ON THE DESIGN OF EXPERT SYSTEMS GRAFTED ON MT SYSTEMS

presented for the seminar on MT
(August 14-16, 1985),
Hamilton, Colgate University

by
R.GERBER & Ch.BOITET

GETA, BP 68
Université de Grenoble
38402 Saint-Martin-d'Hères, FRANCE

ABSTRACT

Our MT systems integrate many advanced concepts from the fields of computer science, linguistics, and AI: specialized languages for linguistic programming based on production systems, complete linguistic programming environment, multilevel representations, organization of the lexicons around "lexical units", units of translation of the size of several paragraphs, possibility to use text-driven heuristic strategies.

We are now beginning to integrate new techniques: unified design of an "integrated" lexical data-base containing the lexicon in "natural" and "coded" form, use of the "static grammars" formalism as a specification language, and design of a kind of structural metaeditor (driven by some static grammar) allowing the interactive construction of a document in the same way as syntactic editors are used for developing programs.

This paper centers on our study on possible additions of expert systems equipped with metalinguistic and extralinguistic knowledge, in order to solve some problems encountered in second-generation MT systems. Several examples of the possible use of expert-corrector systems in M(a)T (Machine (aided) Translation) systems are given.

Keywords: Machine (aided) Translation, Expert Systems, Knowledge, Specification and Verification Tools.

INTRODUCTION

In this paper, we assume some basic knowledge of M(a)T (Machine (aided) Translation) terminology (MT, MAHT, HAMT, etc.). The starting point of our research towards "better" M(a)T systems is briefly reviewed in I. In II, we present 3 lines of related work: unifying knowledge processing by using new notions from data-base management systems and verification tools, improving current second-generation methodology by incorporating advanced techniques from software engineering (specification languages), and returning to interactive techniques for the creation of a document. In part III, we describe in more detail a research aiming at transforming second-generation M(a)T systems into third-generation systems, by grafting on them expert corrector systems.

I - BACKGROUND

We first want to present some important existing concepts, from which we start to improve current M(a)T systems.

1 - COMPUTER SCIENCE ASPECTS

In second-generation systems, and in projected third-generation systems, emphasis is placed on the use of Specialized Languages for Linguistic Programming (SLLP), which offer built-in data and control structures, with an underlying powerful mechanism.

SLLPs are designed to be easy to use by linguists and terminologists who have almost no computer science background. Hence, they must be integrated in some "user-friendly" environment. At GETA, this environment, called ARIANE-78, is implemented (under VMSP/CMS) as a specialized data-base of what we call "lingware" ("linguiciel") files (grammars, dictionaries, procedures, formats, variables) and of corpuses of texts (source, translated, revised, plus intermediate results and possibly "hors-textes" -- figures, etc.). A conversational monitor interfaces the data base with the users (in French or in English).

The process of rough translation is realized by a monolingual analysis, followed by a bilingual transfer, and then by a monolingual generation (synthesis). In ARIANE-78.5, these fundamental phases have been broken down into several elementary steps, each of them programmed in one of the SLLPs supported by the system. For more details, see [4].

2 - LINGUISTIC ASPECTS

Translation requires a good enough understanding. In actual fact, understanding cannot be defined in an absolute way, but only with reference to some domain. The hierarchy of understanding is organized around a hierarchy of levels of interpretation. We distinguish between linguistic levels and extralinguistic levels.

The linguistic levels comprise the levels of morphology, syntax-1 (syntactic and syntagmatic classes), syntax-2 (syntactic functions and dependency relations), logico-semantics (logical relations, or "inner cases", and semantic relations, or "outer cases"). There are of course other, not "structural" types of information (actualization features, type of statement, semantic features, etc.). Understanding a text in these terms is equivalent to find a formula representing it in a formal system of a linguistic nature. Following P. Pognan, we speak of implicit understanding, characteristic of second-generation MT systems.

The extralinguistic levels are those of expertise (some static knowledge about a particular subject matter, consisting in a collection of facts, rules and procedures), and pragmatics (representation of the facts, events, suppositions, scenarios, etc., described by the text). This presupposes the ability to learn facts and structures, to reason by analogy, and to abstract. In short, pragmatics is related to the most ambitious themes of AI. Until now, only very small illustrative computer models have been presented.

Understanding at some extralinguistic level may be called explicit understanding. Typical applications where it is needed include command of robots and intelligent interfaces. However, for translation purposes, it is not necessary to achieve this level of understanding for all encountered sentences. Implicit understanding is often sufficient. At the level of a technical revisor, complete explicit understanding is required.

Second generation M(a)T systems rely only on "implicit" understanding. In the past, and in some current systems still, the previously mentioned levels are mutually exclusive. By this, we mean that a given text will have separate representations for each of the defined levels. This usually leads to a sequential strategy of processing, with all its drawbacks. This is why GETA uses B.Vauquois' concept of multilevel interface structures to represent all the computed levels on the same graph (a "decorated tree"). In short, such structures are in effect generators of representations at different levels, and also factorize various types of ambiguities.

The organization of the lexical knowledge centers on the notion of lexical unit (LU). A given lexical unit usually groups several lemmas (normal form of words), which are considered to derive from the LU by some derivation schema, which has an interpretation at the levels of morphology, syntax and semantics, so that, at generation time, it is possible to paraphrase by choosing the appropriate lemma, for a given LU.

In the current applications, the units of translation are of the size of one or more paragraphs. This allows for intersentential resolution of anaphoras in some not too difficult cases.

3 - AI ASPECTS

After the morphological analysis, the unit of translation is represented by the current "object tree", which may encode several competing interpretations, as the "blackboard" of other systems. This allows for some heuristic programming, because it is possible to explicitly describe and process ambiguous situations in the production rules.

This is in contrast to systems based on combinatorial algorithms which construct each interpretation independently, even if they represent them in a factorized way.

Looking at the experience just reviewed, we felt that the time was ripe to make one more step towards better M(a)T systems, by using other existing AI techniques.

II - PARALLEL LINES OF AI RELATED WORK

The experience gained by the development of a Russian-French translation unit of a realistic size over the last three years [6] has shown that maintaining and upgrading the lingware, even in an admittedly limited second generation M(a)T system, requires quite a lot of expertise. Techniques are now being developed to maintain the linguistic knowledge base. Some of them deal with the lexical data-base, others with the definition and use of specification formalisms ("static grammars") and verification tools.

1 - LEXICAL KNOWLEDGE PROCESSING

In the long run, dictionaries turn out to be the costliest components of M(a)T systems. Hence, we are working towards the reconciliation of "natural" and "coded" dictionaries, and towards the construction of automated verification and indexing tools. Natural dictionaries are usually accessed by the lemmas (normal forms). Coded dictionaries of M(a)T systems, on the other hand, are accessed by morphs or by lexical units. Moreover, the information the two types of dictionaries contain is not the same.

However, it is highly desirable to maintain some degree of coherency between the coded dictionaries of a M(a)T system and the natural dictionaries which constitute their source, for documentation purposes, and also because these computerized natural dictionaries should be made accessible to the revisors.

Let us briefly present the kind of structure proposed by N. Nedobejkine and Ch. Boitet at an ATALA meeting in Paris in 1983. The central idea here is to start from the structure of modern dictionaries, which give access by the lemmas, but use the notion of lexical unit. Each item may be considered as a tree structure. Starting from the top, selections of a "local" nature (on the syntactico-semantic behavior in a phrase or in a sentence) give access to the "constructions". Then, more "global" constraints lead to "word senses".

At each node, codes of one or more formalized models may be grafted on. Hence, it is in principle possible to index directly in this structure, and then to design programs to construct the coded dictionaries in the formats expected by the various SLLPs. Up to this level, the information is monolingual and usable for analysis as well as for generation. If this language is source in one or more language pairs, each word sense may be further refined, for each target language, and lead to equivalents expressed as constructions of the target language, with all other information contained in the dictionary constructed in a similar way for the target language.

Hence, we have not tried to define a unique "many-to-many" dictionary, but rather a set of "one-to-many" dictionaries, in particular because we don't know of any dictionary of the first type developed for the basic core of several natural languages, which would include relatively unambiguous technical terms (usually noun phrases) in a given domain as well as highly ambiguous and difficult verbs.

This part of the work, hence, aims at finding a good way of representing the lexical knowledge.

But there is another problem, perhaps even more important. Because of the cost of building machine dictionaries, we need some way to transform and transport lexical knowledge from one M(a)T system to another. This is obviously a problem of translation. Hence, we consider this type of "integrated structure" as a possible lexical interface structure. Research has recently begun on the possibility of using classical or advanced data base systems to store this lexical knowledge and to implement the various tools required for addition and verification. VISULEX and ATLAS [1] are first versions of such tools.

2 - A SPECIFICATION FORMALISM FOR LINGUISTIC APPLICATIONS

Just as in current software engineering, we have long felt the need for some level of "static" (algebraic) specification of the functions to be realized by algoritms expressed in procedural programming languages. In the case of M(a)T systems, there is no a priori correct grammar of the language, and natural language is inherently ambiguous. Hence, any usable specification must specify a relation (not a function) between strings and trees, or trees and trees: many trees may correspond to one string, and, conversely, many strings may correspond to one tree.

Working with B.Vauquois in this direction, S.Chappuy has presented a formalism of static grammars, [7], presented in charts expressing the relation between strings of terminal elements (usually decorations expressing the result of some morphological analysis) and multilevel structural descriptors. This formalism is currently being used for all new linguistic developments at GETA. Of course, this is not a completely new idea. For example, M.Kay [14] proposed the formalism of unification grammars for quite the same purpose. But his formalism is more algebraic and less geometric in nature, and we prefer to use a specification in terms of the kind of structures we are accustomed to manipulating.

3 - AIDING THE CREATION OF THE SOURCE DOCUMENTS

Lingware engineering may be compared with modern software engineering, because it requires the design and implementation of complete programming systems, uses specification tools, and leads to research in automatic program generation. Starting from this analogy, a group of researchers at GETA have recently embarked on a project which could converge with still another line of software engineering, in a very interesting way. As a matter of fact, they are trying to design and implement a syntactico-semantic structural metaeditor, that uses a static grammar given as parameter, in order to guide an author who is writing a document, in much the same manner as metaeditors like MENTOR are used for writing programs in classical programming languages.

This could offer an attractive alternative to interactive M(a)T systems like ITS, which require a specialist to assist the system during the translation process. As a matter of fact, this principle is a sophisticated variant of the "controlled syntax" idea, like that implemented in the TITUS system. Its essential advantage is to guarantee the correctness of the intermediate structure, without the need for a large domain-specific knowledge base. It may be added that, in many cases, the documents being written are in effect contributing some new knowledge to the domain of discourse, and hence cannot be already present in the computerized knowledge base, even where one exists.

III - GRAFTING EXPERT SYSTEMS

Seing that linguistic expertise is already quite well represented and handled in current ("closed") systems, we have oriented our research towards the possibility of adding extralinguistic knowledge to existing M(a)T systems.

We believe that this level of knowledge, and a mechanism to handle it properly, is the key to the solution of some problems encountered in current M(a)T systems (analysis of some kinds of noun compounds, coordination, anaphora, selection of equivalents,...).

Extralinguistic knowledge comprises specialized knowledge of some technical or scientific field, some notions of common sense and pragmatic knowledge (aquired dynamically during text processing).

In his thesis [9], the first author attempted to design such a system, and to propose an initial implementation.

Two types of expert systems have been studied. The first is called expert-interactor system. It cooperates with the MT system during phases like analysis or transfer: it is called by grammar rules to perform tests according to the knowledge base. This method is in fact very similar to man/machine interaction and needs high expertise and complicated modelling of the different processes.

The second type is called expert-corrector system and operates between phases.

A complete translation system may then be constructed by combining a "closed" system, based only on linguistic knowledge (an application written in ARIANE-78), and one or two "open" systems from the second type. The first is inserted between analysis and transfer, and the second between transfer and generation. This is illustrated by the following diagram.

This architecture is very advantageous for several reasons. First, it leaves the linguistic applications (lingwares) unchanged. Large-scale lingwares may be reused as they are.

Second, it is modular: both the lingware and the expert-corrector system can be independently modified.

Third, there is some added modularity, because it is possible to develop a core grammar used by the MT system and to incorporate the domain dependent treatment into the expert system. This is contrast to some previous systems, where some domain-specific knowledge has been incorporated into the lingware, making it extremely difficult to adapt the system to another domain.

Fourth, in systems based on transducers rather than on analyzers, it is perfectly possible that the results of analysis or of transfer (the "structural descriptors") are partially incorrect and need correction. Hence, we need some knowledge about the types of errors made by linguistic systems. Such knowledge may be called metalinguistic (see figure above).

```
+-----------------------------------------------------------------+
!      --------------> metalinguistic <---------------             !
!     !                  knowledge                     !           !
!     !                                                !           !
!     !          ---> extralinguistic <----            !           !
!     !         !         knowledge        !           !           !
!     !         !                          !           !           !
!     !         !                          !           !           !
!     !         !                          !           !           !
!     !         !                          !           !           !
!     !         !                          !           !           !
! +----------------+   +----------+   +----------------+           !
! !Expert corrector!   !          !   !Expert corrector!           !
! !system for      !--->!TRANSFER !--->!system for     !           !
! !analysis        !   !          !   !transfer        !           !
! +----------------+   +----------+   +----------------+           !
!         ^                 ^                ^                     !
!         !                 !                !                     !
!         !                 !                !                     !
!         !                 !                !                     !
!         !                 !                !                     !
!         !                 !                !                     !
! +----------------+                  +----------------+           !
! !                !<-- linguistic --->!               !  !        !
! !   ANALYSIS     !    knowledge     !   GENERATION   !  !        !
! !                !                  !                !  !        !
! +----------------+                  +----------------+  !        !
+-----------------------------------------------------------------+
```

The control structure of a corrector system is as follows:

```
[1]   TRANSFORM the result of analysis into a suitable
        form;

[2]   while there is some error configuration do
[3]       SOLVE (by using the meta- or extralinguistic
             knowledge);
             if solving has failed then EXIT endif;

[4]       PERFORM a partial reconstruction of the
             structure, according to the solution found;
      endwhile;

[5]   OUTPUT the final structure in ARIANE-78 format.
```

In [1], the "suitable form" defined is a set of PROLOG clauses representing the decorated tree. This representation allows for easy search and tests by unification.

Step [2] uses only metalinguistic knowledge encoded in "error profiles". This kind of knowledge is of the type used by a linguist looking at an intermediate result (of analysis or transfer). He or she knows the kind of errors possibly made by his application. An error profile is a kind of subtree pattern. Step [2] tries to match the profile on the intermediate "object tree".

The result of phase [2] is a list of configurations, which locate in fact potential errors. The corresponding parts of the structure are then tested by SOLVE, using selectional restrictions, preferential semantics and expert system reasoning techniques.

At this point, SOLVE tries to compute "explicit" understanding, starting from the "implicit" understanding incorporated in the linguistic descriptor, and using the extralinguistic knowledge base.

This knowledge base is a set of logic formulas represented by PROLOG clauses, defining lexical entries (static knowledge) and expert rules from the domain.

This process of going one level further during understanding can be compared to the operation performed by a human technical translator referring to a schema or to an expert ("personne ressource") during translation. Many such experiences have been reported by human professional translators from the Canadian Translation Bureau. For more detail, see [10].

"EXIT" does not mean that no translation result is given. If SOLVE fails on one problem, it leaves the structure as it is and goes to the next error configuration.

The first author has implemented a prototype in Foll-PROLOG [8]. The lingware used corresponds to a small English-French system developed for teaching purposes (BEX-FEX).

We have chosen the following few examples to illustrate the main ideas of the method.

EXAMPLE 1: ADJ + N N

(1) Standard free-energy change is calculated by this equation.

The analyzer proposes that "standard" modifies "change", while "free-energy" is juxtaposed to "change", as shown in (tree 1), hence the erroneous translation:

"La variation standard d'énergie libre est calculée par cette formule."

In order to correct the structure to obtain (subtree 1), some knowledge of chemistry is required, namely that "standard free-energy change" is a... standard notion.

EXAMPLE 2: (ADJ) N AND N N

(2) The mixture gives off dangerous cyanide and chlorine fumes.

(2') The experiment requires carbon and nitrogen tetraoxyde.

Let us develop this example a little more. Sentence (2) presents the problem of determining the scope of the coordination. The result of analysis (tree nù2) groups "dangerous cyanide" and "chlorine fumes", "chlorine" being juxtaposed to "fumes" (SF(JUXT) on node 12). Hence the translation:

"La préparation dégage le cyanure et la vapeur de chlore dangereux."

```
+----------------------------------------------------------+
!                  TEXTE RENEG PHRASE1                      !
!             Analyse structurale corrigée                  !
!                                                           !
!                                *NP                        !
! +------------------+          ......4                     !
! !   Subtree  ù1  !             .!                         !
! +------------------+            !                         !
!                                 !                         !
!                          -------------                    !
!                          !            !                   !
!                        *NP         CHANGE                 !
!                        ......4'    ......8                !
!                          !                                !
!                          !                                !
!                          !                                !
!                    ----------                             !
!                    !         !                            !
!                  *AP      FREE_EN                         !
!                  ......5  ERGIE.7                         !
!                    !                                      !
!                    !                                      !
!                    !                                      !
!                    !                                      !
!                    !                                      !
!                  STANDARD                                 !
!                  ......6                                  !
!                                                           !
+----------------------------------------------------------+
+----------------------------------------------------------+
!     RESULTAT DE L'EXECUTION, TEXTE : RENEG   PHRASE1     !
!                   ANALYSE STRUCTURALE                    !
!                                                          !
!                             ULTXT                        !
!                             ......1                      !
!                                !                         !
!                                !                         !
!   +------------+               !                         !
!   !  Tree   1 !            ULFRA                         !
!   +------------+            ......2                      !
!                                !                         !
!                                !                         !
!                                !                         !
!                             *VCL                         !
!                             ......3                      !
!                                !                         !
!                                !                         !
!                                !                         !
!            -----------------------------------------     !
!            !               !           !           !     !
!        *NP             *VK         *NP         .         !
!        ......4         ......9     ......11    .....14   !
!            !               !           !                 !
!            !               !           !                 !
!            !               !           !                 !
!   -----------------        !      ---------              !
!   !       !       !        !      !       !              !
! *AP     FREE-EN CHANGE  CALCULA THIS    EQUATIO          !
! ......5 ER....7 ......8 TE...10 .....12 N....13          !
!   !                                                      !
!   !                                                      !
!   !                                                      !
!   !                                                      !
!   !                                                      !
! STANDAR                                                  !
! D.....6                                                  !
!                                                          !
!  SOMMET 4  ' ': UL('*NP'),RL(ARG1),K(NP),SF(SUBJ),       !
!     CAT(N),SUBN(CN),NUM(PLU),SEM(CONC),VL1(N).           !
!  SOMMET 5  ' ': UL('*AP'),RS(QUAL),K(AP),SF(ATG),        !
!     TYPOG(CAP),CAT(A),SUBA(ADJ).                         !
!  SOMMET 6  '*STANDARD': UL('STANDARD'),SF(GOV),TYPOG(CAP)!
!     CAT(A),SUBA(ADJ).                                    !
!  SOMMET 7  'FREE-ENERGY': UL('FREE-ENERGY'),RS(QUAL),    !
!     UNSAFE(RS),SF(JUXT),CAT(N),SUBN(CN),NUM(SIN),        !
!     SEM(ABST).                                           !
!  SOMMET 8  'CHANGE': UL('CHANGE'),SF(GOV),               !
!     CAT(N),SUBN(CN),NUM(PLU),SEM(PROC).                  !
+----------------------------------------------------------+
+-----------------------------------------------------------+
!   RESULTAT DE L'EXECUTION, TEXTE : RENEG   PHRASE2        !
!   ANALYSE STRUCTURALE                                     !
!                                                           !
!                          ULTXT                            !
!                          ......1                          !
!   +------------+            !                             !
!   ! Tree nû 2  !         ULFRA                            !
!   +------------+         ......2                          !
!                             !                             !
!                          *VCL                             !
!                          ......3                          !
!                             !                             !
!       -----------------------------------------------     !
!       !           !                 !                !    !
!    *NP         *VK               *NP                 .    !
!    ......4     ......7           ......9         .....17!
!       !           !                 !                     !
!   ---------       !       -------------------             !
!   !       !       !       !       !         !             !
!THE     MIXTURE GIVE    *AP     CYANIDE    *NP             !
!......5 ......6 ......8 .....10 .....12    .....13         !
!                            !                !             !
!                            !        -----------------     !
!                            !        !       !       !     !
!                        DANGERO   AND     CHLORIN FUMES    !
!                        U....11   .....14 E....15 .....16!
!                                                           !
!  SOMMET 9  ' ': UL('*NP'),RL(ARG1),K(NP),SF(OBJ1),CAT(N),!
!     SUBN(CN),NUM(SIN),SEM(CONC),SEMCO(SUBST),VL1(N).      !
!  SOMMET 10  ' ': UL('*AP'),RS(QUAL),K(AP),SF(ATG),CAT(A),!
!     SUBA(ADJ),IMPERS(IMPED),SUBJR(INF).                   !
!  SOMMET 11  'DANGEROUS': UL('DANGEROUS'),SF(GOV),CAT(A),  !
!     SUBA(ADJ),SUBJR(INF).                                 !
!  SOMMET 12  'CYANIDE': UL('CYANIDE'),SF(GOV),CAT(N),      !
!     SUBN(CN),NUM(SIN),SEM(CONC),SEMCO(SUBST).             !
!  SOMMET 13  ' ': UL('*NP'),RL(ID),K(NP),SF(COORD),CAT(N),!
!     SUBN(CN),NUM(PLU),SEM(CONC),SEMCO(SUBST),VL1(N).      !
!  SOMMET 14  'AND': UL('AND'),CAT(C).                      !
!  SOMMET 15  'CHLORINE': UL('CHLORINE'),RS(QUAL),          !
!     UNSAFE(RS),SF(JUXT),CAT(N),SUBN(CN),NUM(SIN),         !
!     SEM(CONC),SEMCO(SUBST).                               !
!  SOMMET 16  'FUMES': UL('FUMES'),SF(GOV),CAT(N),SUBN(CN),!
!     NUM(PLU),SEM(CONC),SEMCO(SUBST).                      !
+-----------------------------------------------------------+
+----------------------------------------------------------------------+
!                                                                      !
!                       TEXTE RENEG PHRASE2                            !
!                  Analyse structurale corrigée                        !
!                                                                      !
! +--------------+            *NP                                      !
! ! Sub-tree nù2 !            ......9                                  !
! +--------------+               !                                     !
!                                !                                     !
!           ------------------------------------                       !
!           !                    !                 !                   !
!         *AP                  *NP               FUMES                 !
!         .....10              .....9'           .....16               !
!           !                    !                                     !
!           !             ------------                                 !
!           !             !           !                                !
!         DANGERO      CYANIDE      *NP                                !
!         U....11      .....12      .....13                            !
!                                     !                                !
!                                 ---------                            !
!                                 !       !                            !
!                               AND     CHLORINE                       !
!                               .....14 ......15                       !
+----------------------------------------------------------------------+
```

But, if we know that cyanide is dangerous as fumes, and not as crystals, we can correct the structure by grouping "(cyanide and chlorine) fumes" (see subtree nù2). The produced translation will then be:

"La préparation dégage la vapeur dangereuse de cyanure et de chlore."

Of course, some more sophisticated analyzers would (and some actually do) put a semantic marker like "chemical element" on both "chlorine" and "cyanide", and then group "cyanide and chlorine" on the basis of the "semantic density" (e.g., number of features shared). But this technique will fail on (2'), because there is no "carbon tetraoxyde" in normal chemistry! Hence, without extralinguistic knowledge, the system will produce:

"L'expérience demande du tétraoxyde de carbone et d'azote."

instead of:

"L'expérience demande du carbone et du tétraoxyde d'azote."

This kind of knowledge on carbon tetraoxyde should be deduced by reasoning on more elementary knowledge (static knowledge on chemical elements). The corrected fragment of the structure is shown in subtree nù2.

```
+-----------------------------------------------------------+
!     RESULTAT DE L'EXECUTION, TEXTE : RENEG   PHRASE3      !
!                   ANALYSE STRUCTURALE                     !
!                                                           !
!                                 ULTXT                     !
!                                 ......1                   !
!                                    !                      !
!                                    !                      !
!       +------------+               !                      !
!       !  Tree nù 3 !               ULFRA                  !
!       +------------+               ......2                !
!                                    !                      !
!                                    !                      !
!                                    !                      !
!                                 *VCL                      !
!                                 ......3                   !
!                                    !                      !
!                                    !                      !
!            -------------------------------------------    !
!            !                       !       !         !    !
!         *NP                     *VK     *AP        .      !
!         ......4                 ....19 .....21   .....23!
!            !                       !       !              !
!            !                       !       !              !
!            !                       !       !              !
!   ------------------               !       !              !
! !       !       !                  !       !              !
! THE     WATER   /NP             BE      DANGERO           !
! ......5 ......6 ......7          .....20 U....22          !
!                 !                                         !
!                 !                                         !
!                 !                                         !
!   -------------------------------------                   !
!   !        !        !                  !                  !
! IN       THE      BEAKER            *RELCL                !
! ......8  ......9  .....10           .....11               !
!                                        !                  !
!                                        !                  !
!                                        !                  !
!                           ----------------------------    !
!                           !            !             !    !
!                        *NP         /NP          *VK       !
!                        .....12     .....14      .....17!
!                           !            !             !    !
!                           !            !             !    !
!                           !            !             !    !
!                           !        ---------         !    !
!                           !        !       !         !    !
!                        BEAKER  THE     CHLORIN  COMBINE!
!                        .....13 .....15 E....16  .....18!
+-----------------------------------------------------------+
+------------------------------------------------------------------+
!  SOMMET 4   ' ': UL('*NP'),RL(ARG0),K(NP),SF(SUBJ),CAT(N),  !
!     SUBN(CN),NUM(SIN),SEM(CONC),SEMCO(SUBST),VL1(N).          !
!  SOMMET 5   '*THE': UL('THE'),SF(DES),TYPOG(CAP),CAT(D),      !
!     SUBD(ART),NUM(SIN).                                        !
!  SOMMET 6   'WATER': UL('WATER'),SF(GOV),CAT(N),SUBN(CN),     !
!     NUM(SIN),SEM(CONC),SEMCO(SUBST).                           !
!  SOMMET 7   ' ': UL('/NP'),RS(LOCAL),UNSAFE(RS),K(NP),         !
!     SF(COMP),CAT(N),SUBN(CN),NUM(SIN),SEM(CONC),               !
!     SEMCO(OBJ),VL1(IN).                                        !
!  SOMMET 8   'IN': UL('IN'),RS(LOCAL),UNSAFE(RS),SF(REG),       !
!     CAT(S),SUBS(PREP),VL1(IN),JPCL(IN).                        !
!  SOMMET 9   'THE': UL('THE'),SF(DES),CAT(D),SUBD(ART),         !
!     NUM(SIN).                                                  !
!  SOMMET 10  'BEAKER': UL('BEAKER'),SF(GOV),CAT(N),             !
!     SUBN(CN),NUM(SIN),SEM(CONC),SEMCO(OBJ).                    !
!  SOMMET 11  ' ': UL('*RELCL'),RS(QUAL),UNSAFE(STR),            !
!     SLOCK(1),LOCKZ(1),LOCK2(1),K(RELCL), SF(ATG),CAT(V),       !
!     SUBV(VB),NUM(SIN),TENSE(PRES),SEM(PROC),SEMV(PROC),        !
!     JPCL(TOGETHER).                                            !
!  SOMMET 12  ' ': UL('*NP'),UNSAFE(RS),RL(ARG2),K(NP),          !
!     SF(OBJ2),CAT(R),SUBR(REL),NUM(SIN),VL1(WITH).              !
!  SOMMET 13  'WHICH': UL('BEAKER'),MCUL(SUBS),SF(GOV),          !
!     CAT(R),SUBR(REL),NUM(SIN).                                 !
!  SOMMET 17  ' ': UL('*VK'),RS(QUAL),SLOCK(1),LOCKZ(1),         !
!     LOCK2(1),K(VK),SF(GOV),CAT(V),SUBV(VB),NUM(SIN),           !
!     SYM(SYM12),TENSE(PRES),SEM(PROC),SEMV(PROC),VL1(N),        !
!     VL2(WITH),JPCL(TOGETHER).                                  !
!  SOMMET 18  'COMBINES': UL('COMBINE'),SLOCK(1),LOCKZ(1),       !
!     LOCK2(1),SF(GOV),CAT(V),SUBV(VB),NUM(SIN),SYM(SYM12),      !
!     TENSE(PRES),SEM(PROC),SEMV(PROC),VL2(WITH).                !
+------------------------------------------------------------------+
```

EXAMPLE 3: ANTECEDENT OF "WHICH"

(3) The water in the beaker with which the chlorine combines will be poisonous.

The analyzer takes "beaker" instead of "water" as antecedent of "which". The corrector may know that chlorine combines with water, and not with a beaker. This correction gives sub-tree nû3.

```
+-------------------------------------------------------------+
!                     TEXTE RENEG PHRASE3                      !
!                 Analyse structurale corrigée                 !
!                                                              !
!                             *NP                              !
!     +---------------+  .......4                              !
!     ! Sub-tree   nù3 !     !                                 !
!     +---------------+      !                                 !
!                            !                                 !
!   ----------------------------------------------------       !
!   !       !                !                      !          !
! THE     WATER             *NP                   RELCL        !
! ......5 ......6           ......7               .....11      !
!                            !                      !          !
!                            !                      !          !
!                            !                      !          !
!                   ------------------         ----------      !
!                   !        !        !         !    etc...!
!                  IN       THE     BEAKER     *NP             !
!                  ......8 ......9 .....10     .....12         !
!                                               !              !
!                                               !              !
!                                               !              !
!                                           ---------          !
!                                           !       !          !
!                                         WITH    WATER        !
!                                         .....13 .....14      !
+-------------------------------------------------------------+
```

EXAMPLES 4&5: ANTECEDENT OF "IT" IN OR OUT OF THE SENTENCE

(4) The state in which a substance is depends on the energy that it contains.

When a substance is heated the energy of the substance is increased. (5) The particles vibrate more vigorously, and it becomes a liquid. (5') It melts.

In order to choose between "substance" and "state" (4), one must make some type of complex reasoning using detailed knowledge of physics -- and one may easily fail in a given context: it is not correct to simply state that a substance may have an energy, while a state cannot (as we did to solve this particular case). Here, perhaps it is better to rely on some (metalinguistic) information on the typology. For (5), there are simple, but powerful rules like: if the antecedent cannot be found in the sentence, look for the nearest possible main clause subject to the left.

REFERENCES

(1) Bachut D. - Verastegui N.
Software tools for the environment of a computer aided translation system
COLING-84.

(2) Barr A. - Feigenbaum E., eds
The Handbook of Artificial Intelligence (vol 1, 2)
Pitman, 1981.

(3) Boitet Ch.
Research and development on MT and related techniques at Grenoble University (GETA).
Tutorial on MT, Lugano, April 1984, 17 p.

(4) Boitet Ch. - Guillaume P. - Quézel-Ambrunaz M.
Implementation and conversational environment of ARIANE 78.4, an integrated system for translation and human revision"
Proc. of COLING-82, Prag, July 1982, North-Holland, 19-27.

(5) Boitet C. - Nedobejkine N.
Recent developments in Russian-French Machine Translation at Grenoble.
Linguistics 19, 199-271, 1981.

(6) Boitet Ch. - Nedobejkine N.
Illustration sur le développement d'un atelier de traduction automatisée.
Colloque "L'informatique au service de la linguistique", Université de Metz, juin 1983.

(7) Chappuy S.
Formalisation de la description des niveaux d'interprétation des langues naturelles.
Etude menée en vue de l'analyse et de la génération au moyen de transducteurs.
Thèse de 3è cycle, USMG, Grenoble, juillet 1983.

(8) Donz Ph.
Foll, une extension au langage PROLOG.
Document CRISS, Grenoble, Université II, février 1983.

(9) Gerber R.
Etude des possibilités de coopération entre un système fondé sur des techniques de compréhension implicite (système logico-syntaxique) et un système fondé sur des techniques de compréhension explicite (système expert).
Thèse de 3è cycle, Grenoble, USMG, janvier 1984.

(10) Gerber R.
Les leçons de la traduction humaine pour la traduction automatique.
Rapport de stage au Bureau des Traductions du CANADA, Novembre 1984.

(11) Hayes-Roth F. - Waterman D. A. - Lenat D. B. eds
Building expert systems.
Reading MA, London Addison-Wesley, 1983.

(12) Hobbs J. R.
Coherence and co-reference.
Cognitive Sciences 3, 67-90, 1979.

(13) Isabelle P.
Perspectives d'avenir du groupe TAUM et du système TAUM-AVIATION.
TAUM, Université de Montréal, mai 1981.

(14) Kay M.
Unification grammars.
Doc. Xerox, 1982.

(15) Laurière J.-L.
Représentation et utilisation des connaissances.
TSI 1(1,2), 1982.

(16) Vauquois B.
La traduction automatique à Grenoble.
Document de Linguistique Quantitative nù 29, Dunod, 1975.

(17) Verastegui N.
Etude du parallélisme appliqué à la traduction automatisée par ordinateur.
STAR-PALE : un système parallèle.
Thèse de Docteur Ingénieur, USMG & INPG, Grenoble, mai 1982.

-o-o-o-o-o-o-o-