Kepler's laws $zs003$ applications $v$ of $z3103$ computer systems

As an illustration of the entry produced, when the term "orbits" is in the lead position, the whole would be:

Orbits. Planets Calculation (Kepler's laws). Applications of computer systems
The difference between "$n$" and "So" is that the latter will generate a lead for its text whereas the former will not.

This exhausts the repertoire of operators that indicate the essential logic of the system. So far nothing has been said about the "parts" as opposed to the "nuts and bolts" of the system. Two words have been used without explanation, neither of which has one generally accepted meaning in information science. For present purposes the definitions given by Austin (1974, p7-10) will be followed.

A concept is "a unit of thought which, being expressed in words selected from natural language, can logically be matched by one or more role operators in PRECIS". A term is "the verbal representation of a concept and may consist of one or more words". The following are terms drawn from the British Library's files:

- moon
- space flight
- instantaneous interstellar space flight
- interstellar space
- stars in region of galactic equator at longitude 140

We have seen above that the indexer has the choice of allowing or disallowing terms to be used as leads, and that several features of PRECIS are designed to achieve collocation in a printed index. If PRECIS consisted just of the operators described so far, then there would be no collocation between "space flight", "manned space flight" and "instantaneous interstellar space flight". Equally, it would only be an accident of language that this last would file near "interstellar space". This lack of collocation would be a problem for anyone who wanted to retrieve all documents on, for instance, "space flight". PRECIS offers the facility to split its terms into parts
consisting of a main part (the "focus") and subsidiaries (the "differences"). The account presented here is limited in two respects. Firstly it does not explain when a differencing operator should be used, rather than an interposed operator. This is because the research presented below is concerned with translating the strings as they stand, rather than prescribing limits to how they should be written. Secondly, this is one of the areas where PRECIS has developed, and the operators described in the Manual have been superseded by a more powerful version.

The new differencing codes are prefixed by a dollar sign, and have two further characters. The first controls whether or not the text following the code is to appear in the lead, and if it should have a space before it when placed before another word. It is summarized in this table (Austin, 1982, p216):

<table>
<thead>
<tr>
<th></th>
<th>Space generating</th>
<th>Close-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Non-lead</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

So from the examples given above, "manned space flight" would be written as "space flight $2$- manned". If the term was "windmills" and an entry were required under "mills" as well as the whole, it would be written as "mills $3$- wind".

In these examples a hyphen has been used in the second position. This may be a number in the range one to nine, and indicates how "far" this difference should be from the focus when it is in the lead. So this term taken from Austin (1982, p217):

\[\$zl103 \text{ panels} \quad \$21 \text{ reinforced} \quad \$22 \text{ steel} \quad \$21 \text{ coated} \quad \$22 \text{ plastics}\]

would produce the following entries:

Panels
This exhausts the repertoire of role operators, and there remains one more significant construction in the language. Two features of the system have been illustrated, being the ability to create natural language phrases to aid user's comprehension; and the writing of strings that will provide good entries and collocation. Once the terms have performed this second function, it is sometimes desirable to re-express some of the terms in a more natural form. The process is in effect the substitution of some terms by another. The string used to illustrate the parenthetical difference, "$0\), was a re-writing of a British Library string which in full is:

$z1103$ planets
$zp103$ orbits
$z2103$ calculation $0$ Kepler's laws
$z2032$ calculation of orbits of planets by Kepler's laws
$zs003$ applications $v$ of $w$ in
$z3103$ digital computer systems

When the last term comes into the lead, the entry would be:

**Digital computer systems**

Applications in calculation of orbits of planets by Kepler's laws

It is possible to substitute terms by nothing; that is to say a
substitute can be used to delete a part of the string which may be judged redundant. In this example, it could be judged that the terms before "Stonehenge" are unnecessary once it has arrived in the lead. The fifth character of the manipulation code indicates how many terms are to be deleted by the substitute:

$z1103 Wiltshire
$zp103 Amesbury
$z1003 monuments $2l henge $2l megalithic
$z1032
$zql03 Stonehenge

This constitutes the range of the syntagmatic relationships of PRECIS. The operators can be seen as a "syntactic" system, and some writers have in the past chosen to contrast this with a "semantic" system, the thesaurus. The thesaurus is an integral part of PRECIS, and holds the paradigmatic relations between terms. Terms are held in records in the file that are allocated a running number called the "Reference Indicator Number" (RIN). The RIN file packets are connected in several ways, but no detail will be given here because it does not immediately affect the translation of PRECIS. The mechanism provides the see and see also references in the printed index. In British Library PRECIS, the complete strings are included in records (which also contain other subject information, such as classification numbers and subject headings) which are allocated a running number, which give these records the name "Subject Information Number" (SIN). The version of PRECIS used in this research did not at first have a thesaurus facility, which will probably explain in part some of the decisions taken in the work.

The fifth objective of the original research that produced PRECIS was to represent concepts by natural language terms, although it was noted that there was an element of control imposed on the vocabulary. Apart from the choice between synonymous terms to represent particular concepts, and the
hierarchical links in the thesaurus, the form of the words used is also controlled. The preference of certain forms over others has been a matter of evolution and PRECIS has largely taken over existing practice. Almost all terms included so far have included nouns and adjectives. When there is a choice, then a noun form should be chosen, so "teach" is represented by the gerund, "teaching"; and "calculate" by "calculation". The product of this action is represented by "calculations", clearly showing the preference for plural forms when "things" (as opposed to actions) are indexed (Hutchins, 1975, p20-22). As regards parts of speech, apart from nouns and adjectives, there are only prepositions, some determiners, a couple of conjunctions, and very occasionally a relative pronoun.

This account has been written from the point of view of translating PRECIS. It is not the individual words or terms that give difficulty, but when they are manipulated into the display or the qualifier to produce phrases which may contain more than one term linked by prepositions. For this reason, attention has been paid to the results of the manipulation, rather than the meaning and application of the operators alone.

PRECIS was designed for a printed bibliography, and Figure 4.1 gives a typical page from the index to BNB. The British Library publishes a number of other indexes in which it is used, and its Bibliographic Services Division has the task of writing strings for the catalogue of the British Library (Reference Division), which was formerly the British Museum Library. All this data is included in the British Library's on-line retrieval service, BLAISE. Other libraries may arrange with the British Library to have their catalogues produced from British Library data, and some such as East Sussex County Library, chose to have a PRECIS index. By virtue of the data being held on a computer, it is possible for the British Library to produce various listings of their data, of which the most important to the work being presented here have been the microfiche of strings arranged
Figure 4.1

Extract from subject index to annual volume of BNB (1980)
alphabetically by terms, and of strings listed by the Dewey Decimal Classification numbers assigned to individual SINS. There have been a number of pilot projects to create PRECIS indexes, and those in languages other than English are reviewed in the next Chapter, together with an attempt to translate strings automatically between English, French and German.
Main line operators

0 Location
1 Key system (Object of transitive action; agent of intransitive action)
2 Action/effect
3 Agent of transitive action; aspects; factors; instrument

4 Viewpoint-as-form
5 Sample population; study region
6 Target; physical form

Interposed operators

p Part; property
q Member of quasi-generic group
r Aggregate
s Role definer
t Author-attributed association
g Co-ordinate concept

Differencing operators

$n Non-lead parenthetical difference
$o Lead parenthetical difference
$d Date as a difference
$01-$39 Compound term differences

Connectives

$v Downward reading connective
$w Upward reading connective

Theme interlinks

x First theme element in a co-ordinate theme
y Subsequent element in a co-ordinate theme
z Element of common theme

Table 4.1

Schema of PRECIS role operators
References


CHAPTER V
APPLICATIONS OF PRECIS IN LANGUAGES OTHER THAN ENGLISH
AND THE PRECIS TRANSLINGUAL PROJECT

Natural curiosity soon led those responsible for the implementation of PRECIS within the British National Bibliography to attempt its application to languages other than English. Staff whose native tongues were other than English translated strings into their language, and these were evaluated by (preferably other) native speakers of the languages under consideration. This process reached its summation with the publication in the Manual of an exemplary string in ten languages (Austin, 1974, p503-509).

This exercise demonstrated that the order of terms need not change between languages; indicating underlying inter-term relationships that appear to be common to all languages. On the surface level, the forms of terms are likely to vary as the string is manipulated. This is not a constant between languages, of course, for English has virtually no inflection, whereas German has some and Polish rather a lot. Apart from these two fairly fundamental conclusions, there was a third lesson to be learnt: that it is possible to divide the different language versions of PRECIS into two classes upon a simple test. Where there is in a string a second action with its agent, the first type of language (for instance English, French and German) employ an upward reading substitute with two connectives, and bring the predicate transformation into play, as in this English example:

String
$z1103 urban regions
$z2103 regional planning
$z2022 urban planning
$zs003 role $v of $w in
The second group (which includes Chinese, Finnish, Swedish and Norwegian) do not resort to prepositions to indicate relationships, and thus the subject is expressed as two themes as in this Finnish example:

String
\$x1103\ kaupunkilaissuunitelma
\$y2103\ seutukuntasuunitelma
\$y3003\ sosiologeina rooli
\$x1103\ sosiologeja
\$yp003\ rooli kaupunkilaissuunitelmassa

Entries
Kaupunkilaissuunitelma
Seutukuntasuunitelma. Sosiologeina rooli

Seutukuntasuunitelma. Kaupunkilaissuunitelma
Sosiologeina rooli

Sosiologeja
Rooli kaupunkilaissuunitelmassa

When PRECIS has been applied to languages other than English,
they have almost without exception been of the first group, which may have less to do with difficulties of language structure and more to do with the relative development of the information field in the associated countries.

Following on the publication of the Manual in 1973, several people have studied the application of PRECIS in French. Guy Dionne (1975) published a description of PRECIS I (based on a course taught in Montreal), which is now only of historical interest. An annex included a catalogue of ten items in classified order, together with an index and a list of the strings assigned.

At about the same time, Françoise Lamy-Rousseau (1974) was presenting proposals for the organization of a Canadian audiovisual materials index. For the purposes of illustration, a sample index of some four hundred and thirty-seven items was included, together with an analysis of the operators used in the strings. By way of comment on the possibility of the automatic translation of strings, she included five English strings, which were translated word-by-word into French ("traduction littérale par l'ordinateur") and then presented in an edited version ("traduction améliorée") (Figure 5.1). These examples presumably were intended to show the difficulties of machine translation, but serve rather to show the author had a less than complete grasp of the potential of the machine.

At the Université de Rouen, Germaine Lambert produced an experimental index of some one hundred and thirty theses in the life sciences, divided into a listing by university, and the subject index itself (Figure 5.2) (Université de Rouen, 1976). The form of the documents is immaterial, for it did nothing to lessen the difficulties of indexing, although the scientific vocabulary would have eased some thesaurus problems. The experience gained in this pilot project was used in a fairly detailed article describing the application of PRECIS in the
1. CHAINES ANGLAISES

Documents
Subject indexing. Applications of computer systems

Indexing. Documents
Subject indexing. Applications of computer systems

Subject indexing. Documents
Applications of computer systems

Computer systems
Applications in subject indexing of documents

2. TRADUCTION LITTERALE PAR L'ORDINATEUR

Documents
Sujets indexation. Applications de systèmes informatiques

Indexation. Documents
Sujets indexation. Applications de systèmes informatiques

[Sujets indexation. Documents
Applications de systèmes informatiques]*

Systèmes informatiques
Applications à l'indexation des documents

3. TRADUCTION AMÉLIORÉE

Documents
Indexation par sujets. Applications de l'informatique

Indexation par sujets. Documents
Applications de l'informatique

Informatique
Applications à l'indexation par sujets. Documents

From: Lamy-Rousseau, 1974, p34. (* indicates the addition of an entry omitted from original).

Figure 5.1
Manual translations by Lamy Rousseau
Figure 5.2
Extract from the Rouen index
Romance languages, including the problems of the creation of lead terms in adjectival constructions (Lambert, 1976). The main part of the project, to produce a full index to seven and a half thousand items, seems not to have come to fruition.

At about the same time, the Département des Arts du spectacle of the Bibliothèque Nationale produced an experimental index to one hundred items on the performing arts. Again the experience gained was used for a paper, this time giving a very brief outline of the system (Ferrier, 1978).

All the work so far reviewed has produced an index (albeit sometimes very small). A different approach was taken by Madeleine Laliberté who, in 1977, completed a Ph.D at Case Western Reserve University, entitled Selected grammatical and syntactic problems in applying PRECIS to the French language (Laliberté, 1977b). This was followed by an article summarizing her conclusions (Laliberté, 1977a) and by some corrective comment (Verdier, 1978a). Her starting point was that Lamy-Rousseau's experiment had not overcome all the basic "syntactic" problems (Laliberté, 1977b, p34). She therefore offered a theoretical rather than a practical study of some difficulties that seemed likely to arise. The concern of the work seems to have been for the monolingual use of PRECIS in French, and for the generation of French PRECIS as part of a machine translation system. This work was undertaken at a time when the work of Lambert and Ferrier had not been published, and thus Laliberté's prime inspiration was the work of Lamy-Rousseau. Unfortunately, she seems to have not investigated the then current state of machine translation, seemingly accepting Lamy-Rousseau's "traduction littérale par l'ordinateur" as being the results to be expected from any system. The two works on machine translation cited in her thesis are the collection of essays edited by Booth and Locke and published in 1955, and the ALPAC Report. It is easy to show that the word-for-word translation of PRECIS from English to French (preserving the original manipulation coding) will produce
unacceptable results, with articles omitted and adjectives appearing in the wrong place. It is naïve to believe that this is how a reasonable machine translation system would operate.

To cope with the bad syntax of the word-for-word translations of English terms such as:

$z1103$ bridges $s21$ suspension

Laliberté suggested two new operators ("$e$" and "$f$") to cause the lead instruction to be altered to form idiomatic French:

$z1103$ $f$ ponts $s21$ suspendus

This was designed to produce the entries:

Ponts
Ponts suspendus

Ponts suspendus

Even if it is accepted that these two entries are of use (why have the first entry if it is going to appear on the printed page adjacent to the second?), then why is the differencing operator "$s21$" kept? The addition of these two new operators is tantamount to suggesting that French strings should be indexed as if in English, and then adapted to the requirements of French. Space has been devoted to the explication of this unfortunate misunderstanding, because once this is recognized, then the wheat may be separated from the chaff in this thesis. The work of value in this thesis lies in the study of the implications for the construction of the thesaurus of the parenthetical difference, and the account of the use of the article with prepositions and with geographical and political names.

The principle difficulty that has preoccupied those who would wish to use PRECIS in a Romance language is the difficulty of providing a high number of entries where the adjective follows the noun, and cannot really stand alone. This has been treated in a British Library draft specification (Austin, 1979).
There has been some experimental work in languages other than English and French, most notably in a demonstration index for the Danish national bibliography, and one for the "energy" collection held in the library of Chalmers Technical University, at Göteborg. Reinhard Supper (1975) published a description in German, and Jutta Sørensen and Austin (1976) a more generalized study of its application to the Germanic languages. Austin (1982) has submitted a thesis to the University of Sheffield which describes the development of PRECIS as a multilingual system, and therefore considers many of the problems specific to particular languages.

Recently, the National Film Board of Canada has begun to issue bilingual versions of its catalogue, with PRECIS indexes separately created in English and French. This is the only multilanguage index to be produced non-experimentally. The distinction of producing a trilingual index belongs to the research team of the PRECIS Translingual Project. This was again an experimental index, to some two hundred and forty-nine items included in an issue of the EUDISED (European Documentation and Information System for Education) index. This was manually translated and undertaken as a pilot study for the major part of the Project (Figure 5.3).

This project was the natural progeny of several trends in information retrieval and international co-operation in the early and mid nineteen-seventies. The exchange of data between national bibliographic centres had been made possible and encouraged by the development of the MARC format. On-line information retrieval systems, searched from remote sites via telecommunications, had come into being, not least BLAISE, the British Library's own system. International exchange seemed only more likely to increase with the proposal for Euronet, a European computer data network. BLAISE was seen as a suitable host for this network, making available the UK MARC records. The subject information
English index

Laboratory techniques, Blood transfusion, Manchester 615.30
Labour market, Industrialised countries 529-792
- 1997-2002 — Foreman 331
Lake ecosystems 573-558
Laserechirurgie, England
Hospitals for mentally handicapped persons. Group relationships 301.16
Laminaria See Amphipod
Land forms See Landforms
Land, France
Geographical features 914.4
Landes de Garonne, Landes (Dép. des) France Geographical features 914.4
Landes (Dép. des) France
Landes Geographical features 914.4
Landes de Gascogne Geographical features 914.4
Landforms
See also
- Coast
- Mountains
Language
See also
- Speech
Language, Deaf persons 801
Sign language
Language skills
- See also
- Speech skills
Landscape
- For primary school 599.355
Laps, Northern Sweden 944
Social life — For primary school 944
Largely key children 362.7
Late teens
Use in removal of — take
Lausanne 621.94
Landes 621.94

French index

La Masselle (Dép. des) France
Val de l'Orne, Union agricole de Romorantin. Ameublement — Voir aubacures 669
Laboratoire
Voir aussi
- Techniques de laboratoire
Lacs, Milieux écologiques 573-578
Lanes, Grande-Bretagne
Maisons pour handicapés mentaux. Relations de groupe 301.16
Landes, France
Caractéristiques géographiques 914.4
Landes de Gascogne, Landes (Dép. des) France Caractéristiques géographiques 914.4
Landes (Dép. des) France
Landes Caractéristiques géographiques 914.4
Landes de Gascogne, Caractéristiques géographiques 914.4
Language
Voir aussi
- Littérature
Pardo
Langage par signes, Sourd 801
Lapin
Renon — Pour l'enseignement primaire 599.735
Lapins, Nord de la Société
Vie quotidienne — Pour l'enseignement primaire 948
Los groupes d'âges, Fête
Fête de Muise, Tours d'écureuil. Découverte par Olimpia, Venise 391
Le Corbusier, Architecte 72.036
Legislation, Grande-Bretagne 34

German index

Laborzeichen, Bluterkrankheiten, Mann 615.38
Lancashire, England
Hospitals for psychologically handicapped persons. Group relationships 301.16
Landes, Frankreich
Geographische Züge 914.4
Landes de Gascogne, Landes (Dép. des) Frankreich Geographische Züge 914.4
Landes (Dép. des) Frankreich
Landes, Geographische Züge 914.4
Landes de Gascogne, Geographische Züge 914.4
Landformen
Sehe auch
- Coasts
- Mountains
Language
See also
- Speech
Language, Deaf persons 801
Sign language
Language skills
- See also
- Speech skills
Landschaft
- For primary school 599.355
Lappen, Norra Sverige 944
Social life — For primary school 944
Largely key children 362.7
LauteTeen
Use in removal of — take
Lausanne 621.94
Landes 621.94

Figure 5.3
Extracts from the EUDISED index
would consist either of classification numbers or indexing terms derived from English. It seemed not only useful, but necessary to translate this language dependent data, so as to make the records as widely accessible as possible.

This then was the environment of the Project. Its orientation owes much to the ideas expressed in the Manual (Austin, 1974, p418-422). The Reference Indicator Numbers suggested a number language or interlingua and this, coupled with the techniques for organizing multilingual thesauri led the proposers of the Project to concentrate on an interlingual approach.

This Project was granted sixty-three thousand pounds by the British Library Research and Development Department to run for a period of two and a half years from the beginning of 1976. In the public announcement of the funding, it was stated the the general aims were to be:

"to create a set of routines and computer programs which will add a translingual component to the PRECIS system. This will enable the computer to convert the input string into a series of language-independent codes and translate these later into appropriate terms in a target language. These terms will then be manipulated into index entries in the target language without further intervention by the indexer" (Development of PRECIS..., 1976).

The pilot stage of this project confined itself to translating a small subject database, which turned out to be the EUDISED index referred to above. The second stage was intended to involve the extension of this experience gained to the translation of larger databases, and perhaps building a new multilingual thesaurus. At all stages it was intended that experiments should be limited to English, French and German.

The research team appointed were chosen for their language abilities rather than for knowledge or experience of using PRECIS
or of machine translation and computation. This might have been less of a severe handicap had not the British Library Bibliographic Services Division, in a spirit of economy, withdrawn the promised data processing support. The team were allowed only to approach the systems analysts for advice about the feasibility of their ideas.

The difficulties that this embargo would cause were immediately recognized, and the detailed objectives of the Project changed accordingly. It was no longer possible to study the cost and management aspects of an operational system, and the two remaining objectives were recast as:

- the design of detailed specification for all translingual procedures, set down in the form of flowcharts;
- the manual testing of these algorithms in a manner that would simulate (as far as is reasonable) the decision making procedures of a machine system (Verdier, 1980, p12).

It is not a fundamental objection that the proposed system was not programmed: but there must remain an awareness that detail would have been changed if the team had had their system implemented.

As has been shown above, the fundamental approach to the problem (translation via an interlingua) had already been decided on. The methodology of the Project was completely empirical, in that the EUDISED database was taken as a starting point, and routines were devised to deal with the problems encountered. New data was then devised, being designed to test and add to the existing procedures. The methods employed owed much to the experience of multilingual thesauri, and where this proved insufficient, then ad hoc methods were used to overcome problems encountered. Techniques drawn from computational linguistics that could have provided elegant and extensible solution to some of the problems were rejected as "too complex for use with an indexing system" (Verdier, 1980, p148).
The system was centred around three lexicons and a "pivot file", which held the numerical interlingua together with addresses for the three language lexicons. The lexical record (Figure 5.4) was closely allied to the PRECIS thesaurus in that it held information about a term rather than a syntactic unit. Some of this information related to the character of the term (for example, the grammatical number, syntactic structure, and its ad hoc characteristics), while the rest of the record held data more associated with the thesauri, such as the Reference Indicator Number, manipulation coding data, and notes.

The similarity of the lexicons to thesauri was influenced not only by the PRECIS thesaurus itself, but also by a draft international standard on the creation of multilingual thesauri. This deals with the definition of degrees of equivalence between languages, and recommends strategies for dealing with each case. Of the five categories, two seem to have caused the research team most problems. Non-equivalence between two languages will always be encountered, and here the recommendation was to construct semi-artificial equivalents in the lacking language. So the English term "open plan education" was assigned the French equivalent: "enseignement (p) nouvelles methodes (q) 'open plan education'". The other difficult category was where a single term in a single language mapped onto an equivalence constructed of more than one term. The example that illustrates this was the German "Schneeken" which was rendered in English as "slugs $v \& (g) snails".

These lexical problems have to be faced in any mechanized translation system. The corollary is that a method has to be devised to ensure that when it is necessary to retrieve a lexical unit which consists of more than one term, then the appropriate unit is retrieved, and not (in the cases of both the examples given above) three individual units. This is also a problem that has had to be faced by designers of any computer system that
<table>
<thead>
<tr>
<th>LANGUAGE</th>
<th>FRENCH</th>
<th>LEXICON NO.</th>
<th>0024</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TERM</th>
<th>003</th>
</tr>
</thead>
</table>

**Single Term**

- 003/1

**Head of Block**

- 003/2

**Lexical Block**

- 003/3

**Begin with a Noun**

**ECUREUVILS**

<table>
<thead>
<tr>
<th>TERM TYPE</th>
<th>004</th>
</tr>
</thead>
</table>

**Preposition**

- 004/1

**Proper Name**

<table>
<thead>
<tr>
<th>Person</th>
<th>004/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genus</td>
<td>004/3</td>
</tr>
<tr>
<td>Corp.</td>
<td>004/4</td>
</tr>
<tr>
<td>Opus</td>
<td>004/5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Taxon</th>
<th>004/6</th>
</tr>
</thead>
</table>

**Compound**

<table>
<thead>
<tr>
<th>Structure</th>
<th>004/7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ</td>
<td>ADJ</td>
</tr>
</tbody>
</table>

**Rules for Compounding Noun with Noun**

- Insert 's' 004/8
- Insert 'n' 004/9
- Type (a) or (b) 004/10

**Action**

- 004/11

**Trans.**

- 004/12

**Intrans.**

- 004/13

**Neither**

- 004/14

**Number**

- 005

**Singular**

- 005/1

**Plural**

- 005/2

**Neither**

- 005/3

**Other Form**

- 005/4 ECUREUVILS

**Gender**

- Male 006/1

**Masculine**

- 006/5

**Noun**

- 006/7

**ADJECTIVAL FORM**

- 007

**4th Digit**

- 008/1

**5th Digit**

- 008/2

**6th Digit**

- 008/3

**Manip. Coding**

- 009

**Reference**

- 010

**Translation**

- 010/1

**Preposition**

- 010/2

**Refinement**

- 010/3

**RIN**

- 011

**0001823**

**PIVOT FILE**

- 012

**0031**

**NOTES**

- 013

**REFERENCES**

- 014

**DATE**

- 015
deals with text, for it is the problem of ensuring that the longest possible match is retrieved. To term it a "problem" is a misnomer, for it has been solved, probably many times over, and has become nothing else but a fairly complicated procedure. It is unfortunate that the research team saw it as a problem (perhaps not realizing that it was a difficulty common to all machine translation projects). Their solution consisted of labelling all entries in the lexicon as either nothing; the head of a lexical block (that is, the term was used as the first term in a lexical block); or a lexical block (a unit made up of more than one term). On finding that the term being searched for was the head of a lexical block, the intention was to perform a sequential search of the file from that point to find the longest equivalent. The explanation of the procedure given by the research team was not detailed enough to serve as instructions for a programmer, but surrounded with enough unnecessary complications to give their proposals a mystique which was unnecessary and misleading.

The proposal to use more than one term in a lexical unit had inherently a power to distinguish between homonyms. It has already been shown that context terms can be inserted into a string to indicate an intended meaning where otherwise ambiguity may be present. Thus "orbits" would be disambiguated by being used in either the form:

$z1103$ planets $zp103$ orbits

or

$z1103$ eyes $zp103$ orbits

This device, together with the control of word forms in PRECIS provided an invaluable aid to analysis. Its disadvantage in application was of course that each term admitted would need a separate lexical record, even if it was composed of syntactic units used in other terms. Yet the lack of a file of single syntactic units makes the translation of substitute phrases very difficult indeed.
The switching procedure as described above can be diagrammed as at Figure 5.5. All the while that translation is restricted to terms drawn from the thesaurus the task is going to be (in comparison to natural language translation) very easy. Some strings contain prepositions used as connectives, and these are notorious for their difficulty in translation and the research team were aware of this. They first conducted a survey of one thousand English strings drawn from the UK/MARC file held on the BLAISE retrieval system. This showed that fewer than ten per cent of the strings contained a preposition. Most of these strings fell into one of two standard patterns, being either:

$z2103$ action
$zs003$ role definer $v$ of $w$ in
$z3103$ agent

or

$z1103$ key system
$z2103$ action $v$ by $w$ of
$z3103$ agent

As for the particular prepositions used, these numbered about thirty (including locutions), some of which were very rarely used. It was found that this also held for French and German. Of these prepositions, only six had single equivalents in all three languages, and could therefore be added to the lexicons in the same way as were terms.

To deal with the choice of the correct translations of prepositions, it was decided that the computer should be presented with an algorithm which would use information about the terms in the target language string. A system of ad hoc categories were introduced to resolve the choice, because a method of processing a series of production-type rules related to terms in the strings would be too complicated.

Category numbers were assigned to the exceptional
The switching procedure

1. Terms from the source language string are matched one by one against the source lexicon, to find the address of these concepts in the PIVOT FILE.

2. The Pivot file is a language-independent numerical file, where the addresses of all equivalents for a given concept are held.

3. By means of this address, the equivalent in the chosen target language is retrieved from the target lexicon.

Figure 5.5
Switching procedure of PRECIS Translingual Project
translations of prepositions. "De" would normally be translated by "of", but exceptionally by "in", "for", "by", "from", "to" or "as to". At analysis, the lexical record for "de" would (via the pivot file) direct the processing to an equivalent. As this was set down by the research team in the form of an algorithm, it would probably have been held as a program, rather than as a series of rules in a data record. Such an algorithm would be as at Figure 5.6. Interestingly, the resolution of the target language preposition was not carried out with reference to the ad hoc category alone, but included some information about the string itself. This latter information included the category number assigned to the previous and succeeding terms; and their respective role operators (Verdier, 1979; 1980, p61-76).

It was recognized that some processing time could be saved if some terms and associated connectives were treated as single lexical units. Thus "influence $v$ of $w$ on" could be mapped onto its usual equivalent, and exceptions suitably allowed for. The advantages of this would be several. It would eliminate some processing because, first the number of possible translations would be reduced because of the inclusion of an extra term (here "influence"); and second because the preposition algorithms would not have to be negotiated so often. It would also utilize the existing procedures for retrieving and translating the longest possible match.

The procedures as devised were not based on any linguistic theory, or indeed on any sort of theory. They were fairly complicated and all the less explicable for not being based on any "human" method of processing.

In English PRECIS it is almost always acceptable to omit the article after a preposition, as in "effects on social change". In French and German the opposite is true, as in "effets sur le changement social". Strings written by a French indexer would have articles where necessary, and for a translation into English
All paths in this algorithm terminate with the direction.

Figure 5.6
PRECIS Translingual Project algorithm for "of" and alternative translations
these would have to be removed to produce idiomatic PRECIS. At the analysis stage, the research team reduced the preposition plus article construction to a representation of a single preposition in the pivot file. So to analyse "pour les", the system would have attempted to retrieve a lexical record with the key "pour 1". Its equivalent in the English lexicon was "for" and its associated exceptions.

When French or German was the target language, the pivot file was designed to point to the preposition record only. So the translation into French of "for" would be "pour" and its exceptions, not "pour" plus the article. At the end of the algorithm for deciding between "pour" and its equivalents, the processing was directed to a general article insertion routine (Figure 5.7). Apart from the articles "à" and "de" (which had their own algorithms), and "en" which the Project recognized as needing no article, all French prepositions were be treated by this algorithm. The information needed in the processing was the gender and number of the succeeding noun phrase, and if the next word had a "vowel start". Certain cases were marked as exceptions to the general rules on insertion, such as proper names.

This routine was designed to add articles wherever possible, rather than wherever necessary. For instance, the EUDISED French index includes a number of strings with the prepositional phrase "pour enfants". This algorithm would have produced the phrases "pour les enfants". This is clearly because the data on the behaviour of the article with individual prepositions was collected in a single algorithm, rather than spread around the system, perhaps in the lexical records for individual prepositions.

In German and in Polish (which the Project temporarily adopted) there is an element of inflection. For PRECIS, this only occurs when there is a prepositional phrase, and is allowed for only by some extra codes to indicate which term is to be selected
Figure 5.7
PRECIS Translingual Project article
insertion algorithm
for linking with a connective, but which does not indicate the surface case which is present. In this German example, the accusative form of "Mensch" is present:

**String**

$z1103$ Mensch $t2$ Menschen
$zs003$ Einfluss $v$ der $w$ auf den
$z3103$ Religion

**Entries**

**Mensch**

Einfluss der Religion

**Religion**

Einfluss auf den Menschen

For this Project, the analysis routines were designed to ignore the inflected forms, translating just the nominative forms into the pivot file representation, and from there into English and French. This was because if the latter two languages were the target languages, there would be no point in analysing the inflected forms as well. For translation into either Polish or German, inflected forms obviously had to be added where appropriate. Procedures were designed to determine which surface case should be present, and to add the suitable form. For German, these algorithms were centred largely around the preceding preposition. In passing it should be noted that the Project designed routines to either extract a given case form from the lexical record, or to construct it from the nominative form. Again, this was duplicated work.

The final part of the project dealt with the translation of substitute phrases. The survey of the strings retrieved from the BLAISE file showed that about five per cent contained a substitute. The first approach to their translation was to treat each as a single lexical unit, thus using the same procedures as
for translating the terms, and avoiding the problems that arise when the substitute is not just a rewriting of the terms elsewhere in the string. The disadvantages were that the size of the lexicons would be increased, although it is quite likely that a given substitute would only be used once in the entire database. The corollaries of this method are that the speed of matching in the source language lexicon would be slower (given the searching algorithm proposed); and the amount of work in creating the lexicons would be increased. From an indexing point of view, because there would be no thesaurus control, there would be a possibility of inconsistencies of phrasing of the substitutes (for instance: "educational policies" and "policies on education"). Verdier states that for these reasons, it was decided not to translate the substitutes, but to create them from the terms in the target language strings (1980, p147-48).

It should be noted that their system design rendered them incapable of translating substitutes. The algorithms designed for processing prepositions introduced by connectives relied on the role operators and ad hoc categories on either side of the datum. With a substitute, the former information isn't present, so their method would have collapsed if a preposition for which there were multiple equivalents was present. Substitutes that coincidentally occurred as terms in their own right could have been translated of course. Because there was no dictionary of single syntactic units and parsing methods had been rejected, the Project could only fall back on the automatic construction of substitutes.

The Project devised two methods which differed from in each other in that one took the substitute coding already provided in the source language string, whereas the other did not. The first method was founded on the assumptions that the substitute in the source language strings would meet exactly the requirements of the target languages; and that a target language string would not need a substitute if the source language string did not have one.
The survey of substitutes in the strings retrieved from the BLAISE file were split into three groups by the Project. The "standard substitutes" were composed of terms and prepositions already included in the string. The "simple substitutes" had the terms but not all the necessary prepositions in the string. The final group, the "complex substitutes" were divided into three sub-groups. The first were those that eliminated some context-providing terms for a given noun:

**String**  
$z_{1103}$ schools  
$z_{p003}$ curriculum subjects  
$z_{q103}$ science  
$z_{2103}$ teaching  
$z_{2032}$ teaching of science  
$z_{3003}$ use $v$ of $w$ in  
$z_{3103}$ audiovisual aids

When "audiovisual aids" is in the lead the substitute phrase cuts out the terms, "curriculum subjects", "science" and "teaching":

**Audiovisual aids.** Schools  
Use in teaching of science

The second sub-group consisted of strings which had some noun-form terms substituted by an adjectival form:

**String**  
$z_{2103}$ medicine  
$z_{p003}$ research  
$z_{2022}$ medical research  
$z_{2102}$ historiography

The third sub-group was characterized by the compression of two separate terms into a compound phrase:
The aim of the Project's first method was to cast all substitutes in the form of a noun phrase. The manipulation coding was retained, on the assumption that it was common to all the three main languages of the Project, but the text was discarded. The extent of the regime of the substitute was determined by the coding, and the terms and (where they were present) the prepositions from this part of the string were set down in a defined order. This procedure was adequate for the standard substitutes, but further algorithms were needed for the other two types. A series of "refinement algorithms" were designed to add prepositions and create adjectival forms where necessary. The information that these routines used was mainly the operators of the terms on either side of the preposition and more ad hoc information, called "refinement categories". These routines were also capable of providing an adjectival form, for instance "human" instead of "man". This last routine shows up the real weakness of the PRECIS Translingual Project, for the routines resort to traditional grammatical information, such as part of speech, gender and number, even though the automatic creation of substitutes was an attempt to render such processing unnecessary. In fact the flowcharts aren't so sufficiently detailed as to show how an adjectival form is to be transposed to the correct position before or after a noun, or indeed how it would be decided with which noun the adjective has to agree. This is less important than that the Project had eventually to move away from a purely data processing to a more computational linguistic approach, although the type of linguistic processing proposed reminds one more of the Georgetown system than for instance, of
TAUM-Meteo.

The first method of providing substitutes was judged by the research team to be deficient in three respects. First, the lead terms in the target string could be different in number and position from the source language string. If that meant that a substitute was included which was not used, this was just a trivial waste of time, but if the target string needed a substitute which was not in the source because of a non-led term, then there would be a loss of information. Second, the number of terms in the source language string could be different from the number in the target string, because of the inclusion of a lexical block. The calculation of the regime of the substitute from the retained coding would (as the system was envisaged) give an incorrect result. Finally, it was always possible that the indexer has incorrectly decided to include or omit a substitute phrase.

All of these problems could be overcome if the substitute codes could be provided automatically, and indeed the substitutes would not even have to be provided by the indexers who wrote the source language strings. This process would need to make three decisions. The first would be whether or not this particular string needed a substitute. If so, then the second would be where it should be placed, and how many terms it should replace. The team designed procedures to carry out these tasks. The regime of the substitute was denoted by adding a new piece of information to the string, the "level code". This was produced by calculating how many terms would appear in the heading when the term after the central term (that is the term that would be at the heart of the newly constructed substitute) came into the lead. With this information, the algorithms for providing substitutes according to the Project's first method could be used. The level code could have been added automatically, but the Project investigated both this and manual provision and chose the latter, perhaps because time ran out.
The Project recommended the programming of their system as expressed in their flowcharts. It was suggested that three areas would profit from further research. It should be stressed that these recommendations were made after the system presented in the body of this thesis had been generally designed. First, it was suggested that the lexicons should include not terms, but smaller syntactic units. The rationale was that the lexicons would take less storage space because of the repetition of words in several terms would be avoided. The Project seem to have overlooked that they, in effect, designed their lexicons to be PRECIS thesauri, with some extra information added. If the lexicons were revised, then there would be even more wastage of space because the terms would be stored in the thesauri, while individual words would be stored in the lexicons. The justification for separate dictionaries of single syntactic units must be that translation would be significantly improved over the use of just extended multilingual thesauri. Yet the Project seems not to have parted from the aim of creating target language substitutes rather than translating those in the source language. The rest of the recommendations suggest in effect that the system should be given some justification by appeal to linguistics and to machine translation research. The procedures recommended for further examination were the provision of articles, and of inflections; and the construction of substitutes. Finally it was suggested that the “contents of the categories should be investigated on linguistic grounds, since an analysis of the type of term of which each is composed may prove to be of value to the field of automatic translation generally” (Verdier, 1980, p278).

To evaluate the Project: the specification of program routines for all the processes meant that the modification of the performance of the system would be difficult, as it was for the Georgetown system. As for the adequacy of the routines as they stand, it is difficult to judge whether or not they would produce adequate results for a random sample of the BLAISE file. The
strings used to test the algorithms have not been published, and there is no account of how the sample was chosen. It is probable that after collecting a basic set of strings from the EUDISED indexes, new examples that would be in some way difficult to translate were constructed. The system design makes the introduction of new languages inherently difficult, because for each new language there would have to be a thorough analysis of any features that could not be translated via the lexicons, with a new set of ad hoc categories created and new programs to be written. For these reasons, one cannot agree with Hutchins's comment that "the methods ... are admitted to be ad hoc, but defensible on practical grounds of computability and the restricted context of translation in a highly formalized indexing language" (1980, p89).

The final criticism is related to the economies of the system. There is nowhere in the final report a judgement on the economic feasibility of the proposed system. Given that the Project was situated in the very institution for which the system was intended, this seems an unfortunate omission. Some parts of the design are cost-effective, given certain assumptions about the proposed system. For instance, if it is assumed that there is to be a thesaurus for each language, then the use of these as lexicons is good sense. If the thesauri are separate from the lexicons, then it should still be possible to create both together for only some extra cost. The majority of work went into the creation of the algorithms for processing prepositions and substitutes. No comment is made on whether it is economically worthwhile to program these routines, rather than allowing for post-editing of the minority of strings that would have needed processing over and above just lexical transfer.

One's personal feeling is that it would not be economically worthwhile, and these criticisms have been made with the experience of implementing a transfer system, which will be the preoccupation of the remainder of this thesis.
References


Laliberté, M. (1977a) Quelques problèmes rencontrés dans


In

The thesis of the work presented here is that it is possible to implement on a minicomputer, a translation system for PRECIS which is readily extendible in two ways. Firstly, it should be easy to add more European languages that use the Roman alphabet; and secondly, it should be adaptable to other string indexing systems.

It is probably as well to set down some of the circumstances and attitudes held at the start of this research. The facilities available in the Department of Library and Information Studies (DLIS) of Loughborough University of Technology seemed ideally suited to the proposed work. It had its own Digico MTS 16 minicomputer (which has since been upgraded to a Digico M28 min-mainframe), with adequate backing storage. A suite of programs for the manipulation of PRECIS replete with some of the newer features which allow for multilingual processing had been written by Frederick Smith, and these are at least the equal of the suite currently available to the British Library. The programming language available was Digico BASIC which is superior to many versions of BASIC by virtue of its powerful string-handling features, which in turn makes it a good tool for data processing. The DLIS has been amongst the foremost UK library schools in the teaching of computer applications in library and information processes, taking the view that information workers would get the systems that they deserve, rather than the ones they needed, unless they made sure that they were involved in the design of their computerized systems. The author had just completed his first degree in the DLIS and adhered (and still adheres) to that view. The difference between the attitudes at the beginning and
end of this research is simply put: at first it seemed to be just a data processing problem, whereas now it seems mainly a computational linguistics problem.

As originally conceived, this research was to start with the programming of the PRECIS Translingual Project's system, and then go on to test its adequacy, to improve it, and to attempt to explain its more empirical features in theoretical terms. A series of informal meetings and communications had taken place between the Project's research team and some members of the DLIS. The fundamentals of the switching procedure had been described, together with some of the more complex processes proposed. The programming of such parts of the system as for which descriptions were available was carried out, together with the entry of a subset of their lexicons, kindly released by the British Library (Research and Development Department). This same body also made available a pre-publication copy of the final report and algorithms, and it was then that the dependence of the system on ad hoc routines became apparent. As has been shown in the previous Chapter, the proposed methods were judged not to be suitable for the achievement of the objective of producing a system readily extendible to other languages. On this evaluation, it was decided to abandon the PRECIS Translingual Project's approach and attempt the design of a more elegant solution.

The orientation of this research has never been theoretical in the sense that it has never been in itself an attempt to explain the structure of indexing languages. That is not to say that it has rejected theory, but those chosen have been used only on account of their ability to solve problems such as the translation of prepositional phrases. If the opposite of theoretical research is practical research, then this is the latter. But it was never practical in the sense that it stood or fell on whether an institution adopted the completed system. If the aim had been to produce such a (saleable) system, then the system presented would have been altered early on to make it more
commercially attractive.

It was useful during the design and implementation to keep a specific institution in mind, and the obvious candidate was the British Library (Bibliographic Services Division). The PRECIS suite they used at the beginning of this research was a batch system, with the indexers writing their strings on input forms. These were then keyboarded, processed, and the results returned. It was decided that the translation of strings should best be carried out after the keyboarding of strings, and the system designed as a module of this batch system.

Following on from this, several other features suggested themselves. A feasible system should be able to achieve good results while being inexpensive to maintain and improve. The cost of building the initial dictionaries and adding such new entries as the policy of an open-ended vocabulary would entail must not involve the user in an unjustifiably high cost, although it must be recognized that there will be some significant cost involved if a complicated process such as translation is to be undertaken. Should another language be added, then it should equally be inexpensive to incorporate. (This in particular was where the PRECIS Translingual Project would have performed badly.)

The other two objectives were always going to be difficult to attain. Firstly, the output from the system should be as good as the input; and secondly, the system should be a tool that would verify the source language strings in some respects. The PRECIS system already has some validation routines that check the characters in the manipulation codes, and to a limited extent, the sequence of some operators (Austin, 1974, p483–85). Given the kinds of processing proposed below, then it is possible to check on the syntactic ambiguity of phrases that appear in the manipulated string, and (by using semantic information) the sequence of role operators.
To summarize the objectives:
1) The system should be implemented on a minicomputer.
2) It should be readily extendible to other languages.
3) It should be extendible to string indexing systems other than PRECIS.
4) The problem is practical in that it needs solution, not explanation.
5) The system should be a batch one, suitable for insertion into a PRECIS suite.
6) The building and maintenance of the dictionaries should be inexpensive.
7) The addition of new languages should be inexpensive.
8) The target language strings should be as good as the source language strings.
9) Some verification of the indexers' strings should be included.

Two factors in particular influenced the design of this system. As noted in Chapter IV, the DLIS implementation of PRECIS current at the time the system was designed did not have the thesaurus facility, although this has since been added. There is a school of thought in information retrieval that holds the thesaurus an unnecessary, and even unhelpful, feature of retrieval systems, and this view was held at the beginning of this work. It was easy then to abandon the lexicons consisting of terms that the PRECIS Translingual Project used. Those with more conventional views of information retrieval will be pleased to know that opinion has changed during the course of the research, and has come to rest on a conviction that a thesaurus is necessary for a printed index. It should be noted, that this does not imply an acceptance of the thesaurus for all retrieval systems, and especially not for on-line systems; and also that the PRECIS thesaurus is particularly noteworthy for being catholic in its acceptance of new vocabulary, rather than being just a relatively inflexible list of subject headings.

The second factor was a view that the fundamental weakness
of the Translingual Project was that it could translate terms such as would appear in the lead position, but that it never really addressed itself to the translation of the blocks of text that would appear in either the qualifier or the display. The use of connectives to link two or more terms in these areas produced phrases similar to some of substitute phrases that caused the Project so much difficulty. This work recognizes the essential similarity of these two types of phrase, and treats them in the same way. To make a generalizing summary of the difference between this work and the Translingual Project: the latter translated terms in the thesaurus, whereas the former translates the text as it appears on the printed page after manipulation.

Before passing on to an account of the general system design, there is one matter that should be disposed of. This thesis presents a translation system. As such, there have to be some dictionaries, but the contents of these are of secondary importance to the mechanisms devised for dealing with the problems of translation. In other words, the equivalents presented here are not definitive, and it is recognized that a good English/French terminologist would have, in part, chosen others, and made a better and more consistent job of creating the dictionaries.

**General system design**

The PRECIS Translingual Project had adopted an interlingua of sorts. This was rejected in favour of a transfer strategy because notice was taken of the difficulties experienced by those who had worked with interlingual natural language MT systems. Even the TITUS system has not achieved the results desired here for the translation of PRECIS. The less stretching nature of the transfer strategy seemed to be a more promising candidate. In retrospect, it must be said that the gulf between the translation of natural languages and indexing languages wasn't completely grasped at the design stage.
In keeping with good computational linguistics practice, the complete separation of linguistic data and the programs that operate on that data was aimed for. This is good practice in that it makes changes to the performance of the system a matter of changing the grammar or dictionaries, rather than the programs, unless of course a particular process has been overlooked altogether. It is good sense if the system is to be extendible to other languages and indexing systems, because their incorporation would again hopefully be a matter of writing some more rules, rather than new programs. There are places in the system where this separation has not been ruthlessly applied, and where this is so, it will be noted in the ensuing descriptions.

As was shown in Chapter 11, the transfer strategy has three distinct parts: analysis, transfer, and generation. The first should take account of no other than the source language, and the last no other than the target language. If maximum economy is to be achieved when translating from and into several languages, then the majority of processing has to be done by these routines and as little as possible by the transfer module.

The schema in Figure 6.1 shows the flow of translation. Each process will be described in detail in the following Chapters, and here only an overview will be given. As has probably become clear in the preceding pages, the smallest unit in the dictionaries was the single syntactic unit; rather than the term as the PRECIS Translingual Project had used. The easiest method of ensuring that correct equivalents would be chosen at transfer was to carry out some limited syntactic analysis. The view was taken that it needed nothing more than the identification of parts of speech and phrase groups. At no stage was there an attempt to identify surface syntactic structures such as object and subject. The identification of parts of speech was used because it seemed that in a practical environment it would be easy for a number of terminologists over a period of time to
Figure 6.1

System flowchart of the transfer translation system
build up the dictionaries consistently. The application of syntactic category labels was idiosyncratic to a degree (as will be shown in Chapter VII), and this was because the view was taken that syntax here should do no more than show up the underlying patterns of the syntagmatic arrangement of words.

It was found that syntactic analysis would solve almost all problems of choosing the correct equivalence. It was insufficient for obtaining the correct translations of prepositions. Further, "deeper" analysis is needed to uncover the semantic structure of the text. This project used an analyser based on case grammar to produce a semantic structure for those PRECIS strings that included one or more prepositions.

After the analysis had been performed, the strings could then be passed to the transfer routine for the selection of equivalents in the target language. It is here that the advantages of using a limited language are really obvious. In essence, the process is nothing much very more than lexical transfer. There is certainly nothing akin to the transfer of a source language surface structure to a different target language one.

Generation in this system is concerned with two general functions. It has some of the functions usually associated with transfer, in that it has the task of choosing the appropriate preposition, given the semantic information obtained at analysis. Where this information isn't present, information from the bilingual dictionaries has to be used as a "safety net". The second function is the re-ordering of syntactic units, the insertion and deletion of articles, and morphological generation. In other words, the classic functions of generation.

Methodology

The first stage in the project as re-defined was to analyse and learn from the strengths and weaknesses of the PRECIS
Translingual Project. After deciding on those areas which needed different methods, the literature (and thereby the experience) of computational and theoretical linguistics was used to find alternatives. In particular the monograph by Hutchins (1975) and conversations with Derek Austin who was preparing Chapter VIII of his thesis entitled Language-independent features of PRECIS (1982, p126-213) directed attention toward case grammar.

At this stage, the design of the project existed in outline, in about as much detail (but without the same degree of clarity and confidence) as the account given above. It was at this point that a sample of the British Library's PRECIS strings drawn from BLAISE was adopted. Those chosen were taken from the file of strings arranged in order of the Dewey Decimal Classification numbers assigned to the SIN records. These were drawn from the "PRECIS DC fiche" for May 1981 (British Library, 1981b). The range of classification numbers chosen were 520 to 525.35024553 and 629.4 to 629.8. The sample therefore constitutes strings on the subjects of astronomy and astronautics. Many, if not all, visitors who have discussed the project have enquired why this particular subject area was chosen. There were two reasons. The first was that a science or technology field was thought to be the best to start with. The second reason is more eccentric, and explains the choice of these particular subjects. At the time the sample was chosen, Voyager 2 was approaching Saturn, and the author shares a mild, dilettantish interest in astronomy and astronautics with two cousins, and the choice of these subject areas was a tribute to their interest in the work, as much as anything else.

The intention was to use this sample to develop the system, which it was supposed would take about a year. The strings that the British Library had added to BLAISE in that time could then have been taken and used to test whether or not it had achieved a measure of generality, or whether solutions had been tailored to the problems encountered in the original sample. Also, statistics
could be collected on the amount of lexical updating that would be needed over a year, from which some economic judgements could be made. The creation of the original system took far longer than originally envisaged, and so this second stage was never undertaken. There was to be a third stage of testing, which was to adopt another subject field, more associated with the social sciences (education was the favoured candidate). The purpose of this sample was to find out how many changes (mainly to the dictionaries) would be needed to accommodate different subject areas, and whether the performance of the system on the science and technology sample would degenerate as a result of any changes made.

This use of only one sample is acknowledged to be a serious drawback to the credibility of the system, and it is intended to rectify the situation as soon as possible.

The sample chosen was entered as a single file of PRECIS strings alone, rather than with the complete information from the SIN record. Each string was allocated a running number. They were transcribed exactly as in the file, except that the character set and methods for representing diacritic marks developed for this project was used (Appendix A); and the manipulation coding was reduced from nine to six characters. Even obvious errors were copied. Shortly after this sample was transcribed, the British Library published a new authority list for some terms used to describe formats and introduced by the operator "6" (British Library, 1981a). The strings in the sample were changed under the direction of Derek Austin, to be in accord with the new standard. The total number of strings in the sample ended up as four hundred and twenty-three, and are presented in full in Appendix B.

The following Chapters will present the modules of the system one-by-one, giving details of dictionaries and grammars used as and when necessary. There are several programs that have
the sole function of changing the format of the string into another format so that succeeding linguistic modules can work efficiently. These text preparation programs are interspersed amongst the linguistic modules, and will be described as and when necessary.
References


British Library (Bibliographic Services Division). (1981a) PRECIS categories of forms. 3rd ed. British Library (Bibliographic Services Division).

British Library (Bibliographic Services Division). (1981b) PRECIS DC fiche. British Library (Bibliographic Services Division). [Negative microfiche].

The reason for using syntactic analysis has been set down in Chapter VI. Essentially, it had the objective of assigning the correct parts of speech to words in the source language, so as to ensure that the correct equivalent was chosen at transfer. Thus at the simplest level, "outer space" (being an adjective and a noun) was translated as "espace intersidéral" (a noun and adjective), rather than "spatial intersidéral" (two adjectives).

The analysis of the source language text for the purposes of transfer was the most important objective of this module. The second was to act as a preparation module for the semantic analyser. As will be shown below, the semantic analyser operated on the nouns and verbs uncovered by syntactic analysis, and it was by virtue of this that this module can be said to be a preparation, as well as an analysis, module.

The third objective was the discovery and rejection of ambiguous strings. Syntactic analysis is a fairly blunt tool, and the grammars used for this project were particularly un honed. Two types of syntactic ambiguity came to light during the work; neither of which are new to linguists. The first is typically caused by the ability of English to qualify a noun by another in the same way as an adjective does. There were no examples in the sample drawn from BLAISE, but the following string (taken from the PRECIS Translingual Project's index for the issue of EUDISED referred to in Chapter V), is an especially good example:

\$z1103 \text{ metals \$21 turning}

Does this refer to the "turning of metals" or "metals for turning"?
The second type of ambiguity is most readily explained by reference to an example such as "The man saw the girl with the binoculars". The ambiguity lies in the difficulty in deciding whether the man was using the binoculars, or whether the girl possessed them. Nothing of comparable clarity was found in the sample used here, the best example being "paintings of manned space flight by Smith, Ralph A compared with Apollo Project" (sample string 395). Although the ambiguity is unconsciously resolved by the reader, to a syntactic analyser using a suitable grammar, "Smith, Ralph A" could have been the agent of either the paintings or the manned space flight.

The view was taken that as PRECIS is an artificial, and therefore limited, language and because the economic feasibility of the system would rely on the least amount of information needing to be recorded in the dictionaries, it would be reasonable to, in some ways, circumscribe the strings accepted by the analyser. As these restrictions were governed not by the programs, but by the grammars, those imposed in this project will be noted when the grammars are discussed.

**Text preparation (Program MTl1C1)**

As the system was designed to operate in the batch mode, strings had to be read one-by-one from a file, rather than entered on-line. These were held in the form shown in Appendix B. This program converted these strings into a format that the syntactic analyser could readily handle. The documentation and program listing is given in Appendix G.

The point should be made that this program was specific to the PRECIS indexing system. It would be capable of processing any PRECIS string that does not include the extra codes introduced to cope with inflected forms. That is to say, it could process strings in English and French, but not German.
The first part of this program (lines 1 to 804) split the original string into the manipulation coding and the text. A brief routine (lines 820 to 850) labelled all themes in the following way. The elements common to all themes (those introduced by "$z"), were labelled "0", while the first theme was labelled "1", the second "2", and so on. This allowed themes to be processed individually.

The largest part of the program (lines 1200 to 1967) took each theme and created a downward and upward reading. So the following string (number 50):

$z1103 astronomy
$zs003 theories $v of $w of
$z3103 Copernicus, Nicolaus

had the following readings:

**Downward**

astronomy|theories of Copernicus|

**Upward**

Copernicus, Nicolaus|theories of astronomy|

Here the connectives have been used to join the terms in the same way as they would in the display or qualifier. Where there was a substitute (whether blank or not), the effect that this has on the manipulated entry was also simulated. The importance of the downward and upward readings must be emphasized. It is by this feature that all possible natural language-type phrases that will appear in the entry were simulated for the syntactic analyser.

Several items were omitted from the downward and upward readings, and listed separately. The date difference ("$d") was the most obvious candidate, because it neither fitted easily into the downward and upward readings, nor did it need much analysis because it is easily transferable into the target language. The
parenthetical differences ("$n" and "$o") were also given individual listings because their form (eg "calculation (Kepler's laws)") was not easily amenable to analysis. The third category was the terms that occur only in the lead position. As has been shown in Chapter IV, they are a device for obtaining a good, well-collocated, printed index. It was decided that as their function is, in essence, an extension of the thesaurus, and because they would appear in neither the qualifier nor the display, they would be listed separately. It was also considered that in the target language, such lead only terms as would be needed would be drawn from that language's dictionary or thesaurus (if present).

Once all the themes had been processed, the data was written to the first of the intermediate files, MT1XD1. Figure 7.1 gives the results of a manipulation of a string that contains both connective and a substitute phrase. After writing to file, the syntactic analyser proper was called.

**Transition network analysers**

The syntactic analysis program adhered almost completely to the accepted practice in computational linguistics of the separation of grammar, dictionary and program. The mechanism at the heart of this program and the semantic analyser was of the recursive transition network type. Before describing the use made of the dictionary and grammars, an explanation of the mechanism will be given.

At the simplest level there is the transition network analyser; the recursive transition network analyser (RTN) being an elaboration of this, and the augmented transition network analyser (ATN) being a still further elaboration. It has long been recognized that a "sentence" can be represented by a finite state transition network. This consists of start and end nodes, with other nodes representing intermediate states. The arcs
Venus origins theories of Velikovsky, Immanuel
criticism 1950-1977

Length of this section: 13
Number of dates: 1 1950-1977

Figure 7.1
Output from the text preparation module (MT11C1)
between the nodes can be labelled with conditions. To give a simple example, "John hit Bill" could be represented as:

```
Start  →  S1  →  S2  →  S3  →  End
  John   hit   Bill
```

Such a network would be of next to no use to anyone. It can be generalized by substituting syntactic categories for the words:

```
Start  →  S1  →  S2  →  S3  →  End
  Noun  Verb  Noun
```

Such a network would account for "John saw Bill", "Bill saw John", and indeed for "*Tree looked house", and many other varieties. ("*" indicates an unacceptable utterance.)

Even with allowance for repetition and alternative paths to states, the limitation of such finite state transition networks are obvious. Some have argued on theoretical grounds that it is unsatisfactory (Chomsky, 1957, p18-25). On practical grounds it was undesirable for this project because it would have needed many rules to account for all possible constructions found even in a limited language like PRECIS. Moreover it was uneconomic in that a sequence of categories found at one point in the network might be repeated several times elsewhere. Equally, a sequence found somewhere in the network might be needed at another point, but had been overlooked by the writer of the grammar.

So on grounds of size of the network, and of ease and economy in writing grammars, an alternative was needed. This was provided by the RTN analyser. Instead of arcs being labelled by just the names of terminal syntactic categories, some are labelled with the names of other networks. So in the case of the rather trivial network presented above, there could be two networks. The "sentence" network could have two calls on a "noun phrase" network:
The noun phrase network could be the following:

```
Start  S1  S2  S3  End
Noun   Verb   Noun
```

This could account for the sentences given above, as well as "John hit the ball", "John hit the big ball" and "The very big ball hit John". The use of recursive calls on sub-nets allows the analyser to function as a context-free parser.

The RTN analyser may be extended to give the ATN analyser by adding some extra storage registers that can hold information applicable to the whole sentence or text being analysed. Such registers can hold data about, for instance, the mood, subject and object of a sentence. The information stored therein can be exploited by the imposition of more tests on the arcs apart from, or instead of, the tests for syntactic categories already described. By virtue of its ability to use information collected elsewhere in the analysis, an ATN parser is able to implement a context-sensitive grammar. The judicious use of action commands to be executed on the successful traversal of an arc can allow a structure to be built that represents a transformation of the original sentence.
The popularity of the ATN analyser owes much to its use by William Woods in the LUNAR system for retrieving information about the geological samples brought back by the Apollo missions, and requested in natural language form, rather than stilted data base commands (Woods, 1970). It should not be thought that the ATN analyser can only be used for syntactic analysis. It is, in effect, a mechanism for searching and comparing a store of information (such as a grammar) against some data (such as a sentence). To give one example of an application to something other than syntactic analysis, R F Simmons used this type of analyser to implement a case grammar (1973).

An ATN analyser was judged more powerful than was needed at any point in this project, and nothing more powerful than an RTN analyser was used. Before describing the processes carried out by this program, the RTN analyser must be explained in more detail.

Analysers can be divided into two types; "breadth-first" and "depth-first". The difference between the two shows up where there are multiple interpretations of the data being processed. With the former approach, all alternative structures are recorded at a given time, and none given precedence. In "depth-first" parsing, the alternatives are processed sequentially (Wilks, 1976). RTNs are of the second kind, and therefore have the significant practical advantage of taking less working space in core storage because of only recording one alternative at a time.

Another way of categorizing this analysis is as "data-driven" and "hypothesis-driven" parsing. "Data-driven" is synonymous with "bottom-up", and refers to methods that start with the lowest possible structures (eg words) and attempt to build higher level structures over them. "Hypothesis-driven" is synonymous with "top-down", and refers to methods that "hypothesize several nested levels of structure before positing any constituents which can be checked against the input string itself" (Marcus, 1980, p15). RTN analysers are of this latter type.
The description so far has tacitly assumed that the routes taken through the networks will always be the correct ones. Parsers that do not allow an interpretation to be altered, once assigned, are called "deterministic parsers". This would be an acceptable mode of operation providing that there was never any doubt as to the choice to be made at a node. However this is frequently not the case, and the following example will illustrate a simple case of choice. Let the dictionary comprise two words with their syntactic categories attached:

- deep = noun, adjective
- pond = noun

Let the grammar be:

```
<table>
<thead>
<tr>
<th></th>
<th>Noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Adjective</td>
</tr>
</tbody>
</table>
```

Let the text to be parsed be "deep pond".

Note that the arcs of the initial state have been ordered. The parser would assign to "deep" the noun category (ie the sense of a "watery deep"), but be unable to account for "pond". This situation may be remedied by the inclusion of "backtracking".

There are at least two types of backtracking (Charniak, Riesbeck and McDermott, 1980, p258). The type used throughout this project was "chronological backtracking". With this type, a record is kept of each decision made, so that on failure the interpreter can return to the least decision point (ie the one
that is chronologically closest), and follow an alternative path. In the example given immediately above, on being unable to assign a category to "pond", the parser would undo its assignation of "nouns" to "deep", return to the previous state (here the initial state), and cast around for another arc off the node. It would take the arc labelled "adjective" and therefore be able to subsume both words. It follows that if a grammar were unable to account for the structure of a text, the parser would eventually backtrack to the initial state (assuming that it ever managed to get away from it). It also follows that by backtracking after the completion of a successful parse, any alternative structures will be discovered. Hence, "a synonymous phrase [for chronological backtracking] is depth first search. This name refers to the fact that a chronological backtracker can be thought of as exploring just one branch of a search tree at a time" (Charniak, Riesbeck and McDermott, 1980, p258).

As the analyses are produced sequentially, it makes practical sense to attempt to order the grammar in such a way as to have the most likely interpretation the first to be tested. This technique is known as "heuristic parsing".

ATNs and RTNs are open to criticisms on several counts, but there are two serious deficiencies. The first is that they can go a long way before discovering that a wrong decision has been made; this being because they are not exploring a single hypothesis, but many (ie one at each level) at any one time. Both Marcus (1980) and Milne (1980) have used methods of "looking ahead" to restrict the options tried by the analyser. The second deficiency is inherent in chronological backtracking. Because the latest decision is erased at each backtracking call, the parser can undo well-formed structures in order to get back to the point at which the error was made. It then proceeds to recreate the well-formed structure anew. Again, an improvement can be made; in this case the preservation of well-formed sub-strings when backtracking (Sheil, 1976).
Another objection can be made against techniques that use automatic backtracking (as opposed to backtracking as and when specified by the grammar writer). In a perfect situation there would be no need for backtracking, but in a less-than-perfect situation it is necessary to recognize where the error occurred and to undo it. Chronological backtracking doesn't recognize errors, but undoes previous decisions blindly, until it stumbles on the solution. Because all side effects (i.e., the incorrect structure) are erased, it is difficult to report where errors occurred, and thus to modify the grammar or programs. Automatic backtracking is a brute-force technique, because it is able to overcome many failures until it finds a solution. It thereby gives a sense of power, and can thereby lead to poor analysis of problems and poor design of programs and data (Sussman and McDermott, 1972).

There have been other objections made to purely hypothesis-driven parsing that stem from their seeming to be inappropriate as models of human language processing (although there has been at least one claim that ATNs can be used as such models (Kaplan, 1972)). Pulman describes a "realistic parser" as "one which works at two levels simultaneously: the lowest level mostly bottom-up, assembling phrase level constituents NP VP PP etc... The second, more 'top-down' routine would organize these phrases into functionally complete units - roughly, a verb with its obligatory arguments and any optional arguments or modifiers which might be present" (1980, p54). This is a proposal similar in intention, but not in means, to those of Marcus and Milne referred to above. From a practical point of view, bottom-up parsing can have some advantages. If information about some items is not found in the dictionary, then it is possible to analyse the remainder of the text partially, and even hypothesize the missing syntactic categories. In this way, the source language data could be carried over untranslated into the target language, particularly useful if the unretrieved unit or units are proper names.
These are the objections that have been made against RTNs. In spite of these, the RTN mechanism was used for this project for the following reasons. Core storage was at a premium and therefore a technique that developed one structure at a time was thought preferable. The design of an heuristically satisfying grammar could mean that the "best" analysis was the first developed. This was important in the syntax analyser because in the presence of multiple readings the first reading was accepted, but the ambiguity noted in the error file. It assumed more importance in semantic processing, for reasons that will be explained in the next Chapter. The lack of a dictionary entry (and thereby information about the syntactic categories of lexical units) was considered a sufficient reason for rejecting the whole string. Because strings are small units of text, and because rejection of one wouldn't affect the translation of the remainder, it was considered acceptable to reject strings at certain stages in the processing. Therefore the practical advantages of some bottom-up processing outlined above did not apply, even though the grammars used in this project did not include categories other than those Pulman suggested as appropriate for bottom-up processing.

Other advantages relate to the ease of constructing grammars. The pictorial form of networks used above are only indicative of what seems a very straightforward way of writing a grammar. Even if the grammar was poorly written, the desired solution would eventually be uncovered through the use of automatic backtracking. The apparent simplicity of the grammar formalism was on occasions found to be deceptive, for backtracking occasionally uncovered routes that were entirely unimagined at the time of writing.

There were two remaining advantages. The examples of lexical ambiguity were relatively rare, given the limited nature of the language, and the restricted range of syntactic categories employed. So while backtracking was always available (thus making
the mechanism non-deterministic), the parser in fact operated almost entirely as a deterministic machine which made its operation relatively quick. Finally, recursive functions proved easier to write in the version of BASIC available than iterative routines.

To re-state the previous pages succinctly: the syntactic analysis was done by a top-down (i.e. hypothesis-driven) parser which operated in a depth-first fashion and therefore developed parses sequentially. It resorted to automatic chronological backtracking on encountering failure, and was therefore non-deterministic.

The dictionary

The dictionaries used in this system were of two types. Those used for analysis and synthesis were monolingual, while those for transfer were bilingual. There is nothing special about this, as it is part of the classic model of a transfer system. It could be claimed that the system design was cumbersome in comparison with recent MT systems, and especially Eurotra, which apparently holds its lexical information as a single integrated data base (Knowles, 1982).

The monolingual dictionaries had three data areas. The first was the key; the second the analysis data, and the last the synthesis data. The analysis information was of two types: that relating to syntactic analysis, and that relating to semantic analysis. The latter will be considered in the next Chapter, while the former is of present concern.

The syntactic information consisted of a small repertoire of categories. Those used were applied in such a way as to indicate patterns in the syntagmatic structure of the phrases being analysed. No attention was paid to the niceties of linguistic theories, which explains the rather cavalier application of the
categories. The number of categories defined was kept to a minimum so as to make the creation of lexical records as easy and as economic as possible.

The system as finally conceived allowed for categories of up to four characters in length, although all were of two characters: a legacy from a previous system. The categories and their definitions were as follows:

- **no** ordinary noun: applied to units such as "man", "films" and "shuttles".
- **nn** name noun: applied to the names of people and of things, but not of places: eg "Newton, Sir Isaac", "General theory of relativity" and "Mariner 10".
- **np** place noun: applied to the names of places, such as "Cambridge", "America" and "Great Britain". It was distinguished from "nn" items because it could be prefaced by adjectives, whereas "name nouns" couldn't.
- **ad** adjective: applied to all words that could qualify a noun. So apart from conventional adjectives such as "outer", nouns that qualified other nouns (such as "space vehicles") were included, as well as possessives such as "Kepler's laws".
- **ve** verb. This was the most eccentric labelling of all. Certain past participles were used, and produced a distinctive syntactic pattern and, more importantly, a distinctive semantic pattern. In a spirit of pragmatism, units such as "visible", "compared" and "related" were labelled as "verbs".
- **pr** preposition: which included some prepositional locutions such as "as to".
- **de** determiner. Although broadly defined in theoretical linguistics, this class included only "the".
- **co** conjunction. This included only "and", "&" and a dummy symbol "*", used when processing lists.

These then were the categories. Any number could be applied to a lexical unit, although given the limited nature of the language and of the categories, a minority had more than one. Should the
need for more categories become apparent, there would be no problem in their addition.

The English dictionary allowed for no morphological decomposition. In particular, singular and plural forms of nouns were treated as separate entities. As was noted on page 68, the singular form in PRECIS in English is often used to denote an "action", while the plural form is used to denote the "thing" or product of the action. Thus the allocation of separate records to differing forms served primarily as a method for distinguishing differing meanings in semantic analysis.

Although not strictly relevant to the contents of the dictionary, this seems the best place to explain the method used to access it, and indeed the other dictionaries in the system. It was not feasible to hold the entire dictionary, or even just the keys in core storage, because of the limited size of core storage. On the other hand, directed searching of the dictionary on disc storage was too slow, even for a research rather than an operational system. The solution was to use a two-stage search, or in other words to use a compact index to the keys.

Each lexical unit in the dictionary was allocated four bytes of core storage, which contained two integer numbers, each compressed into two bytes using the ENC$ function of Digico BASIC; this being a function to produce integers stored in a format similar to packed decimal format. The second of the integers was the relative record number of the main (ie first) record of the lexical unit. The first of the numbers was an integer produced by a process of working through each character of the lexical unit, and multiplying the position of the letter in a string of valid characters by a prime number associated with the position of the letter in the lexical unit. The number obtained was added to a running number, and when the whole of the key had been so processed, the total was divided by a "golden number" (again a prime) to produce a near unique number. In the
three dictionaries in the system, there were less than ten clashes for about five hundred entries.

The index entries had to be computed before the MT system was run, and a failure to re-compute them after updating the dictionaries led to large scale errors. Obviously for each piece of text that was matched against the dictionaries, an index number had to be calculated. This was quicker than a binary search of backing storage, and search time was further improved by storing the length of the longest lexical unit, so that the number of fruitless searches could be cut down.

This method of accessing the dictionaries was a practical solution to a system difficulty. It is not suggested that it should be part of any implementation of the system.

The grammars

Two grammars were used in the English analysis module of this project. Well-designed grammars were the means by which strings could be recognized as syntactically acceptable, or unacceptable by virtue of being either ambiguous or ill-formed. It was found that terms introduced by the operator "6" and which indicated the target population could start with a prepositional phrase such as "for children", which was inadmissible elsewhere in the string. Two grammars were therefore used, one for most of the string, and the second for operator 6 terms.

There were two types of entry in the grammar. The first was the net name, which was always a set of up to four characters, which had to begin with an upper case letter. The second format was for individual states of the network. This had the basic format of:

state name - test - destination - action

If a state had more than one arc or test, the subsequent entry or entries were represented in the form:
test - destination - action

Two states from a hypothetical grammar could be as follows:

\[
\begin{align*}
\text{NP} & \\
S1 & \text{adj S1 WRITE} \\
& \text{noun S2 WRITE} \\
S2 & \text{*END LOAD}
\end{align*}
\]

Note that the end test needs no destination as it signals the end of a recursive call, and/or the end of the parse.

There remains two points that should be made in respect of the grammars. Firstly, the actions specified are not comparable to those specified in grammars for ATNs, in that they were never used to collect global information about texts. In fact they were used only in one of the semantic analysis grammars to control case frames, as will be shown in the next chapter. The second point that should be made is that, for ease of use the grammar was stored as a BASIC string matrix.

The two grammars used for syntactic analysis are shown in Figures 7.2 and 7.3. As will be seen, there was no attempt to build anything above phrase level constituents, because higher categories such as subject and object were judged to serve no function in the translation of PRECIS. Also no information as to number was included, because it was thought that this would not contribute to either determining syntactic structure or to the choice of the correct equivalent. The phrases in PRECIS could not be said to be sentences in the sense that that word is used of natural language. The name "FULL" was adopted instead because of its reference to the phrases being processed as being the fullest version of the text to be found in either the qualifier or the display of the entry.

There are several problems associated with context-free phrase structure grammars (Palmer, 1971, p124-134). The detection of ambiguity has already been discussed at the beginning of this chapter, with "paintings of manned space flight by Smith Ralph A"
Figure 7.2

Syntactic analysis grammar 1
Figure 7.3

Syntactic analysis grammar 2
being cited as an example of uncertainty as to where the second prepositional phrase should be attached. As will be seen from a study of the grammars presented, this ambiguity cannot be detected because all prepositional phrases were given equal "importance" in the "FULL" phrase; which may have been misguided, in as much as a more sophisticated grammar could have been a part of a better interface for the semantic analyser. Another problem that causes the natural language grammar writer some difficulty is discontinuous elements, such as "She rang John up". In English at least, it would seem that PRECIS has the advantage of not having this feature.

In this project, it was decided to reject strings that contained two adjacent noun constructions but which could possibly be open to some doubt as to its interpretation. The example of "turning metals" has already been given. This NP-NP structure was not allowed, and the indexer was forced to use a NP-PP structure. Unfortunately, such are context-free grammars, that (combined with a limited set of categories) the system rejected semantically well-formed phrases such as "forecasting eclipses". There is no doubt that this concerns the "forecasting of eclipses" and not "forecasting for eclipses".

This concludes the account of the grammars used. As with the syntactic categories, it would make no difference to the operation of the analyser if an alternative grammar was used, with the one condition that its initial network would have to be labelled "FULL".

Syntactic analysis (Program MT21Cl)

This program is listed and documented in Appendix H. The first process was to load the results of text preparation from the intermediate file, and to load the index to the analysis dictionary (lines 100-255). Each theme was processed in turn, and the results written to the intermediate file, a control record
written and one of two programs called. If a preposition had been encountered in any phrase that was not prefaced by the operator "6", the semantic analyser was called. Otherwise the transfer program was called. Prepositions in terms assigned operator "6" were excluded as a condition for calling the semantic analyser, because it was felt that these terms were likely to be drawn from a pre-defined list of descriptions, such as the British Library's PRECIS categories of forms referred to in Chapter VI. Given the use of such a restricted list, it would be reasonable to include a set of "official" equivalents in the transfer dictionary. In fact, the assumption that the British Library's terms introduced by operator "6" were capable of being definitively listed was wrong, because the open-ended vocabulary in PRECIS meant that new "targets" (i.e., phrases like "for children") could be introduced.

The processing of individual themes had an obligatory part and a conditional part. For all themes, both downward and upward readings had to be processed. Both readings had to be searched for the presence of operator "6", to enable the appropriate grammars to be loaded as and when necessary. As was noted earlier, the text preparation program listed "$d" differences, "$n" and "$o" differences, and lead only terms separately, if they were present. Date differences were not analysed, as their structure is sufficiently simple as to be easily translatable by the most rudimentary procedures. "$n" and "$o" and lead only terms were analysed in the same way as were downward and upward readings.

The processing of each portion of text was essentially the same. The entire portion was taken and converted into a search key by converting upper to lower case, and swapping spaces by "%". If the text was greater in length than the longest key in the dictionary, then the index was not searched. Otherwise an index number was computed and the index searched. If the number was found, the main record was retrieved from the dictionary. Overlength records had trailers, and if necessary, these were
retrieved. A failure to find the trailers caused an error (number 503) to be written to the error file, for reporting after the entire file had been processed. The error manual used for the project is included in Appendix Q. After all trailers had been concatenated, the portion of text being searched for was compared with the lexical record's key. If they were identical, the lexical unit and its category and categories were written to a storage area. The pointer to the text was then incremented, and any remaining text searched for.

If the index had not been searched because of overlength text, or if the appropriate lexical units had not been found, the program tested the last character of the text to see if it was a comma. This routine was written as part of the program, rather than part of the grammar. PRECIS forms entries that have more than two co-ordinate concepts in the format: "Mariner 6, Mariner 7 & Mariner 8". On finding a comma, this routine removed it and substituted "*" which was assigned the conjunction syntactic category in the dictionary. So the text above would become: "Mariner 6 * Mariner 7 & Mariner 8". This routine was simply a method of putting cojoined phrases into an easily manipulable format. If subsequent dictionary searches were successful, the data was added to the parse store, and the pointer advanced as above. Otherwise the text was retained in its previous form.

If the search for a comma at the end of the text was unsuccessful, then the program attempted to remove the last orthographic word from the text, and searched again. If there was only one word left, this could not, of course, be reduced and therefore a gap in the dictionary was reported, using error number 502.

In this program, dictionary searching was deterministic. This was feasible because of the limited nature of the texts, which meant that homonymous sequences of words were unlikely. There was nothing in the sample such as the (perhaps apocryphal)
SYSTRAN error of interpreting the words "Prime Minister Begin" not as a proper noun, but as a noun construction plus verb; "Prime Minister" and "begin". For the purposes of recognizing proper names in particular (which usually consist of more than one orthographic word), a routine for retrieving the longest match had to be included. There was no need for complicated marking of records as "lexical block" or "head of lexical blocks", as the PRECIS Translingual Project had done.

At the end of dictionary searching, there existed a store with each lexical unit and its associated categories. The next process was to decide which of the two grammars should be loaded, and if the correct one was not already in core, to load it. The RTN analyser was then called, and individual portions of text processed.

The RTN analyser has been described above. This particular implementation developed a tree structure as it developed the analysis. The design of the tree store was relatively naive, in that it not only held the links to fathers and sons, but also information such as syntactic label, rule invoked, and the extent of text subsumed, together with space for the role operator and the link to the semantic tree. Apart from the links to the tree itself, it is recognized that a better structure would have included the father and son links; possibly a link to the contiguous brothers; and a pointer to a store where any other necessary information could have been stored. This structure could make the system more flexible.

A failure to create a parse led to the reporting of error 501, and the rejection of the current string. Once the first parse had been created, backtracking was used to search for alternative parses, which, if found invoked the reporting of error 504 (ie an ambiguous phrase) and rejection of the string. A successful analysis is shown in Figure 7.4. All portions of the text were processed, until such times as the entire text had been
done. Each successful analysis was written to the intermediate file, and it was at this point that text introduced by operators other than "6" were checked for the presence of prepositional phrases.

After syntactic processing the majority of strings were passed to the transfer module. The remainder went to the semantic analyser, and it is this that is described in the next Chapter.
Figure 7.4
Syntactic analysis of string 69
Figure 9.1
Transfer grammar
References


As stated in Chapter VI, semantic analysis was used to uncover a further, "deeper" structure than syntactic analysis provided. It was on the results of this analysis that the correct choice of preposition could be made at transfer. It was also foreseen that semantic analysis could be useful in the choice of the correct lexical equivalent when transferring units other than prepositions, but for this sample (at least), this proved not to be necessary.

"Case grammar" is a distinctive area of linguistics attributable to one man, Charles Fillmore. In 1968 he wrote an article with the title The case for case, which presented the notion of "deep case", together with some proposals on how this idea could be incorporated into generative grammar. He reasoned that surface case endings show the semantic relationships between nouns or noun phrases, and verbs. Moreover, languages without case endings have to resort to other methods (such as the use of prepositions, word order or intonation) to achieve the same effect. He reasoned that underlying all languages must be the same "deeper" structures which are manifested in different ways in the surface structures. He therefore held that case categories must be assigned to the kernel (untransformed) sentences of generative grammar to account for the varying surface forms.

As originally set out, Fillmore envisaged simple sentences in their base forms to include two parts; the modality (which covered features such as tense, negation and mood), and a proposition. The proposition itself consisted of a verb and a set of participants. Each individual verb had associated with it a "case frame" which in effect was a list of the roles that participants were allowed to assume in relation with the verb.
Drawing on logic, the verbal element is usually called the "predicate", and the participants the "arguments". "There are only a small number of ways the arguments of a predicate are semantically related to the predicate itself. These ways are called cases". (Charniak, 1975, p1).

The verb "hit" may be part of several surface forms; for instance:

"The nail was hit by John"
"John hit the nail with the hammer"
"The hammer hit the nail"

Clearly the reader understands that John was always the agent; that the hammer was the instrument, and that the nail was the patient of the action. A case system is built in a similar way to this, in that a number of verbs are examined and their arguments classified. It is probably true to say that there are as many case systems as there are devisers, although it must be said that a degree of similarity between many systems is easily detectable.

Chafe (1970) introduced the classification of the predicate into those that refer to states, processes, actions, and action-processes. This classification can be further divided by particular cases that are to be expected with the predicate, such as "experiencer", "beneficiary" and "location" (Cook, 1978). Thus the process predicate "enjoy" is an experiential predicate, and therefore contains slots in its case frame for the patient (the thing being enjoyed) and the experiencer (the animate entity who enjoys).

"Selection restrictions" are often placed on the slots, to account for the acceptability of a sentence. "Eat" requires an animate agent, and therefore a sentence like "John ate the apple" is acceptable, whereas "*The post ate the apple" is not. Again there is no general agreement on the selection restrictions to be used, or even on their name. Chafe (1970, 1972) produced two
lists which exerted a significant influence on this project.

This isn't the place for a detailed consideration of the systems devised by theoretical linguistics, but should an interested reader wish to explore further, the main influences on this work (in order of importance) have been: Chafe (1970), Fillmore (1968, 1977), Cook (1978), and to a lesser extent Longacre (1976).

Case grammar has been criticized on several points. Firstly, it isn't a grammar as such, for it has no provision for features such as phonology. The term "case systems" is used here instead. Secondly, Fillmore originally envisaged case as being part of the structure from which surface forms were generated. It is not clear from those theoretical linguists who have sought to apply case systems to analysis just how the process should be carried out.

From the point of view of this practical project, the other objections are more interesting. Given a simple subject statement like "smoking as a cause of cancer in humans", it is possible to argue that smoking is not the mechanism that causes the growth of the tumour. This is illustrative of the difference Fillmore draws between "internal" and "external" semantics. "The notion of deep cases ... concerns, not the semantics of truth or entailment or illocutionary force, but rather the semantic nature of the inner structure of a clause." (1977, p60). For MT one is not interested in the truth or otherwise of an utterance, but in providing an analysis of its semantic syntagmatic structure.

The last objection has already been touched upon, and it is that there is no agreement on the number of cases. It is theoretically attractive to hope that a definitive list could be produced, if only because it would be a candidate for inclusion in an interlingua. For practical purposes such as in this project, it seems inappropriate to search for a definitive list,
in that the need is only for a set that will help to provide an adequate analysis.

A number of natural language processing systems have included a semantic component that may be described as a case system. Depending on how a case system is defined, more or fewer systems may be included in the class. There have been a number of review articles, of which that by Bruce (1975) covers the most. Papers by Samlowski (1976), Charniak (1975) and Wilks (1976) provide an interesting three-sided debate on the use of case in artificial intelligence systems.

Two systems in particular provided inspiration for the design of the system presented here. The first was Wilks's English to French translator (reviewed in Chapter II); the second was Harold Somers's system for "meaning analysis" and "dictionary-making", PTOSYS (Somers and Johnson, 1979). Wilks's system was an obvious influence, in that it was a working MT system. Although his case categories might have been adopted wholesale, they were not, mainly because it was felt that they included categories such as "containment" and "goal" which were not thought necessary for PRECIS. As Samlowski (1976) pointed out, some of Wilks's cases such as "possession" and "accompaniment" were not case relations as Fillmore would understand them. So apart from the inspirational effect, Wilks's system contributed the idea that the case system used did not have to adhere rigidly to the theories of the case grammarians, but could include other, more general, semantic categories.

The benefits derived from Somers's system were again twofold. Firstly, was the notion that one could use case without having to accept anything of transformational-generative grammar. Secondly, his system of cases was the starting point for the definition of the cases given below.
Unlike the previous Chapter, the dictionary will be described before the programs.

**The dictionary**

The syntactic information held in the English analysis/synthesis dictionary has been considered in Chapter VII. Categories except "no", "nn", "np" and "ve" were not assigned semantic information, and took no part in semantic analysis. The dictionary was in theory capable of holding up to sixty-three different semantic meanings for each syntactic category, although the actual maximum used in this project was three. There were two types of "semantic meaning" allowed: predicates and arguments.

Unlike natural language systems such as PTOSYS, that use some syntactic processing to uncover case items, it could not be assumed that only verbal elements would be predicates, even given the idiosyncratic definition of "verb" used. As has already been explained, "actions" in PRECIS are usually in noun form, and therefore nouns, as well as "verbs" could be assigned predicate records.

The predicate record consisted of a number of case slots. For each case frame there could only be one occurrence of each case slot. Also, each case frame applied to an individual predicate, rather than an entire string. To recast this is in natural language terms, the case frame applied to the regime of a verb, not to whole "sentences", unlike some case systems (Rosner and Somers, 1980).

Although information was collected as to the type of action (eg state, action-process (factive)), this was not used in analysis or synthesis, and will be ignored. The cases chosen were:

- **Agent** - The usually animate instigator of an action or process:
  - eg observations by American astronomers

- **Experiencer** - The animate entity that undergoes some change of
internal state when instigating an action or process:
  eg the hearing of machines by children

Patient - The entity on which an action is carried out:
  eg maintenance of satellites

Factive - The product of an action:
  eg the carving of statues

Beneficiary - The animate entity that receives some advantage or disadvantage from an action:
  eg teaching of mathematics to children

Instrument - The usually inanimate entity used to aid a process:
  eg driving of nails with hammers

Comitative - Designed to allow the expression of two-way relationships:
  eg foreign relations of France with Germany

It was not used in this project, and may be unnecessary for PRECIS.

Location - The locale of an action. It was recognized that this is usually expressed by the operator "0", but it was anticipated that it may occasionally turn up within a term.

From location - The starting point in space of an action. It was recognized that a temporal starting point may also be assigned this slot:
  eg probes from Earth

To location - The destination usually in space, but perhaps time, of an action:
  eg space flight to the Moon

Through location - The action of passing through, over or under a locale:
  eg journeys across the Sahara

For each case, three fields of information could be added. Firstly, there were the selection restrictions, which numbered nine:

Biotic
Animate
Human
Physical object
Abstract
Place
Potent
Unique
Predicate
These are on the whole self-explanatory, with the exception of the last. Some slots were typically filled by another case frame. This particularly applied to the instrument slot, which frequently takes in PRECIS a form similar to "use of word processors".

Each restriction could be coded as having value 0, 1 or 2;
where:
0 - not applicable
1 - possible, but not mandatory
2 - mandatory

It was recognized that these nine restrictions may not provide sufficient discrimination to differentiate between all cases, so another field was added which could contain a number of three character markers. These were envisaged as being similar to the markers used in SYSTRAN. These were not used in this project.

The third field contained the preposition or prepositions most associated with the slot. The rationale behind this was that at analysis certain prepositions would be associated with certain cases, and therefore ambiguity after matching of the selection restrictions could sometimes be resolved by the matching of prepositions. At synthesis the appropriate preposition for the case could be extracted from the case slot, if required.

The argument record included the eight selection restrictions applicable to the meaning:
Biotic
Animate
Human
Physical object
Abstract
Place
Potent
Unique

These were assigned values in the same way as were the restrictions in the predicate record. There was also provision for the extra markers described above. Again, a third data field included information about the prepositions. This time the dictionary-maker could specify which prepositions occurred with particular case relationships. This was a feature intended more for French, where one has heard it said (but never been able to trace it in print) that prepositional usage depends on the argument (ie the word after the preposition), rather than the predicate. In the version of the semantic analyser used for English, the prepositions attached to predicates were not used in the matching algorithms.

Finally, in the description of the dictionary, a note on multiple meanings. As has already been explained, some lexical units, such as "government" could have both an argument and a predicate record. So the two senses present in "The government of Britain" and "The policies of the British government" could be distinguished. If there were two meanings for an argument, or for a predicate, or within a predicate for a case slot, these could be entered in the dictionary, but not handled by the semantic analyser. This presented no problems in the sample used, but it would do so if the system were extended to the entire BLAISE database, although it is not possible to speculate on what scale. The best example discovered centred around the lexical unit "mechanics". To judge by the way in which PRECIS operators are applied in the BLAISE file, this may be both a predicate (ie a discipline that people do), or a quality that systems have. Unfortunately, it is applied to animate as well as inanimate systems. When a wider coding was allocated to indicate the "part" of the animate entity, as well as the quality that a device may
have, it produced a wrong analysis in the sample, because the analyser interpreted "mechanics" as an agent. The one consoling feature was that it provided an example of how an improvement to a dictionary in a less-than-perfect system could introduce degredation in performance.

From the previous discussion, the method used to determine the case information for the English dictionary may have been guessed. For each predicate and for many of the arguments, the BLAISE SIN file was consulted to ascertain what information should be included.

The programs

With the constraint on the size of individual programs, the semantic analyser had to be split into three parts. The first was a text preparation-type module; the second was the analyser proper; and the final one a kind of re-uniting module. Each will be described in turn.

Text preparation (Program MT25C1)

This program is listed and documented in Appendix I. It first established where the syntactic analysis information was held in the intermediate file, MTIXDI (100-147). The index to the source language analysis/synthesis dictionary was loaded (150-210). Thereafter each theme was processed in turn, with the results being written to another intermediate file, until all themes had been processed, when a general control record was written to the second file (967-968).

For each theme processed, the syntactic data had to be read into core storage; a process which included the concatenation of overlength data from trailer records (250-360). The downward and upward readings only from each theme were processed. That is to say that dates, "$n$" and "$o$" text, and lead-only terms had no part in semantic analysis.
In general terms, the processing of both readings followed the same lines, although the specific steps involved differed. For both readings, the limit of the text to be processed had to be determined. It was decided that only the text that made up the subject data, as opposed to pragmatic information, was to be passed to the semantic analyser. More concisely, any terms introduced by, or after, the operators "4", "5" and "6" were ignored.

Essentially this routine (which, because it processed a tree structure, was recursive) stored all "no", "nn", "np" and "ve" lexical units. The dependency of this semantic analyser on syntactic information is thus easily seen. Adjectives (which are usually considered to be significant in processing case) were ignored, although it was recognized that this could occasionally lead to anomalous results. For instance, string 407 contained in the downward reading, the terms "temperature" and "control". In the upward reading this was substituted by "thermal control". Because of the omission of adjectives, the entity being controlled was therefore left out. This was not felt to be unduly significant in that semantic analysis was intended for the resolution of preposition translations. The omission of adjectives only became significant if it led to a wrong analysis being assigned.

The lexical unit alone did not provide enough information whereby a good analysis could be made. The additional information judged necessary included the operator associated with the term in which the lexical unit occurred, together with the prepositions or conjunctions that occurred immediately before and after the lexical unit. So in the phrase from string 395, "Apollo Project compared with paintings of manned space flight by Smith, Ralph A", "flight" would have the prepositions "of" and "by" associated with it in the store. This process was not straightforward, in that it necessitated some movement through the levels of the tree structure to obtain the correct terminals.
The final information added to this store was the semantic information from the dictionary. This had to be retrieved from the dictionary, using a similar method to that used in syntactic analysis. It should be noted here that the dictionary search was completely deterministic, because the lexical units had already been determined in syntactic analysis.

After all structures in a reading had been processed, the data was written to an intermediate file, MT2XD1. After both readings had been processed, the next theme (if any) could be retrieved, or the final control record written, and the semantic analyser called.

All operations so far described were applicable to both downward and upward readings. The latter needed one extra routine. For reasons that will become clear below, in order for analysis to be satisfactory, the dependent operators "p", "q" and "g" had to be in the correct order following their main line operators. In an upward reading, this is obviously not so, because they are listed before the main line operator. The sequence (from string 327) that would be in effect:

$y2003 \text{ effect...}$
$yp103 \text{ elasticity}$
$yp103 \text{ core}$
$x1003 \text{ earth}$

was changed into:

$y2003 \text{ effect...}$
$x1003 \text{ earth}$
$yp103 \text{ core}$
$yp103 \text{ elasticity}$

This had unfortunate effects when the indexer had included a term to set the context in the downward reading, but had specified that it should be omitted from the upward reading. In the following (string 213)

$x0103 \text{ Wiltshire}$
$yp103 \text{ Amesbury}$
$y1003$ monuments $\$21$ henge $\$21$ megalithic
$y1032$
$yq103$ Stonehenge
$y2003$ use in simulation of solar system
the upward reading would be reduced by the inclusion of blank
upward reading substitute to:
$yq103$ Stonehenge
$y2003$ use in simulation of solar system
After processing by this routine, the actual store would be in
the state:
2 use
- simulation
- system
q Stonehenge
The semantic analyser interpreted "Stonehenge" as being a part of
"system". While one can argue that it is logically contained
within the solar system, this was not the message intended by the
indexer. The relationship between "use" and "Stonehenge" (that of
patient), was lost.

Some space has been devoted to the analysis of the
shortcomings of this routine. It could clearly be re-written to
produce more reliable results, perhaps with the deleted main line
operators being transferred to dependent operators in cases
similar to that above. It points to the fundamental weakness of
this program, which is that it is little more than a data
processing routine.

There are two possible approaches that would serve better,
and in the process, eliminate the need for this program. The
first would be to collect the significant items for the case
analyser by turning the RTN syntactic analyser into a lightly
augmented RTN. Thus when it encountered an appropriate item, it
would record them in a separate store. The second and more
elegant approach would be to impose semantic analysis on top of
syntactic processing. In this way, it would take the place of the
traditional syntactic features of subject, object and the rest. It should be noted that firstly this approach would be easier if the tree storage structure was more efficiently designed (as outlined in the previous Chapter); and secondly, it would take more core space, which would mitigate against its use with this particular minicomputer.

Semantic analysis (Program MT27Cl)

This program is listed and documented in Appendix J. The analysis is done in two stages. To understand why, it is necessary to refer to the systems of Wilks and of Somers. As was shown in Chapter II, Wilks used templates which had what were, essentially, subject, verb and object slots, although there was of course some fairly complicated matching undertaken to associate a portion of text with a template. Somers used some limited syntactic processing to isolate his case significant units (Somers, 1980a). If one can make a comparison between natural language processing and processing a restricted language like PRECIS, Somers and this project employed a similar level of syntactic analysis. Both Wilks and Somers were aided by their predicates being in the main (if not wholly) in the form of verbs (ie real verbs, rather than as defined in this project). This project had the problem of the predicates being almost entirely in the form of nouns and therefore syntactically impossible to distinguish from the arguments. For this reason the operators assigned to terms were used to isolate arguments and predicates.

Austin (1982) produced a thorough explanation of the PRECIS operators (applied to the subject rather than the pragmatic text) in terms of a case system. This relied on the classification of predicates by Chafe’s verb features described above. His description was not taken over and used wholesale for two reasons. The first was that he is concerned with the description of PRECIS as it should be done: a practical system has to be able to accept a certain amount on inaccuracy in the application of the operators. Secondly, Austin’s account refers to terms, and so
to use his system would imply that the unit of translation would be the term, rather than the lexical unit. It should be made clear that while he doesn't explicitly describe a procedure whereby the case categories could be mapped onto the string, it is implicit and very easy to envisage from his description. In passing, it should be noted that his use of terms and therefore his concentration on role operators leads him to define a slightly different set of cases, smaller in number. This is mainly because he did not feel the need to distinguish between the differing forms of location.

The parsing mechanism was again an RTN. The first parse grammar used in this project is shown in Figure 8.1. (A key to the meanings of the labels is given in Table 8.2.) Essentially, it tested two things: the role operator, and whether the lexical unit had been assigned the matching category of predicate or argument. All parses were recorded, using the backtracking facility of the RTN. A failure to produce an analysis led to error 511 being recorded, and the next reading (if any) being retrieved. A failure to assign a semantic structure did not lead to the string being rejected.

This process was controlled by the grammar. The next linguistic process was written as program. It was essentially another small text preparation routine, which examined each "FRAM" (ie case frame) uncovered, and wrote its predicate and thereafter its arguments to a store. Like the previous text preparation program (MT25C1), its function might have been better preformed by the use of an ATN to collect these items. Certainly the routines would have been more easily controlled, although again, space would have been a problem.

After the new store of items had been created, data contained therein was analysed according to the case system already described. The problem facing those who wish to use case in language processing is how to arrange for the individuality of
Figure 8.1
Semantic analysis grammar 1
the case frame to be reflected in a grammar. It is clearly useful to have a grammar that is external to the program. On the other hand, it is difficult to make it sensitive to particular case frames, because it is obviously necessary for the grammar to contain the amount of information needed to process the most extensive case frame. Again, how can a general grammar reflect the selection restrictions imposed on a slot for any given case frame?

The solution devised in this project was to add a very small number of actions to the grammar to be performed on the completion of a condition, over and above the action of building the parse structure. These actions were:

LOAD(FRAM) - This was attached to the "pred" test. It stored the current grammar, and then built a new grammar from the case frame of the predicate just subsumed. So if the case frame had the two slots, patient (marked with the selection restrictions, physical object (of value 2) and predicate (2)), and instrument (with the restriction of predicate (2)), the new grammar would assume the form:

```
FRAM
S1   pred S2 LOAD(FRAM)
S2   aP S2 WRIT(DONE)
PP   S2 WRIT(DONE)
PI   S2 WRIT(DONE)
*END
```

Here "aP" indicates an argument that is a patient and "PP" and "PI" both indicate subnets. The LOAD(FRAM) action would also create the subnets:

```
PP
S1   FRAM S2 ERAS(FRAM)
S2   *END
PI
S1   FRAM S2 ERAS(FRAM)
S2   *END
```

Here provision has been made for those case slots to be filled by
other frames. This clearly shows the recursive nature of the structures envisaged, and also points to the fact that this project was unable to avoid utterances that had more than one predicate.

**ERAS(FRAM)** - This was attached to the "FRAM" condition. As this was executed after the successful traversal of a subnet that was itself created from a case frame, this action stored the state of the current grammar (i.e., the one that controlled the processing of the subnet); and the reinstatement of the grammar used to call the subnet. This illustrates that this action and the LOAD action were devices to control the store (in this case the grammar store) during recursion.

**WRIT(DONE)** - This was not concerned with the control of stores during recursion, but merely to record that a particular slot had been filled. It was a necessary feature because (as already stated) each case could only be filled once in any frame. Once filled, it had to be marked as such, together with any variants. So in the example given above, when the patient slot was filled, both the options "aP" and "PP" were marked as done. It is because the grammar was adjusted to show the remaining options that reference was made to the "state" of the grammar.

On backtracking, these actions had to be undone, and a provision was made for this. It should be noted that the store that held the grammars was not a push-down stack, and therefore its control was not a matter of adding and removing from the top of the stack. This feature (which played havoc with analysis until properly mastered) would be of no concern to the writer of the grammar.

It may be argued that there is no writer of the grammar, in that it consists of just a few lines (Figure 8.2), and the program supplies the rest of the case frame. A sample constructed grammar is presented in Figure 8.3. On the other hand, the writer
MTC2G1
Case grammar
for PRECIS -
2nd parse
FULL
S1 FRAM S2 ERAS(FRAM)

Figure 8.2
Semantic analysis grammar 2
MTC2G1
Case grammar
for PRECIS -
2nd parse
FULL

S1  FRAM  S2  ERAS(FRAM)
  argl  S2
S2  FRAM  S2  ERAS(FRAM)
  argl  S2  
  *END
FRAM
S1  pred  S2  LOAD(FRAM)
S2  aA  S2  WRIT(DONE)

aP  S2  WRIT(DONE)
PP  S2  WRIT(DONE)
aI  S2  WRIT(DONE)
PI  S2  WRIT(DONE)
aL  S2  WRIT(DONE)

*END

**Figure 8.3**

A constructed grammar for "calculation"
does not have to use the actions which control the grammar. It may also be argued that the person who codes the original dictionary entry for a predicate is writing the grammar. It may fairly be said that the grammar for semantic analysis was largely external.

Apart from the execution of those actions that were necessary, the RTN analyser worked in exactly the same way as that used in syntactic analysis: indeed the coding was almost completely identical.

The matching algorithm was kept separate from the analyser proper. A more sophisticated process was needed than just the matching of a four character condition. Terminal items could be tested in two ways. Firstly, the correct category of either predicate or argument had to be satisfied. If the category was not an argument, it was tested against the whole of the four characters allowed in the condition. This applied only to "pred" and "arg1" (i.e. location argument of the type introduced by operator "O") in this project. Arguments were tested to determine if their selection restrictions matched those specified in the case slot. If successful, there was an option to test any prepositions associated with a case against the prepositions attached to the lexical unit. In the version of the analyser used, it was the prepositions attached to the case frame slot, although it could have been the prepositions added to the argument record. Calls on nets were also tested for prepositional requirements where appropriate, but this couldn't be done from the argument record.

The operation of the RTN analyser for this final analysis was different than previously described in one respect. Up to now backtracking had been used to uncover all readings. At this point, the program supplied to the RTN analyser each alternative set of units as uncovered by the first parse of the role operators. The analyser proceeded, testing for prepositions as
well as other features. If it completed a successful analysis, this was accepted immediately, without any others being constructed. It was originally intended to compare all analyses, but space proved to be a near insurmountable problem. Therefore the system worked on the principle of accepting the first adequate structure. If no structures were found, then the matching of prepositions was dropped as a requirement, and the first parse structures re-examined, using only the selection restrictions for matching. If an analysis still did not result, again error 511 was reported, and the next reading processed (if any).

On success, both the first and second parses were stored for use in the next program. If there were more readings to be processed, then the next was retrieved from the intermediate file. Otherwise the re-unification program was called.

Re-unification of parses (Program MT29Cl)

This program is listed and documented in Appendix K. If there were no results produced by the semantic analyser (MT27Cl), then the transfer module was called. Otherwise, each pair of first and second parses were taken in turn and run together to form a complete structure. The various parses for the downward reading of string 303 are given in Figure 8.4. Note here how the patient ("PAT" is a co-ordination of two arguments, "novae" and "supernovae"). Figure 8.5-8.6 shows the various upward readings of string 278. This shows clearly the recursive nature of the semantic structure, with two frames filling case slots. The complete parses were written to the intermediate file, MT2XD1. A pointer to each unit that featured in the semantic analysis was written onto the syntactic structures.

Some comments on the efficiency of the semantic analyser

That it operated on a first match principle was a disadvantage, although it would be difficult to improve on with the machine available. It was also very slow in comparison with
First parse

FULL
| | | | FRAM |

| | | | | | | ARG | PRED | ARG |
| | | | | | | | | | | | | | 1&A | g&A | 2&P | 3&A |

novae supernovae observation astronomers

Second parse

FULL
| | | | FRAM |

| | | | | | | pred | AP | aA |

observation supernovae astronomers

Unification of the first and second parses

FULL
| | | | FRAM |

| | | | | | | PRED | PAT | AGEN |
| | | | | | | | | | | | | | | | | | | | | pred | arg | arg | arg |

observation novae supernovae astronomers

Figure 8.4
Semantic analysis of downward reading of string 303
Figure 8.5
Semantic analysis of upward reading of string 278
Unification of the first and second parses

\[
\text{FULL}
\]

\[
\text{FRAM} \quad \text{FRAM}
\]

\[
\text{PRED} \quad \text{PRED} \quad \text{BENE}
\]

\[
\text{pred} \quad \text{pred} \quad \text{FRAM}
\]

photometry applications

\[
\text{PRED} \quad \text{PAT}
\]

\[
\text{pred} \quad \text{FRAM}
\]

determination

\[
\text{PRED} \quad \text{PAT}
\]

\[
\text{pred} \quad \arg
\]

distribution stars

Figure 8.6
Semantic analysis of upward reading of string 278
the syntactic analyser. This is attributable to two factors. The first is that three programs had to be called, which took a relatively long time, and because of limited storage space, intermediate results in the semantic analyser proper had to be written to a storage file. The second relates to the amount of searching that was carried out to find a parse. Two stage parsing certainly increased the time taken. The second parse was time consuming, because it was having to write and store grammars. It was also difficult to arrange for the LOAD(FRAM) action to write a grammar that would be heuristically elegant. In other words, the process of creating a network from a case frame did not mean that the most likely cases were going to be the first encountered, although it did mean that the grammar was tailored exactly to the frame being processed. Even so, the number of comparisons made in the matching algorithm could be very large. During debugging, a counter was added that revealed not the forty or so comparisons expected, but numbers in excess of two-hundred for a somewhat complicated string.

As regards the efficacy of the analyser, space precludes printing the entire set of results, but the results gained are summarized in Table 8.1, which is a listing of source language string numbers (see Appendix B). The four distinctions made are those analyses that were completely correct; those where the analysis of the relationships represented by prepositions were correct; those where the string was wrong and this was detected; and complete failures. The penultimate category is particularly interesting, and the faults included the use of operator "1" when "p" should have been used (strings 59 and 250) and the use of "p" when it should have been a "3" (string 66). The analysis of string 250 is noteworthy because while the downward reading failed, the upward was a success, because the latter included a substitute phrase which eliminated the term to which the incorrect operator had been applied, and thus the error was overcome.
<table>
<thead>
<tr>
<th>String Reading Correct Preps String Wrong</th>
<th>String Reading Correct Preps String Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 D /</td>
<td>102 D /</td>
</tr>
<tr>
<td>U /</td>
<td>U /</td>
</tr>
<tr>
<td>50 D /</td>
<td>107 D /</td>
</tr>
<tr>
<td>U /</td>
<td>U /</td>
</tr>
<tr>
<td>51 D /</td>
<td>116 D1 /</td>
</tr>
<tr>
<td>U /</td>
<td>U1 /</td>
</tr>
<tr>
<td>52 D /</td>
<td>12 D2 /</td>
</tr>
<tr>
<td>U /</td>
<td>U2 /</td>
</tr>
<tr>
<td>53 D /</td>
<td>125 D /</td>
</tr>
<tr>
<td>U /</td>
<td>U /</td>
</tr>
<tr>
<td>56 D /</td>
<td>137 D /</td>
</tr>
<tr>
<td>U /</td>
<td>U /</td>
</tr>
<tr>
<td>59 D /</td>
<td>138 D1 /</td>
</tr>
<tr>
<td>U /</td>
<td>U1 /</td>
</tr>
<tr>
<td>65 D /</td>
<td>138 D2 /</td>
</tr>
<tr>
<td>U /</td>
<td>U2 /</td>
</tr>
<tr>
<td>66 D /</td>
<td>141 D1 /</td>
</tr>
<tr>
<td>U /</td>
<td>U1 /</td>
</tr>
<tr>
<td>68 D /</td>
<td>141 D2 /</td>
</tr>
<tr>
<td>U /</td>
<td>U2 /</td>
</tr>
<tr>
<td>69 D /</td>
<td>145 D /</td>
</tr>
<tr>
<td>U /</td>
<td>U /</td>
</tr>
<tr>
<td>77 D /</td>
<td>148 D1 /</td>
</tr>
<tr>
<td>U /</td>
<td>U1 /</td>
</tr>
<tr>
<td>79 D /</td>
<td>148 D2 /</td>
</tr>
<tr>
<td>U /</td>
<td>U2 /</td>
</tr>
<tr>
<td>101 D /</td>
<td>153 D /</td>
</tr>
<tr>
<td>U /</td>
<td>U /</td>
</tr>
</tbody>
</table>

Table 8.1
Collected results of semantic analyses
<table>
<thead>
<tr>
<th>String Reading</th>
<th>Correct</th>
<th>Preps</th>
<th>String Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>154 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>172 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>198 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>207 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>213 D1 / U1 /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>236 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>242 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>248 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>250 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>278 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>292 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String Reading</th>
<th>Correct</th>
<th>Preps</th>
<th>String Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>303 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>305 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>320 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>327 D1 / U1 /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>336 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>350 D1 / U1 /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>352 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>354 D1 / U1 /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>395 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>407 D / U /</td>
<td>/</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.1 (continued)
Collected results of semantic analyses

<p>| 51 | 27 | 7 | 25 |
| 46.4% | 24.5% | 6.4% | 22.7% |</p>
<table>
<thead>
<tr>
<th>Program Name</th>
<th>Comment</th>
<th>Program Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st 2nd 3rd</td>
<td></td>
<td>1st 2nd 3rd</td>
<td></td>
</tr>
<tr>
<td>/ / / FULL</td>
<td>Root of structure</td>
<td>/ AGEN Agent</td>
<td></td>
</tr>
<tr>
<td>/ / / FRAM</td>
<td>Case frame</td>
<td>/ EXP Experiencer</td>
<td></td>
</tr>
<tr>
<td>/ / / ARG</td>
<td>Argument location</td>
<td>/ PAT Patient</td>
<td></td>
</tr>
<tr>
<td>/ / / PRED</td>
<td>Predicate</td>
<td>/ BENE Beneficiary</td>
<td></td>
</tr>
<tr>
<td>/ / ARG</td>
<td>Argument</td>
<td>/ FACT Factive</td>
<td></td>
</tr>
<tr>
<td>/ / VPAR</td>
<td>Part of a predicate</td>
<td>/ INST Instrument</td>
<td></td>
</tr>
<tr>
<td>/ / PART</td>
<td>Part of an argument</td>
<td>/ COM Comitative</td>
<td></td>
</tr>
<tr>
<td>/ PA Agent</td>
<td></td>
<td>/ LOCA Location</td>
<td></td>
</tr>
<tr>
<td>/ PE Experiencer</td>
<td></td>
<td>/ FROM From location</td>
<td></td>
</tr>
<tr>
<td>/ PP Patient</td>
<td></td>
<td>/ THRU Through location</td>
<td></td>
</tr>
<tr>
<td>/ PB Beneficiary</td>
<td></td>
<td>/ GLOC Argument location</td>
<td></td>
</tr>
<tr>
<td>/ PF Factive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ PI Instrument</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ PC Comitative</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ PL Location</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ Pf From location</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ Pe To location</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ Pt Through location</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.2
Table of labels used in semantic analysis - non-terminals
<table>
<thead>
<tr>
<th>Program Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st 2nd 3rd</td>
<td></td>
</tr>
<tr>
<td>/ 2&amp;P</td>
<td>Operator 2 and a predicate</td>
</tr>
<tr>
<td>/ 3&amp;P</td>
<td>Operator 3 and a predicate</td>
</tr>
<tr>
<td>/ s&amp;P</td>
<td>Operator s and a predicate</td>
</tr>
<tr>
<td>/ t&amp;P</td>
<td>Operator t and a predicate</td>
</tr>
<tr>
<td>/ 6P</td>
<td>No operator and a predicate</td>
</tr>
<tr>
<td>/ g&amp;P</td>
<td>Operator g and a predicate</td>
</tr>
<tr>
<td>/ 1&amp;A</td>
<td>Operator 1 and an argument</td>
</tr>
<tr>
<td>/ 3&amp;A</td>
<td>Operator 3 and an argument</td>
</tr>
<tr>
<td>/ 6A</td>
<td>No operator and an argument</td>
</tr>
<tr>
<td>/ 4&amp;A</td>
<td>Operator 4 and an argument</td>
</tr>
<tr>
<td>/ r&amp;A</td>
<td>Operator r and an argument</td>
</tr>
<tr>
<td>/ g&amp;A</td>
<td>Operator g and an argument</td>
</tr>
<tr>
<td>/ 6&amp;A</td>
<td>Operator 0 and an argument</td>
</tr>
<tr>
<td>/ p</td>
<td>Operator p</td>
</tr>
<tr>
<td>/ q</td>
<td>Operator q</td>
</tr>
<tr>
<td>/ g</td>
<td>Operator g</td>
</tr>
<tr>
<td>/ arg</td>
<td>Argument</td>
</tr>
<tr>
<td>/ argl</td>
<td>Location argument</td>
</tr>
<tr>
<td>/ pred</td>
<td>Predicate</td>
</tr>
<tr>
<td>/ aA</td>
<td>Agent argument</td>
</tr>
<tr>
<td>/ aE</td>
<td>Experiencer argument</td>
</tr>
<tr>
<td>/ aP</td>
<td>Patient argument</td>
</tr>
<tr>
<td>/ aB</td>
<td>Beneficiary argument</td>
</tr>
<tr>
<td>/ aF</td>
<td>Factive</td>
</tr>
<tr>
<td>/ aI</td>
<td>Instrument</td>
</tr>
<tr>
<td>/ aC</td>
<td>Comitative argument</td>
</tr>
<tr>
<td>/ aL</td>
<td>Location argument</td>
</tr>
<tr>
<td>/ af</td>
<td>From location argument</td>
</tr>
<tr>
<td>/ ae</td>
<td>To location argument</td>
</tr>
<tr>
<td>/ at</td>
<td>Through location argument</td>
</tr>
</tbody>
</table>

Table 8.2
Table of labels used in semantic analysis - terminals
References


Reprinted in


The objective of this module was transfer from English to French, using information produced by the analysis module. Three kinds of data were transferred. The first were the lexical units themselves; the second the syntactic structures; and the third the information relating to the operators and the semantic analysis.

To obtain an adequate translation, it was not acceptable to translate one lexical unit at a time. A large number of terms had a direct equivalence between two languages; for instance "astronomy" and "astronomie". But, for some single source language lexical units, there was more than one lexical unit in the target language, and obviously vice versa, as for "black holes" which was rendered as "cachots". In order to gain an idiomatic translation, it was occasionally necessary to include two or more lexical units in a transfer dictionary entry. "Ancient" was usually translated as "ancien", but when in the form "ancient world", it was translated as "monde antique".

Adjectives that are placed before their noun in French were labelled "ab". Some adjectives may appear both before or after their noun. It was decided that there would have to be two records in the French analysis/synthesis dictionary. The policy adopted in this project was to assume that if there were a direct translation from English to French, then that would be assigned the "ad" category. So "faux" in the sense of "treacherous" would stand as a single entry in the transfer dictionary. If the meaning of "worthy" or "artificial" was required, then the French would have to be linked to the noun in order to indicate that the "ab" version was required.

There were several other conventions used. Firstly, no
attempt was made to represent the target language nouns in only their plural or singular form, but in whichever was the appropriate translation from the English. Adjectives and "verbs" (i.e., past participles) were always held in their masculine singular form. Although some prepositions had (for the purposes of an indexing language) a direct equivalence between English and French (Verdier, 1980, p64), most did not. This module produced as an equivalence the most usual translation in a stylized form. So "of" was transferred as "prep(de)". The synthesis module included routines to choose between prepositions using the information gained in semantic analysis.

Finally, complex French nouns were treated as single units. So "étoiles" and "étôles binaires s'éclipsant" were both labelled as "no".

Strictly speaking, the complete source language syntactic structure was not transferred. For each target language lexical unit retrieved, an associated category label was also retrieved. These labels were the same as those for English, except for the "ab" class of adjective described above. So after lexical transfer, there could be a string of labels such as "ad no ad". These were assigned a target language syntactic structure by simply providing a list of valid sequences of categories, with an associated higher structure. So this example would become:

```
NP +{ad {*} +no {*}}
```

The symbol "{*}" represented the "slot" to be filled by the appropriate lexical unit. Should this seem familiar to the reader, it will be because it was derived from the TAUM project at Montreal (Le système de traduction..., 1973). Like TAUM, this system did not produce rooted, but free trees at transfer.

The third type of information was the pointers to the semantic information together with the position of the operator in the manipulation coding data. Both these types of information were usually associated with noun phrases, and occasionally with
the type of phrase labelled "VP" in this system. It was therefore relatively easy to copy information from, for instance, a source language to a target language noun phrase. It would have been easier if the tree structures produced by the syntactic analysis module had been better designed.

The dictionary

This is given in Appendix D. It was designed to hold the very minimum amount of information needed for adequate transfer. There was the source language lexical unit or units and their associated syntactic categories. These together were used as the key when searching. Associated with each key was some information relating to where the term was derived, if anywhere. Amongst the sources used in this project were the PRECIS Translingual Project lexicons, the Categories of forms publication already mentioned in previous Chapters, and an IFLA list of names of states for use in international bibliographic exchange. This information was included so as to facilitate any changes to this dictionary made necessary by changes of policy by any authority used in the building of the dictionary.

None of the entries used in the project needed multiple equivalents (the provision for using more than one lexical unit in an entry being sufficient to overcome any problems), and therefore no provision was made for choosing between multiple translations. The lack of multiple equivalents was due entirely to the control of word forms in PRECIS, and the accidental similarities of English and French. The target language lexical unit or units were stored in the same way as the source language, and again information about the source of the data was included.

The grammar

The grammar used for this project is shown in Figure 9.1. As will be seen, it consisted of two parts. The left-hand side consisted of the sequence of syntactic categories, whilst the right-hand side was the structure to be assigned.
Transfer (Program MT31Cl)

This program is listed and documented in Appendix L. The source string's manipulation code was copied to the intermediate file MT3XD1, which held the results of transfer (lines 100-149). This data was transferred because it was felt to be essentially (but not completely) common between languages. The English/French transfer dictionary keys (created in the same way as for the English analysis/synthesis dictionary) was loaded (400-445), as was the grammar (450-497). The first theme was then read from the intermediate file MT1XD1. The downward and upward readings were both translated using a text translating routine, as was text associated with any "$n" or "$o" operators that might be present. As previously stated, lead only terms were not translated. Date differences ("$d") were processed by a separate routine. Both translating routines included the storage of their results. If after successfully processing the current theme, there was another theme, this was processed; otherwise the synthesis module was called.

The text translation routine processed each syntactic tree structure of a reading in turn (2100-2198) and these were decomposed from rooted to free (ie rootless) trees, using a recursive routine (2200-2235). After the complete reading had been decomposed, the semantic information and the location of the manipulation coding was copied from the terminal items onto associated "NP" and "VP" non-terminals (2230-2275). This program routine was similar in function to the "RECOP" grammar of GETA's ARIANE transfer module (Boitet and Nedobejkine, 1981, p239).

The dictionary was then searched, attempting to retrieve the longest match. In order to obtain the correct translation, a backtracking mechanism was included. To illustrate this, let a dictionary be:
manned
manned space
space
space vehicles
and let the text to be translated be "manned space vehicles". The
search would first find the longest match present (ie "manned
space"). The search would then continue for "vehicles", which is
not present. The search would backtrack and attempt to find a
shorter match, which it would find. The next search would be for
"space vehicles", which again is present. It should be noted
that, as with all MT transfer routines, the method of dictionary
search did not guarantee that the required translation would be
produced if the dictionary had not been very well constructed. A
failure to find a complete target language reading caused the
error number 552 to be reported and the processing of the current
string was halted.

On success the grammar was searched and if successful, the
target language lexical units were added to the target language
structure to form a free tree. If the grammar search was
unsuccessful, error 551 was reported and the processing of the
string was ended. The source language semantic information and
operator pointers were added to the target language trees. On
completion, the tree was written to the intermediate file,
MT3XD1.

The translation of dates was very simple, and moreover was
carried out without analysis information as such. There was a
very small date dictionary, which consisted of:
A.D.| ap. J.-C.
B.C.| avant J.-C.
cal| ca
tol| jusqu'à
This dictionary was loaded, and a pointer set to the first
character of the date text. Any figure or hyphen was copied
across without alteration. Text was searched for in the
dictionary, and on success the target language equivalent was
copied. If the search was unsuccessful, the error number 562 was
reported and the processing of the string ended. After the whole
of a data was processed, it was cleaned (i.e., double spaces and space after a hyphen were removed), and stored.

The product of the transfer module was rather poor French, with adjectives in the wrong place, prepositions only provisionally translated, and articles lacking (as can be seen from Figure 9.2). It was the purpose of the synthesis module to correct these shortcomings.
Figure 9.2
Output from the transfer module
References


CHAPTER X
GENERATION

Generation consisted of two distinct parts. First there was the revision of the output of the transfer module to form grammatically and idiomatically acceptable French. The second was the creation of target language PRECIS strings. This second process was the antithesis of the first module of the system, the text preparation module, which broke up the source language PRECIS strings. As was text preparation, so was this process essentially an exercise in data processing rather than computational linguistics.

In keeping with the objective that the system should be readily extensible to languages other than French, it was necessary to make the programs as language independent as possible. This was achieved reasonably well for the first of the two processes, but lack of time meant that the second process was not programmed in depth. As it stood at the end of the project, it was language dependent, but with further elaboration, may not have been so.

The first process (the revision of the output of the transfer module) was designed as a single program. The smallness of the work areas of the machine used meant that the conception became, in implementation, two essentially similar programs. Both used the same finite state interpreter to change the data structure received from the transfer module, according to rules of the grammars.

The grammars were cast in the form of simple production-type rules; which is to say that they took the form of actions to be performed on the data structure, dictionaries or grammars providing that specified conditions were met. The syntax of the rules was limited to a greater extent than would be feasible for
natural language, in that conditions such as:

IF condition-a AND (condition-b OR condition-c) THEN...

were not accepted.

The simplest form of rule was:

(S1 IF condition-a THEN action-a S2
  ELSE action-b S2)

As in the analysis modules, the states within the grammars were labelled in the series S1, S2, S3 .... Here "S1" indicates the state name (or rule calling name), and "S2" the name of the next rule to be tested. Actions ranged from fairly complex operations on trees, to the simple instruction to jump to a specified state without changing the data structure. It should be noted in passing that no backtracking facility was included. The most complicated rules could have several parts:

(S1 IF condition-a AND condition-b THEN action-a S2
  OR condition-a AND condition-c THEN action-a S2
  OR condition-a THEN action-b AND action-d AND ...
      action-n S3
  ...
  OR condition-n THEN action-n
  ELSE action-p S4)

The grammar was originally designed in the form of one control grammar which could call sub-grammars, which could in turn call their own sub-grammars, and so on. These sub-grammars were intended to include rules about the behaviour of lexical units drawn from the lexical records of individual units. Again, restrictions on the size of core space meant that this facility could not be implemented, although this was to some extent ameliorated by allowing grammars to chain into each other (rather than calling others as sub-grammars); and by allocating a field of the lexical record as a "free text" area, which could be tested by the rules.
The grammars were written using the operators illustrated above, together with a set of conditions and actions that will be described below. In order to simplify the writing of the programs and to economize on core space, the grammars were condensed, mainly by converting the operators and the beginnings of conditions and actions into one-character codes. This was done using a simple program, MTGCCl, which for the sake of completeness is presented in Appendix M.

The two programs used for the first part of generation will be described, and thereafter the manner in which they were used will be given, together with the grammars.

French generation 1 (Program MT41Cl)

This is listed in Appendix N. The program automatically called the first grammar which had the name "MTFSCl" (the contracted version of MTFSCl) (line 100), and set the pointer to the first state name (line 300). The main control routine had several parts. First the current state name had to be found in the grammar (lines 350-365), reporting error number 581 if it was not found. "IF" and "OR" conditions were evaluated by a routine which included a number of sub-routines which themselves evaluated the individual condition specified. On failure, the next "OR" statement was found, or if these had been exhausted, then the "ELSE" action was carried out. Each left-hand side of the rule could have more than one condition, joined using the "AND" operator. The inclusion in the grammar of a condition that the program did not recognize triggered error 582, and halted the processing of the current string (lines 400-735).

On successful completion of the left-hand side of a rule, the "THEN" operator was found and the action or actions following it were performed. Again the failure of the program to recognize a portion of the grammar led to error 582 being reported and processing of the current string being stopped (lines 750-895).
On completion of all actions specified, the next state to be processed was identified, except in the case of those actions that loaded a new grammar or called a different program (lines 900-970). Having found the next state name, the rule had to be located, as did the first state, as described above.

At the risk of presenting a slightly less intelligible account, the conditions and actions will be given separately, rather than as they would have been called by a typical grammar.

**Conditions**

These are divided into those that tested the data structure itself and those that were concerned with either the pointers to the data structure, or with grammars or dictionaries. The latter will be described first. The letter in parentheses after the name of the condition or action is the code used when the grammars were contracted.

**DICT=0 or DICT<>0 (d) Lines 2100-2185** - Generation used the same method of storing an index to the dictionary as did the analysis and transfer modules. This condition was used to test if the dictionary index had already been loaded.

**MORE-TL-TREES (m) Lines 2200-2298** - The program was designed to hold one reading (ie downward, upward or parenthetical difference reading) in core at a time. Within each reading there could be one or more "phrases". For the purposes of the commands and actions of this module, a "phrase" was equivalent to one line of a decomposed tree produced by the transfer module, and of which an example is given in Figure 9.2. This condition was used to test for the presence of another reading for the current string. Date differences were not processed by this program nor the next, MT43C1, as the product of the transfer module was satisfactory. If the next reading was found to be a date, this was copied from the file that held the output of the transfer module (MT3XD1), to that which held the output from this module (MT4XD1). The
transfer data was then re-examined, as if no file writing had taken place.

MORE-PHRASES (p) Lines 4050-4080 — This condition was used to test if there were any more phrases to be processed within a reading.

UNIT=*END or UNIT<>*END (u) Lines 4000-4030 — This was used to find out whether or not the current syntactic unit (which could be either a phrase unit such as "NP" or a terminal such as "pr") was the last in the current phrase.

TREE= (t=) Lines 3300-3835 — This condition was designed to test the current state of the phrase being processed. The problem encountered in designing the format of the rule was to avoid making the writing difficult by the use of a complicated form, while trying to produce a format that would not involve a large amount of program code for its interpretation, and so use valuable core storage. The format used was modelled closely on the form of the data structure.

Essentially the rules were cast in the form:

```
TREE=PCATl+{tcat2{lexical unit}}!
```

"PCAT" is used here to represent a non-terminal syntactic label, and "tcat" a terminal. The bracketing is as it would be on the data structure. The exclamation mark was used to indicate the end of the condition. The terminal and non-terminal labels could be drawn from the repertoire used for the target language. If the writer of the grammar wished not to specify an individual label, one of two "blanket" labels could be used. "#CAT" was used to represent any non-terminal, and "#cat" any terminal label.

For each terminal label, an associated lexical unit could be specified; for instance:

```
pr 1{en}
```

It would obviously not be very convenient to have to list in the
grammar all the lexical units that could fulfill a condition. Therefore two further facilities were added. The lexical unit could be made unconditional as in this example:

```
pr 1[\#]
```

or it could be made partially conditional in one of three ways. The beginning could be conditional:

```
no 1[\#abc]
```

or the end could be conditional:

```
no 1[abc\#]
```

or the middle could be conditional:

```
no 1[\#abc\#]
```

The regime of a phrase could be made unconditional by the use of "{\#}". If the data structure was:

```
NP +{no {vol}+ad {spatial}}
```

this could be represented by:

```
NP 1+{\#}
```

Finally, conditions could be added to the labels to test for the presence or absence of a number of features held in the synthesis data of the lexical record of a unit. For French, there were three specific conditions which each had separate fields in the dictionary format. First was gender, which was set down in the rules in the form:

```
GEN=mas or GEN<>fem, etc
```

The second was surface number:

```
NUM=pl or NUM<>si or NUM<>in (ie invariable), etc
```

The third condition was whether or not the unit being tested had a vowel start or not:

```
VWL=V or VWL<>V
```

The grammar contracting program substituted the field markers for the name of the condition, so that "GEN" became "$1", "NUM" became "$2", and "VWL" became "$3".

The fourth field in the synthesis data indicated whether or not the item should be allocated a lead in the printed index, and
therefore had no part to play in the linguistic processes being described. The fifth field was originally intended to hold one or more sub-grammars as noted above. In an attempt to overcome the difficulties of representing the idiomatic behaviour of a lexical unit without having to write it into the main grammars, this field was designed to store free text conditions that the writer of the grammars thought pertinent. In fact this field was not used in the current project, although there were probably cases when it should have been, such as to signal that certain nouns should not take an article.

Finally, the inclusion of numbers after the lexical units should be explained. To simplify later processes specified by the action called "SUB TREE=" each syntactic unit was loaded into a row of a string matrix. On the result of a successful comparison, the "SUB TREE=" action carried out its operations on the string matrix store, before concatenating the results, and inserting them into the data structure. This method allowed, in particular, the easy specification of the re-ordering of adjectives from English to French order.

An example of the use of all the facilities of this condition is difficult to conceive. Indeed, as will be seen from an examination of the grammars given below, the full range was not used in this project. A slightly forced example can be made from the exceptional inclusion of an article after "en", providing that the next word is a masculine noun with a vowel start (Grevisse, 1969, p.957). The output required would be, for example:

...pr {en}+NP +{de {l'}+no {air}...

whereas the output from the transfer module would typically be:

...pr {en}+NP +{no {air}...

A rule to test for the presence of the anomalous condition would be written as:

TREE=pr 1{en}+NP 2+{no 3[NUM=s1/GEN=mas/VWL=V/ $5<ART=0]}{#}!
This means that the condition to be satisfied is the preposition "en", followed by a noun phrase, which has as its first constituent, an ordinary noun which is singular and masculine, with a vowel start, and that has not been marked as not taking an article.

**SEM-TREE=NULL or SEM-TREE<>NULL** (8) Lines 2600-2680 - For the process of choosing the correct form of preposition, recourse was made to the semantic analysis of the reading. The writer of the grammar had no way of telling in advance if there was a semantic analysis for a string, without this test. It was used after the action to load the semantic analysis, and checked if one had been retrieved.

The final three conditions allowed in this program were concerned with the choice of prepositions. Essentially there were three places from which prepositions could be drawn. Where a semantic analysis had been produced, the writer of the grammar could test for the presence of a preposition in both the argument and predicate records. In both these records, the case relation could be marked with one or more prepositions. This facility has already been described in the account of the matching procedures in semantic analysis (page 161). It was by this means that an exceptional usage connected to a particular relation and lexical unit could be used. The other source of prepositions was from the "usual" translation given by the transfer dictionary. It was intended that these "tree prepositions" would function as a safety net, should tests and actions on the semantic-based prepositions fail.

**ARG-PREP=O or ARG-PREP<>O** (not contracted) Lines 2700-2815, 2950-3010 - The case relation that the preposition represented was retrieved from the semantic analysis; the lexical unit of the argument associated with the case relation was retrieved, and the semantic information associated with the syntactic label present on the target language tree found. If the semantic unit type was
an argument (as opposed to a predicate such as "influence of astronomy"), the preposition field associated with the case relation was found. If it was empty, then obviously a negative reply to the condition was returned. Otherwise, the first preposition recorded in the field was added to a temporary store, ready for use by the "PUT" action.

The chief weakness of the preposition handling routines was present in this condition. No allowance was made in this routine to find prepositions in a second predicate which was used to fill a case slot like this:

```
FULL
|  FRAM
|   |  PRED  BENE
|   |  utilisation  FRAM
|   |   PRED
|    astronomie
```

Indeed, there was no data included in the synthesis area of the lexical record for marking case relations with prepositions. Therefore the idiomatic usage of "en" with the name of a subject field or discipline could not be constructed by the ARG-PREP condition, and this explains the presence in the final index of phrases such as "utilisation dans l'astronomie", instead of "en astronomie".
PRED-PRED=O or PRED-PREP<>O (not contracted) Lines 2850-3010 - Again, this retrieved the case relation, associated lexical unit and semantic data. With this data, the case frame was found, and then the case slot. If there was no preposition given, a negative reply was returned; otherwise the first preposition marked was added to a temporary store (as with the ARG-PREP) and a positive reply returned.

TREE-PREP=O or TREE-PREP<>O (not contracted) Lines 3200-3240, 2950-3010 - This tested the current lexical unit on the data structure, which was expected to be taken from the transfer dictionary in the form: "prep(dans)". On success, the preposition was loaded into the temporary store.

This concludes the description of the conditions allowed in program MT4IC1. The actions are classifiable in the same way as the conditions.

Actions
LOAD DICT=naae,volm.e (D) Lines 5200-5295 - This loaded an index to a dictionary specified by the writer of the grammar. The first record of the data contained the name of the file indexed, and after the loading, the dictionary proper was opened.

LOAD NEXT-TL-TREE (T) Lines 5300-5590 - On the first call upon this action, the manipulation codes of the current string were copied from the intermediate file MT3XD1 to MT4XD1, and the file that was used to hold lexical records retrieved by the "LOAD RECORDS" action cleared, together with the core storage area used for the current semantic analysis. The next theme to be processed was read into the main storage area.

LOAD RECORDS (R) Lines 5600-5898 - The purpose of this action was to retrieve from the generation language analysis/synthesis dictionary (which had already been opened by the "LOAD DICT=" action), the subset of lexical records pertinent to the current
theme. These records were held in two parts. The complete records
were held in a work file, "MT5XD1". A list of the keys to each
record and the syntactic labels together with a pointer to the
relative record number of the lexical record in the work file,
was held in core storage. Therefore, this action proceeded
through each tree in the data structure, retrieving each lexical
unit and checking that the syntactic category present on the data
structure was also present in the dictionary entry. A failure to
find a lexical unit led to error 583 being reported, and the
processing of the current string being ended.

LOAD SEM-TREE (S) Lines 5900-5998 - If the writer of the grammar
decided that tests on the semantic analysis were necessary, then
this action allowed the retrieval of the analysis of the current
data, if one existed. This was used in conjunction with the "ARG-
PREP" and "PRED-PREP" conditions.

SET PHRASE= (p) Lines 6100-6177,6350-6397 - This took either the
form of a direct setting, "SET PHRASE=2" (ie set the phrase
pointer to the second tree on the data structure); or a relative
setting, "SET PHRASE=+1" (ie set the phrase pointer to the next
tree on the data structure).

SET UNIT= (U) Lines 6100-6345 - Like the "SET PHRASE=" action
above, this allowed both a direct setting, "SET PHRASE=1" (which
set the unit pointer to the first syntactic category, rather than
the first lexical unit, on the tree); and a relative setting. The
latter could take two forms, either a forward or backward motion,
as in "SET UNIT=+1" and "SET UNIT=--1".

PUT (P) Lines 6500-6535 - This took the data held in the
temporary store used by "ARG-PREP", "PRED-PREP" and "TREE-PREP"
conditions, and substituted it for the lexical unit at the
current pointer settings.
WRITE TREES (W) Lines 6600-6715 - This wrote the index to the lexical records retrieved by the "LOAD RECORDS" action into the intermediate file, MT5XD1. The current theme was then written to the intermediate file, MT4XD1.

GOTO (C) Lines 6000-6010 - This action was a "jumping" device, which allowed the specification of the new state in the grammar which was to be progressed to, without any other consequent actions being performed.

CHAIN (C) Lines 6050-6065 - This term was borrowed from the Digico BASIC's statement set. It allowed the grammar writer to choose the next program to be run. It took the form of "CHAIN 'filename,backing store volume'"; eg "CHAIN 'MT42C1,2771'".

ERROR= (E) Lines 6400-6425 - The grammar writer may decide that a failure to satisfy one or more conditions represents an unacceptable state in the data structure. For instance, a failure to find a preposition to insert after the application of "PRED-PREP", "ARG-PREP" and "TREE-PREP" conditions could leave an unacceptable form on the tree, and an error condition could be enforced here. The exact form was "ERROR='number'", such as "ERROR='585'". This would record the error, and halt the processing of the current string.

French generation 2 (Program MT43C1)
This is listed in Appendix O. In between this program and MT41C1 was a very small program that copied the data in the intermediate file MT4XD1 back into MT3XD1, so as to enable the routines that dealt with the intermediate files in MT41C1 and MT43C1 to be kept identical. The main processing routine was the same as for MT41C1.

Conditions
The conditions allowed in this program were mostly the same as for the previous. "ARG-PREP", "PRED-PREP" and "TREE-PREP" were
omitted, and "PHRASE-" was added.

**PHRASE- (p) Lines 4100-4250** - This was written in the form:

\[
\text{PHRASE=} \text{NP } [\text{conditions}] \text{ or PHRASE=} \text{VP } [\text{conditions}]
\]

The conditions could include the "NUM" and "GEN" conditions described above in the account of the "TREE-" condition. The purpose of the provision of this condition was to enable general tests on the surface number and gender of either a noun or verb phrase. Essentially it functioned by finding the appropriate nominal and testing its characteristics. This cumbersome method would not have been necessary if the design of the data structures at the analysis and transfer stages had been better, because such information would have been added to the appropriate "NP" and "VP" labels.

**Actions**

The actions were also largely the same as for the first program. "PUT" and "LOAD SEM-TREE" were omitted, and "LOAD GRAMMAR" and "SUB TREE-" were added.

**LOAD GRAMMAR (g) Lines 7300-7315** - By using this action, the grammar writer could specify which grammar was to be called. The format was:

\[
\text{LOAD GRAMMAR=} \text{ABCDE,1234}
\]

where "ABCDE" represents the backing store name of the grammar, and "1234" the volume number of the disc on which it is held. After the specified grammar had been loaded, the state pointer was set back to the first state, "SI". The pointers to the phrase and unit currently being processed were not reset however, which could enable sub-grammars to be called for one particular syntactic label.

**SUB TREE- (t) Lines 6800-7270** - This action should not be confused with the condition of the similar name. It was anticipated that it would customarily be used in conjunction with the "TREE-" condition, and there would thereby be a string matrix
holding a representation of part of the data structure. So if the data had been:

\[ PP +{pr \{pour\}+NP +{no \{geologie\}} \]

and the condition had been:

\[ TREE=pr \ 1{\#}+NP \ 2+{no \ [GEN=fem]{\#}}! \]

then the matrix would contain:

<table>
<thead>
<tr>
<th>Row</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>pr {pour}</td>
</tr>
<tr>
<td>2</td>
<td>NP</td>
</tr>
<tr>
<td>3</td>
<td>no {geologie}</td>
</tr>
</tbody>
</table>

A typical action would be one to insert an article:

\[ SUB \ TREE=pr \ 1{\#}+NP \ 2+{de \ 0{la}+no \ {\#}}!* \]

The program would work through each syntactic label of the new tree, checking on the value of the "load number" (ie the number given after the lexical unit). If the load number was greater than zero, the data from that row of the string matrix would be added to a string holding the data to be inserted; or if it were zero it would simply copy the syntactic label and its regime from the condition into the string. On encountering the exclamation mark, processing would then be finished, and any items left unused in the string matrix abandoned. The string in which the new tree was concatenated was added to the data structure in place of that found by the "TREE=" condition. The asterisk at the end of the action was a command to the program not to retrieve from the analysis/synthesis dictionary records for any of the lexical units added.

So in the case of the example given above, the resulting data structure would be:

\[ PP +{pr \{pour\}+NP +{de \ {la}+no \ {geologie\}}} \]

This action was designed to enable individual lexical to be altered morphologically, without having to list the complete old and new forms. A general format was used which had two parts:

\{old form%new form\}
It is rarely that one would wish to alter complete lexical units, so the same system of partially conditional and wholly conditional tests as were used for the "TREE=" condition was introduced.

An action in the following form would succeed a test for a feminine phrase:

\{ad 1{#x%^se}\}

The "x" at the end of an adjective such as "heureux" would be removed and "se" added to make the form "heureuse". As with the associated condition, the beginning and middle of a lexical unit could be changed in the same way. Provision was made to change the lexical unit’s key in the index to the lexical records to its new form.

This concludes the description of the two programs. It remains to describe the grammars used in this project.

The lack of workspace was most critical in the generation module. It has already been stated that the one process had to be split between two programs. Even after this, there was very little room left for the grammars, and it was necessary to split them into no less than seven individual files. The first, "MTFSG1" was used by the first program, MT41C1. It contained a command to chain the second part of the module which used the remaining six grammars.

The grammars were organized into a chain so as to carry out the necessary processes in as logical order as possible. For instance, it was necessary to re-order noun-phrases before applying the article insertion grammar. A schema of the application of the grammars is given in Figure 10.1.

**Grammar MTFSG1 (Figure 10.2)**

This will be explained in more detail than the rest, as it contains many of the more frequently used conditions and actions.
Figure 10.1
A schema of the application of generation grammars
MTFSG1

((S1 IF DICT=0 THEN LOAD DICT=MTFAH1,2771 S2
  ELSE GOTO S2)
(S2 IF MORE-TL-TREES
  THEN LOAD NEXT-TL-TREE AND LOAD RECORDS AND SET PHRASE=1
  AND SET UNIT=1 S3
  ELSE CHAIN 'MT42C1,277)
(S3 IF TREE=pr 1{prepθ}! THEN LOAD SEM-TREE S4
  ELSE GOTO S7)
(S4 IF SEM-TREE<>0 THEN GOTO S5
  ELSE GOTO S6)
(S5 IF ARG-PREP<>0 THEN PUT ARG-PREP S7
  OR PRED-PREP<>0 THEN PUT PRED-PREP S7
  ELSE GOTO S6)
(S6 IF TREE-PREP<>0 THEN PUT TREE-PREP S7
  ELSE ERROR='585‘)
(S7 IF UNIT<>* THEN SET UNIT=+1 S3
  ELSE GOTO S8)
(S8 IF MORE PHRASES THEN SET PHRASE=+1 AND SET UNIT=1 S3
  ELSE WRITE TREES S1))

Figure 10.2
Grammar MTFSG1
State S1 tested if the dictionary index had been loaded. If not, then the index with the backing store name "MTFAHl" was loaded and the pointer set to state S2. If the index was present, then state S2 was sought without any other actions taking place.

S2 tested for more data to process. If there was more, the next reading was loaded and its lexical units' dictionary entries retrieved. The phrase and unit pointers were set to the first item in the reading, and control passed to state S3. Otherwise the next program, "MT42C1" was called and thereby the processing using this grammar stopped.

S3 tested the current syntactic label to determine if it was a preposition, and the lexical unit to determine if it was a "safety net" preposition from the transfer dictionary. If so then the semantic analysis (if present) was loaded and the state set to S4. If the data structure was not as specified by then "TREEc" condition, the S1 was progressed to.

S4 tested if there was in fact a semantic analysis for this reading. If there was then the state S5 was sought, otherwise it was S6.

S5 first tested for the presence of a preposition for the particular case represented in the argument record. This was put first because of the apparent habit of French to determine prepositional usage by the argument (see page 149). If present, this preposition was "PUT" into the date structure, and state S7 called. On failure of this condition, the predicate preposition was tested, and if successful, was added to the tree and state S7 called. Failure with both of these conditions led to state S6 being called.

S6 tested for the presence of "safety net" preposition. On success, S6 was called, otherwise error number 585 was recorded.
MTFSG2

{(S1 IF DICT=0 THEN LOAD DICT=MTFAH1,2771 S2
    ELSE GOTO S2)}

(S2 IF MORE-TL-TREES
    THEN LOAD NEXT-TL-TREE AND LOAD RECORDS AND SET PHRASE=1
    AND SET UNIT=1 S3
    ELSE CHAIN 'MT45C1,2771 S2)

(S3 IF TREE=PP 1+(pr 3{a\}+NP 3+{#}+co 4{#}+NP 5+{#}) S4
    THEN SUB TREE=PP 1+(pr 2{a\}+NP 3+{#})+co 4{#}+PP 0+{pr
    0{a\}+NP 5{#}) S4
    OR TREE=PP 1+(pr 2{de}+NP 3+{#})+co 4{#}+NP 5+{#1
    THEN SUB TREE=PP 1+(pr 2{de}+NP 3+{#})+co 4{#}+PP 0+{pr
    0{de}+NP 5{#}) S4
    OR TREE=PP 1+(pr 2{en}+NP 3+co 4{#})+NP 5+{#}) S4
    THEN SUB TREE=PP 1+(pr 2{en}+NP 3+{#})+co 4{#}+PP 0+{pr
    0{en}+NP 5{#}) S4
    ELSE GOTO S4)

(S4 IF UNIT<>*END THEN SET UNIT+=1 S3
    OR MORE-PHRASES THEN SET PHRASE+=1 AND SET UNIT=1 S3
    ELSE SET PHRASE=1 AND SET UNIT=1 AND LOAD GRAMMAR=MTFSC3,2771)

Figure 10.3
Grammar MTFSG2
S7 tested for the presence of further units in the current phrase. If there were more, then the unit pointer was incremented by one, and S3 called. If there were more phrases then the phrase pointer was incremented by one, and the unit pointer set to the first syntactic label of the new phrase and S3 progressed to. Otherwise, the reading was finished and it could be stored, and the transfer data tested for further readings by calling state S2.

**Grammar MTFSG2 (Figure 10.3)**

This was concerned with the co-ordination of nouns. Usually there was no problem, as the co-ordination was not part of a prepositional phrase. When it was, however, there were circumstances when the repetition of the preposition was required. This condition occurred with the prepositions "à", "de" and "en". It should be noted that the bracketing of the resulting tree was taken from the "SUB TREE=" action.
MTFSG3

\[
\begin{align*}
(S1 & \text{ IF } \text{TREE} = \text{NP } 1+\{\text{ad } 2\{\#\}+\text{ad } 3\{\#\}+\text{ad } 4\{\#\}+\text{no } 5\{\#\}\} \} \! \\
& \text{ THEN } \text{SUB} \text{TREE} = \text{NP } 1+\{\text{no } 5\{\#\}+\text{ad } 4\{\#\}+\text{ad } 3\{\#\}+\text{ad } 2\{\#\}\} \} \} S2 \! \\
& \text{ OR } \text{TREE} = \text{NP } 1+\{\text{ad } 2\{\#\}+\text{no } 3\{\#\}+\text{ad } 4\{\#\}\} \} \! \\
& \text{ THEN } \text{SUB} \text{TREE} = \text{NP } 1+\{\text{no } 3\{\#\}+\text{ad } 4\{\#\}+\text{ad } 2\{\#\}\} \} S2 \! \\
& \text{ OR } \text{TREE} = \text{NP } 1+\{\text{ad } 2\{\#\}+\text{no } 3\{\#\}\} \} \! \\
& \text{ THEN } \text{SUB} \text{TREE} = \text{NP } 1+\{\text{no } 4\{\#\}+\text{ad } 3\{\#\}+\text{ad } 2\{\#\}\} \} S2 \! \\
& \text{ OR } \text{TREE} = \text{NP } 1+\{\text{ad } 2\{\#\}+\text{no } 3\{\#\}\} \} \! \\
& \text{ THEN } \text{SUB} \text{TREE} = \text{NP } 1+\{\text{no } 3\{\#\}+\text{ad } 2\{\#\}\} \} S2 \! \\
& \text{ OR } \text{TREE} = \text{NP } 1+\{\text{ad } 2\{\#\}+\text{np } 3\{\#\}\} \} \! \\
& \text{ THEN } \text{SUB} \text{TREE} = \text{NP } 1+\{\text{np } 3\{\#\}+\text{ad } 2+\{\#\}\} \} S2 \! \\
& \text{ ELSE GOTO } S2 \}
\end{align*}
\]

(S2 IF UNIT>**END** THEN SET UNIT=+1 S1 \\
OR MORE PHRASES THEN SET PHRASE=+1 AND SET UNIT=1 S1 \\
ELSE SET PHRASE=1 AND SET UNIT=1 AND LOAD GRAMMAR=MTFSG4,2771))

Figure 10.4
Grammar MTFSG3
Grammar MTFSG3 (Figure 10.4)

This re-ordered the constituents of noun phrases, if necessary. This grammar in particular illustrates the use of the "load numbers" in both the "TREE=" condition and "SUB TREE=" action.
MTFG4

((S1 IF TREE=pr 1{#}! THEN GOTO S2
   ELSE GOTO S4)
(S2 IF TREE=pr 1{en}! THEN GOTO S4
   OR TREE=pr 1{au}! THEN GOTO S4
   OR TREE=pr 1{aux}! THEN GOTO S4
   OR TREE=pr 1{du}! THEN GOTO S4
   OR TREE=pr 1{des}! THEN GOTO S4
   OR TREE=pr 1[a\partir\de]! THEN GOTO S4
   ELSE SET UNIT=+2 S3)
(S3 IF TREE=de 1{#}! THEN GOTO S4
   OR TREE=nn 1{#}! THEN GOTO S4
   OR TREE=#cat1{1e #}! THEN GOTO S4
   OR TREE=#cat1{1a #}! THEN GOTO S4
   OR TREE=#cat1{les #}! THEN GOTO S4
   OR PHRASE=NP [NUM=pl/J AND TREE=#cat1{#}]!
      THEN SUB TREE=de 0{1es}+#cat1{#}!* S4
   OR PHRASE=NP [GEN=fem/J AND TREE=#cat1{#}]
      THEN SUB TREE=de 0{1la}+#cat1{#}!* S4
   OR TREE=#cat1{#}! THEN SUB TREE=de 0{1e}+#cat1{#}!* S4
      ELSE GOTO S4)
(S4 IF UNIT>1*END THEN SET UNIT=1 S1
   OR MORE PHRASES THEN SET PHRASE=1 AND SET UNIT=1 S1
   ELSE SET PHRASE=1 AND SET UNIT=1 AND LOAD GRAMMAR=MTFGS5,2771))

Figure 10.5
Grammar MTFSG4
Grammar MTFSG4 (Figure 10.5)

This added definite articles to noun phrases. The exceptions fell into two main classes. Firstly there were the contractions of preposition and article that therefore obviated the need for an article. Secondly, there were nominals that already had a definite article as a part, and which therefore did not need another.
MTFSG5

(S1 IF PHRASE=NP [GEN=fem/NUM=s1] THEN GOTO S2
   ELSE GOTO S5)
(S2 IF TREE=∅catl[GEN=inv/][∅]! THEN GOTO S6
   OR TREE=ad 1{∅}! THEN GOTO S3
   OR TREE=de 1{∅}! THEN GOTO S4
   ELSE GOTO S6)
(S3 IF TREE=∅catl{∅e}! THEN GOTO S6
   OR TREE=∅catl{∅gu}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅c}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅f}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅e1}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅e1}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅en}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅on}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅er}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅eur}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅et}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅au}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅ou}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   OR TREE=∅catl{∅x}! THEN SUB TREE=∅catl{∅%∅e}!* S6
   ELSE GOTO S6)
(S4 IF TREE=de 1{le}! THEN SUB TREE=de 0{le}!* S6
   ELSE GOTO S6)
(S5 IF TREE=ve 1{∅e}! THEN GOTO S6
   OR TREE=ve 1{∅}! AND PHRASE=VP [GEN=fem]
   THEN SUB TREE=ve 1{∅%∅e}!* S6
   ELSE GOTO S6)
(S6 IF UNIT<>*END THEN SET UNIT=+1 S1
   OR MORE PHRASES THEN SET PHRASE=+1 AND SET UNIT=+1 S1
   ELSE SET PHRASE=1 AND SET UNIT=1 AND LOAD GRAMMAR=MTFSC6,2771))

Figure 10.6
Grammar MTFSG5
Grammar MTFSC5 (Figure 10.6)

This created feminine forms of adjectives, the definite article, and "verbs" (ie past participles).
MTFSG6

((S1 IF PHRASE=NP [NUM=p1] THEN GOTO S2
ELSE GOTO S5)
(S2 IF TREE=\textit{\#cat1\{NUM=in\}\{\#\}} THEN GOTO S6
OR TREE=ad 1\{\#\} THEN GOTO S3
OR TREE=de 1\{\#\} THEN GOTO S4
ELSE GOTO S6)
(S3 IF TREE=\textit{\#cat1\{\#s\}} THEN GOTO S6
OR TREE=\textit{\#cat1\{\#x\}} THEN GOTO S6
OR TREE=\textit{\#cat1\{\#z\}} THEN GOTO S6
OR TREE=\textit{\#cat1\{\#au\}} THEN SUB TREE=\textit{\#cat1\{\#u\#ux\}}!* S6
OR TREE=\textit{\#cat1\{\#eu\}} THEN SUB TREE=\textit{\#cat1\{\#u\#ux\}}!* S6
OR TREE=\textit{\#cat1\{\#al\}} THEN SUB TREE=\textit{\#cat1\{\#1\#ux\}}!* S6
OR TREE=\textit{\#cat1\{\#\}} THEN SUB TREE=\textit{\#cat1\{\#\#s\}}!* S6
ELSE GOTO S6)
(S4 IF TREE=de 1\{le\} THEN SUB TREE=de 0\{les\}!* S6
ELSE GOTO S6)
(S5 IF TREE=ve 1\{\#s\} THEN GOTO S6
OR TREE=ve 1\{\#\} AND PHRASE=VP [NUM=p1] THEN SUB TREE=ve 1\{\#\#s\}!* S6
ELSE GOTO S6)
(S6 IF UNIT<>END THEN SET UNIT=1 S1
OR MORE PHRASES THEN SET PHRASE=+1 AND SET UNIT=1 S1
ELSE SET PHRASE=1 AND SET UNIT=1 AND LOAD GRAMMAR=MTFSC7,2771))

Figure 10.7
Grammar MTFSG6
Grammar MTFSG6 (Figure 10.7)

This created plural forms of adjectives, the definite article, and past participles. It is important that this grammar succeed MTFSG5, as the reverse would result in incorrect forms of the feminine plural past participle.
MTFSG7

(S1 IF TREE=de 1{le}+$\text{cat2}[\text{WVL}=V/]{\theta}$! THEN SUB TREE=de 0{1'}+$\text{cat2}{\theta}$!* S2
OR TREE=de 1{1a}+$\text{cat2}[\text{WVL}=V/]{\theta}$! THEN SUB TREE=de 0{1'}+$\text{cat2}{\theta}$!* S2
OR TREE=pr 1{#e}+NP 2+{#cat3[VWL=V/]{\theta}$!
THEN SUB TREE=pr 1{#eX'\theta'}+NP 2+{#cat3{\theta}$!* S2
ELSE GOTO S2)

(S2 IF UNIT<>*END THEN SET UNIT=+1 S1
OR MORE PHRASES THEN SET PHRASE=+1 AND SET UNIT=1 S1
ELSE SET PHRASE=1 AND SET UNIT=1 S3

(S3 IF TREE=pr 1{de}+NP 2+{de 3{le}! THEN SUB TREE=pr 0{du}+NP 2+{1* S4
OR TREE=pr 1{de}+NP 2+{de 3{le}! THEN SUB TREE=pr 0{des}+NP 2+{1* S4
OR TREE=pr 1{a\}+NP 2+{de 3{le}! THEN SUB TREE=pr 0{aux}+NP 2+{1* S4
ELSE GOTO S4)

(S4 IF UNIT<>*END THEN SET UNIT=+1 S3
OR MORE PHRASES THEN SET PHRASE=+1 AND SET UNIT=1 S3
ELSE WRITE TREES AND LOAD GRAMMAR=MTFSC2,2771))

Figure 10.8
Grammar MTFSG7
Grammar MTFSG7 (Figure 10.8)

The first part of this was concerned with elision. Before a lexical unit with a vowel start, "le" and "la" were changed to "l"; and prepositions ending with "e" that_preceded a unit with a vowel start lost that letter. The second part contracted lexical units. To summarize its actions:

- de + le becomes du
- de + les becomes des
- à + le becomes au
- à + les becomes aux

A comment on the grammars

The rules given here are by no means exhaustive, even for a limited language like PRECIS. They seemed sufficient for the sample used, but the important thing is that they could be changed and added to in order to surmount shortcomings.

This concludes the account of the truly linguistic processes of this project. It remains to describe the program that created the target language PRECIS strings.

Creating the target language strings (Program MT45C1)

This is listed in Appendix P. The program was not fully worked out, as indicated in the introductory remarks to this Chapter. The first process was to read the source language manipulation codes, on which the target language codes were to be based (lines 200-250), and the first theme was read into a string matrix (lines 300-435). The manipulation codes for the current theme were examined, and each focus, and "$d", "$o" and "$n" difference were assigned two pointers. The first was to the downward reading associated with the code, and the second to the upward. These pointers linked the codes to another string matrix in which the individual terms of the string were held (lines 2000-2065). The code was searched for substitute phrases and these were marked in the matrix, as were blank inserts (lines 4000-4380).
Each reading in the string was processed in turn, and the text added to the second matrix, according to the pointers. Essentially, the program sought out nominal and "verbal" syntactic labels, and examined the pointer to the manipulation code held in their data area. The data was then split at this point and added to the appropriate storage area. After all data had been allocated to their positions, the non-terminal syntactic labels were removed and any "surplus" lexical units moved around the matrix (lines 6000–6680).

It remained for the data to be concatenated into a string. Essentially this was done in two stages. The first was the comparison of the downward and upward versions of the data. The purpose of this was to identify the candidates for "$v" and "$w" terms. So if the two entries were:

- Downward: no {utilisation}pr {de}de {la}
- Upward: no {utilisation}pr {en}

the comparison would show that they should be concatenated as "utilisation $v de la $w en" (lines 7000–7790). The second stage was to concatenate the terms. The lexical units left after the comparison were tested for the presence in their lexical records of output forms different from that in the matrix. This really only applied to names, where the system dealt with lexical units in lower-case, but which required correct capitalization for output (lines 9450–9695). The remaining syntactic labels were then removed, with a single space character being inserted between lexical units where necessary (lines 9300–9397). Finally, the manipulation code was added to the text, and the whole concatenated with any of the target language string previously processed.

In this way, the whole of the string for this tree was built up. If there were more themes, these were processed in the same way, and concatenated with the string (with allowance being made for the duplication of common elements introduced by "$z" rather than "$x" and "$y". On completion of all themes, the string was
written to a backing store file, and the text preparation program called, so that the next string could be processed.

This last program could be somewhat expanded on. A number of facilities could have been added, amongst which were the checking of lexical units to determine whether or not they needed to be made leads. Substitutes were poorly treated in that (apart from drawing their manipulation code directly from the source language) there was no check on the need for a substitute in the target language string. A routine for processing "operator g blocks" was not included because of time considerations. Finally, the program and therefore the performance of the system as a whole, suffered because the lead only terms had been omitted and because there was no way in which they could be replaced from a thesaurus, simply because the system did not have a thesaurus to call upon.

The final results are presented in Appendix F. The successful strings (of which there were three hundred and ninety-nine) are given both as a list of individual strings and as a manipulated index. A list of strings that were either rejected by the system or wrongly translated is also given, and a summary of the causes of rejection in Table 10.1. Strings that produced results that could reasonably be processed by the manipulation programs were included, although they might not have been "good PRECIS". The final module did not allow for the insertion of non-lead differences, so the term "$z6103 geophysics $01 for" was translated as "$z6103 pour la geophysique", and thus there was an entry in the resulting index under the preposition but not the noun. Another failing that was allowed through was where the prepositions within an operator "6" term had been translated by a tree preposition and was thereby not rendered as idiomatically as was desirable. This is particularly evident under the heading "Terre", where "from" has been translated as "depuis" in string 321 (which has an operator "6" and therefore no semantic analysis) and as "à partir de" in string 320 (which has an
operator "3" and therefore a semantic analysis). The most serious error was the failure to translate the downward and upward readings of a preposition within a term consistently, which in turn indicated a poor performance of the semantic analyser.
Cause of rejection          Strings

No syntactic analysis       78,225,277

"g blocks" wrongly handled  91,228,261,262,285
                               288,303,325,336,366

Prepositions within a term  138,327
    differently translated in the
    downward and upward readings

Lead-only term dropped to make the
    resulting string defective  45,46,248

Lead-only term and effects of a "not
    downward" term misinterpreted  312,337

Substitute's regime misinterpreted  79

Core working area exceeded   148,213,395

Table 10.1

Analysis of the causes of failures
Reference

CHAPTER XI
THE OBJECTIVES REVISTED AND CONCLUSIONS

In Chapter VI, the objectives were summarized in nine parts. The first was the system should be implemented on a minicomputer. The implementation turned out not to be practical, mainly because the core size of about 27k was far too small for a fast and efficient system, if it were to produce adequate translations.

The system seems to be extendible to language pairs other than English to French, although some modifications to some routines may be necessary. For the use of the current French analysis/synthesis dictionary for the analysis of French, there would have to be a morphological analyser inserted between the present text preparation module and the syntactic analyser, so that adjectives, past participles and those prepositions subject to contraction and elision could be converted into forms used as the dictionary keys. As for the extension to other string indexing languages, no experiments were carried out, but there seems no reason for different first and last programs being used to process languages other than PRECIS, with the condition that the indexing system would actually "work" in the target language.

The choice of a batch system was not completely ideal. While it fitted well with the way in which the British Library's computer systems were operated in 1979, it should have been realized that on-line operating would increasingly become available to information systems. Indeed, at the time this choice was made, TITUS IV was being developed in its on-line form. Systems used in a multilingual context would need some verification of the translation, and this may best be done when the original string is entered. Furthermore, the verificational checks made on the source language string in syntactic and semantic analysis would be more efficiently used in an on-line system.
The building of the dictionaries was relatively inexpensive in time, taking an average of about twenty minutes for the linguistic research and entry of an item into all the three dictionaries. The ease of maintenance was never really tested, simply because the system was never entrusted to the care of anyone other than the author. Some checks were made to avoid duplication of entries, but on the other hand, there was no check made, for instance, that an entry in the transfer dictionary had entries in the source and target language monolingual dictionaries.

The cost of adapting the system to a new language pair was not tested, although the relative simplicity of the grammars used (in comparison with those necessary for natural language) suggests that the cost would be low.

Finally, it has been shown that the target language strings were not of as high a quality as those in the source language.

Future directions

It has been a contention of this thesis that the British Library PRECIS Translingual Project was by its nature always going to be less than satisfactory. Could it fairly be said that this transfer approach, using more sophisticated techniques, has also been less than satisfactory?

From a cost-effective viewpoint, it could be argued that only ten per cent of PRECIS strings contain a preposition, and that there is a "normal" translation for each preposition between specific language pairs. The use of the lexicons with extra bilingual entries for the translation of prepositions would, on the face of it, give fairly correct results. It would have the advantage of using the thesauri that would have to be present for any good implementation of PRECIS, but would save the expense of the semantic analysis module and the creation of the analysis/synthesis and the transfer dictionaries.
This "cost-effective" solution is misleading, for two reasons in particular. The first is that although it seems very simple, it would not really produce reasonable results between English and French at least. Extra provision would have to be made for the provision and agreement of articles, as well as for elision and contraction. This in turn implies the presence of some information about the syntactic structure of the string somehow attached to the terms. This would have to be done by some parsing mechanism, however crude. It would probably be less expensive to employ a post-editor to work over the bare strings translated by multilingual thesauri, but this in turn introduces problems of tedium and indeed the continuity in the supply of post-editors of suitable quality.

Strings of the required quality could be produced by a human translator, or by a post-editor with a partial translation system as was suggested above. It is my contention that results of a similar quality are possible from an immediate descendant of the system presented in this thesis. Leaving aside the problems caused by the abbreviation of the final program, the remaining problems would largely disappear if multilingual thesauri were used to translate terms. Not only would differences between downward and upward readings of the same term go, but also the problems of translating lead-only and operator "6" terms, as these would necessarily be included in the thesauri.

Of the problems encountered, only the lack of core storage and thereby the slowness of the system would remain, until such time as it is mounted on a larger machine. New problems would certainly arise, but these would be problems of detail and not of great concern to the main substance of the system.

A system based on that here presented is a feasible method of translating PRECIS to a high standard. It has been an interesting project, which seems at times to have taken a direction of its own, and as a consequence, has ended with the
presentation of a system nearly completely different to the one envisaged at the outset. "Better is the end of a thing than the beginning thereof."
BIBLIOGRAPHY


British Library (Bibliographic Services Division). (1981a) **PRECIS categories of forms.** 3rd ed. British Library (Bibliographic Services Division).

British Library (Bibliographic Services Division). (1981b) **PRECIS DC fiche.** British Library (Bibliographic Services Division). [Negative microfiche].


In
Reprinted in


Ducrot, J.M. (1973b) TITUS system: why, how and toward what aim? In

Ducrot, J.M. (forthcoming) TITUS IV: système de traduction automatique et simultanée en quatre langues.


In


In


In


In Coling 80: proceedings of the 8th International Conference on Computational Linguistics, Tokyo, 30 September-4 October 1980. [s.l]: [s.n]. p424-429.

In Coling 80: proceedings of the 8th International Conference on Computational Linguistics, Tokyo, 30 September-4 October 1980. [s.l]: [s.n]. p350-353.


In

In

In


In

Sørensen, J. and Austin, D. (1976) PRECIS in a multilingual context: part 3, multilingual experiments, proposed codes, and


Toma, P.P. (1977) SYSTRAN as a multilingual machine translation system.


