[From: *Perspectives in artificial intelligence, vol II*, ed. J.A.Campbell and J.Cuena (Chichester : Ellis Horwood, 1989)]

# Current projects at GETA on or about machine translation

**Ch. Boitet**
GETA, BP 68, Université de Grenoble et CNRS, 38402 Saint-Martin-d'Hères, France

## SUMMARY

Research and development projects at GETA all concern machine (aided) translation. The majority of projects are concentrating on improving lingware and software techniques for producing quality translations without an explicit representation of the domain of discourse, and on building small to large running systems. New ideas are introduced to improve the now classical transfer approach, new languages are added, parts of modern linguistic theories are incorporated in the grammars, and better software environments for the rule-based specialized languages are constructed.

Systems of the previous kind are dynamic (computational) in nature. At GETA, some other projects also address the static representation of both grammatical and lexical knowledge. A concrete goal is to build a linguistic workstation usable by linguists, lexicographers and end users. Parts of it have already been prototyped.

Finally, several projects concern new types of MT systems, of the second, third and fourth generations.

## 1. INTRODUCTION

The Groupe d'Etudes pour la Traduction Automatique (GETA), led by B. Vauquois, has been studying the problem of automation of the translation process since 1972, pursuing the work done by CETA since 1962. In the course of this long period, it has developed complete and integrated machine translation systems. However, the design principles of such systems have evolved. In particular, current 'second-generation systems' are a far cry from those of the 1960s.

These experiments have been the foundation of the French M(A)T National Project. By M(A)T, we mean 'machine (aided) translation', thus stressing the fact that these techniques centre around the automation of the production of ‘good enough’ rough translations, rather than around machine aids for translation, although they are present in an integrated environment.

In the first part, we begin with a short presentation of the current principles and technology of second-generation M(A)T systems, such as those implemented at Grenoble. In the following three parts, we present the different kinds of research and development projects pursued by GETA, following the order presented in the summary.

## 2. TYPES OF MT SYSTEMS AND THE APPROACH OF GETA

### 2.1 What to automate?

M(A)T systems are a subset of the computer-aided translation (CAT) systems. Although it may be very interesting to investigate the subtleties involved in the translation of a set of test sentences, and how to emulate them by machine, it is even more promising to attack the real problem of automating (totally or partially) the translation of documents.

Hence, the researchers at GETA have constantly looked for techniques which might lead to a practical implementation of large-scale systems, usable in an operational setting, while at the same time looking into theoretical issues in MT and computational linguistics in general. Good M(A)T systems should offer functional aids in a working translation environment.

### 2.1.1 *Main functions of a translation process*

There are four main successive phases in the processing of a document in a translation environment:

— acquisition or creation of the document;
— rough translation, possibly done in parallel by several translators;
— revision, sometimes done in several passes. For technical documents, a technical revision by a (possibly monolingual) specialist in the field is often required;
— output of the final document, including figures, charts, etc.

(i) *Creation-acquisition: free or controlled*

A document may be created in a translation environment (as in the EC), or sent to it in its final form. As soon as some automation is envisaged, machine aids are in current use for putting the document in machine-readable form (text-processing systems, possibly coupled with OCRs).

Strictly speaking, the creation of a document is not a function of the translation process. However, if this creation could be linguistically controlled by some linguistic process, automation of the rough translation would become a lot easier.

The TITUS (Ducrot, 1982) system illustrates this point. However, this kind of system, using a ‘controlled’ language, neither lexically nor grammatically ambiguous, is now restricted to very specialized domains, and severe constraints are placed on the authors. As we shall see later, one of the most exciting lines of research is to generalize this approach to (good approximations of) really ‘natural’ languages, with their inherent ambiguity.

(ii) *Rough translation: automated or manual*

M(A)T techniques centre around the total or partial automation of the rough translation process. Two main approaches have been tried. In the first (‘pure MT’), translation is done by a program, in batch or interactive mode. GETA, TAUM, SFB-100, METAL all follow the batch line. BYU (ITS), ALPS and WEIDNER have tried the interactive approach (‘human-aided MT’, or HAMT). As with usual human translation, there must be some (human) revision.

The second approach is generally called machine-aided human translation (MAHT). Here, emphasis is on the automation of the translator's office, with specialized text processing systems, fast access to on-line terminological data banks, spelling checkers, etc.

Within the M(A)T approach, two strategies are possible. First, one can try to define some subset of a given natural language as a formal language. Then, an analyser is built. If a given unit of translation is 'legal', it will be translated. If not, it will be rejected. Hence, the automatic system is a 'partial system', because it translates only *N%* of the input. TAUM-METEO, or the first CETA systems (before 1970) are good examples of this strategy.

The second strategy, followed in all current GETA systems, is to build a ‘total system’, which will always attempt to translate 100% of the input, even if it is partially ill-formed with regard to the implemented linguistic model.

(iii) *Revision: human only*

The revision of a document is usually done with the help of full-screen text processing systems, sometimes with the possibility of accessing a terminology file on-line (IBM’s DTAF, WEIDNER, ALPS, TAUM).

However, the automation of the revision function itself has not yet been attempted. It seems that the level of understanding and of general knowledge required to perform even a ‘linguistic’ revision is higher than the one required for translation. This is even more true in the case of ‘technical’ revision.

### *2.1.2 Integrated systems*

There are two other functions which should be automated in a modern CAT environment. First, the management of a large database of documents, together with a record of the actions performed on them (modifications, translations, revisions, etc.). In other words, a translation environment should interface nicely with a textual database system.

Second, in the case of M(A)T, creation, debugging, maintenance and evolution of the ‘linguistic software’, abbreviated here as ‘lingware’ (‘linguiciel’), require a ‘programming environment’, that is, a specialized database centred around one or several programming languages. In our case, these programming languages are specialized languages for linguistic programming (SLLPs).

At GETA, we have developed such an integrated programming system, called ARIANE-78. Only the batch approach has been implemented for the rough translation phase. However, the system also supports a subenvironment for MAHT (THAM in French), used for the revision of the rough machine translations as well as for purely human translation (and revision).

## 2.2 Linguistic principles

### *2.2.1 Multilingualism and transfer approach*

Contrary to general practice, the translation systems developed at GETA have been designed to be multilingual. The 'hybrid' interlingua approach previously used by CETA has been replaced by the 'transfer' approach. This means that translation must be composed of three logical steps:

— monolingual analysis;
— bilingual transfer;
— monolingual synthesis (also called 'generation').

Thus, a given analyser may be used to translate from one 'source language' into several 'target languages', and the same synthesizer ('generator') may be used to translate from several source languages into the same target language. The same division is used in modern 'multitarget' compilers for programming languages.

### *2.2.2 'Implicit' versus 'explicit' understanding*

Everybody agrees that a very good translation requires a very deep understanding of the text. However, this is not achieved, even by good human translators, in particular in technical fields, or there would be no need for revision in the first place!

Hence, the objective of M(A)T systems may rather be set to produce good enough 'raw' translations, that is, translations which may be revised with less than twice the effort needed to revise an average human translation of the corresponding text, and whose (subjective) quality makes them acceptable to revisors in the first place. This goal is already met by state of the art systems, tuned to a given typology and a given domain.

By using such expressions as 'very good', 'good enough', or 'medium', we implicitly suppose the existence of some 'hierarchy' of understanding. In actual fact, understanding cannot be defined in an absolute way, but only with reference to some domain.

As we see it, the hierarchy of understanding is organized around a hierarchy of levels of interpretation. We distinguish between linguistic levels and extralinguistic levels.

#### *2.2.2.1 Linguistic levels*

(1) **morphology**: this is the level of the analysis of words or idioms in terms of morphemes, lexical units, potentialities of derivation, semantic features, valencies, etc.

(2) **syntax-1**: at this level, syntactic classes, such as noun, verb, etc., are associated with words, and syntagmatic classes, such as nominal phrase or verbal phrase, with groups of words. This gives a 'bracketing' of the text (or several in the case of ambiguity), often represented as a tree giving the 'constituent structure'.

(3) **syntax-2**: this is the level of representation in terms of syntactic functions, such as subject, object, attribute, or (equivalently) of dependency relations.

(4) **logico-semantics-1**: at this level, the logical relations between parts of the text are identified. They are sometimes also called inner cases. In the GETA systems, they are usually named ARG0 (logical subject, or 'argument 0'), ARG1 (logical object), etc.

(5) **logico-semantics-2**: the semantic relations, such as consequence, cause, concession, measure, localization, etc., are essential to translate correctly the 'circumstants', as opposed to the 'arguments' of a predicate. On circumstantials, semantic relations are also sometimes called outer cases.

Of course, they may also be attached to the arguments (for example, a logical object may be interpreted as a patient). However, they are less indispensable (for translation), because the semantic relations of the arguments of a predicate are often very difficult to compute, and because, even if they are computed, a good translation may often be obtained simply by using directly the lexical unit of the predicate, plus restrictions on the (semantic features of the) arguments.

This list of levels is not exhaustive. The implemented lingwares also use the representation of a sentence's actualization features (surface tense or abstract time, aspect, modality, determination), type of statement (declarative, interrogative, exclamative, imperative, negative) or emphasis (theme-theme, intensification), etc. However, we consider that all of them are relative to a knowledge encoded in a formal system of a linguistic nature.

Hence, we characterize this type of understanding (at any one or all of the preceding levels) as implicit understanding. Systems relying only on this type of knowledge are of the second (and also first) generation.

#### 2.2.2.2 *Extra-linguistic levels*

(1) **expertise**: here, we refer to some static knowledge about a particular subject matter, consisting in a collection of facts, rules and procedures. This level has also been called 'static semantics', in contrast to the 'feature semantics', incorporated in the linguistic knowledge, and to the '*dynamic semantics',* which is discussed below. MT systems using such an expertise of the domain have been strongly advocated, notably by the Yale school (Schank and coworkers), but no translating systems of this kind have yet been constructed. They would constitute the third generation (see below for further comments).

(2) **pragmatics**: this level is taken to be the highest level of understanding. Pragmatically understanding a document means creating a 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. Interacting intelligently with the author of a document might be a good way to enter this presently imaginary world of near-perfect fourth-generation MT (see below).

Understanding at some extralinguistic level may be called explicit understanding. Typical applications where it is needed include expert systems, which should be able to explain their actions.

However, for translation purposes, implicit understanding is often sufficient. An experimental proof of that is given by the daily practice of human translators. Of course, at the level of a technical revisor, explicit understanding is required.

### *2.2.3 'Multilevel' descriptors and fail-soft strategies*

Second-generation M(A)T systems rely only on the 'linguistic' levels of understanding. In the past, and still in some current systems, these levels are mutually exclusive. By this, we mean that a given unit of translation (sentence, paragraph, or text) will have separate representations for each of the defined levels. This usually leads to a sequential strategy of processing, with all its drawbacks.

During analysis, for example, it is often difficult to compute the semantic relations for all parts of the unit of translation, especially if the size of this unit is large (one or several paragraphs). In the sequential approach, one is then forced to refuse the unit, or else to translate the complete unit at the previous level.

This is why GETA uses multilevel interface structures to represent all the structures computed levels on the same graph (a 'decorated tree'). Detailed examples of this kind of structure have been given elsewhere (e.g. Guilbaud, 1984).

In short, such structures are in effect generators of representations at different levels, and also factorize various types of ambiguities.

Incidentally, this type of structure was first proposed by B. Vauquois in 1974, during sessions of the Leibniz group, which led to the launching of the EUROTRA project by the EEC. Since then, it has been refined and tested, on a variety of applications, including Russian-French, Portuguese-English, English-Malay, English-French, English-Chinese, English-Japanese, French-English and German-French.

### *2.2.4 Necessary specialization to 'sublanguages'*

In translation, by human or by machine, specialization is indispensable in order to obtain good quality. A literary translator is usually at a loss to translate a computer manual. This specialization follows two lines: first, specialization to a certain typology of texts, and second to a certain domain. As humans, M(A)T systems rely on a core of knowledge, plus knowledge specific to the application. As a first approximation, we may say that grammars incorporate the typological specialization, and dictionaries the domain specialization. This is why modularity is essential in the construction of M(A)T systems. The same core should be the base of several versions, tailored to different sublanguages.

One might argue that specialization to a certain sublanguage amounts in fact to the incorporation of some extralinguistic knowledge in a M(A)T system. However, the form of this knowledge is not what is required in order to qualify as expertise, because it is expressed by some combinatorics of classes ('combinatoire de classes', to translate one B. Vauquois' favourite expressions). Rather, we may say that, as in Plato's cave, the real world is 'reflected' in the structure of texts and in their peculiarities. In particular, 'in-house' writing habits correspond to some sociological conditions governing document creation.

## 2.3 Implementation principles

Let us now give a brief introduction to the main principles that have guided the implementation of M(A)T systems at Grenoble.

### *2.3.1 Use of SLLPs*

In principle, there are many ways to implement lexical and grammatical knowledge. In SYSTRAN and other first-generation systems, the assembler or macro assembler level is used.

In 'irst and half' generation systems, the implementation language may be some high-level programming language, such as FORTRAN (SUSY I), COBOL (Saskatoon system), PL/I (ITS), COMSKEE (SUSY II), etc. The drawbacks are evident. In particular, either linguists are burdened with ancillary tasks, such as implementing data and control structures, or they require the help of some 'slave' computer scientists to translate their wishes into working programs, with the result that their desires, incorrectly formulated, are also incorrectly translated.

Nowadays, certain groups are trying to use (also directly) very high-level general programming languages, such as SETL (Novosibirsk), LISP (NTT) or Prolog. Not enough experience has yet been gained to say whether the above criticism on the use of general-purpose algorithmic languages applies or not at this level.

In second-generation systems, and in projected third-generation systems, emphasis is placed on the use of SLLPs, which offer built-in data and control structures, with an underlying powerful mechanism.

This is the case in all 'rule systems', based on (extended) CF-grammars (CETA (Vauquois, 1975), METAL (Chauche, 1974), SFB99, ETL-Lingol), adjunction grammars (LSP,LADL), Q-systems (TAUM), ATNs (BBN (Woods, 1970), TAUM), (extended non-deterministic) finite-state transducer (GETA-ATEF), tree-transducers (Friedman, Petrick, GETA-ROBRA, SFB99-TRANSFO). The built-in data structures are usually particular classes of graphs or hypergraphs, such as decorated trees, Q-graphs, 'harts' (MIND), etc.

Choosing one or more implementation languages for SLLP is another matter. The highest and most efficient level should be selected. There is an inherent conflict in this dual goal. Hence, compromises are made, sometimes by using several implementation languages. For example, ARIANE-78 is implemented in ASM360 (macro assembler) and PL360 for the compilers and interpreters of the SLLPs, Pascal and EXEC2-XEDIT for the other tools, the management of the data-base and the interactive interface ('monitor').

### *2.3.2 Balance between combinatorial and heuristic methods*

As in other fields of AI, the declarative and procedural approaches are in competition. The declarative approach leads to rule systems with an underlying 'combinatorial' algorithm, which produces a set or a subset of 'solutions' in some fixed way. It is best exemplified by analysers built on (extended) CF-grammars, or by Q-systems. The main advantage is the relative ease of programming. However, it is almost impossible to implement powerful heuristics, because, in essence, there is no way to control explicitly the computations of several possibly interdependent solutions.

The procedural approach has been followed in the more recent second-generation systems (‘second and a half’?). For example, the TAUM-Aviation system uses REZO, a Q-graph transducer based on the ATN model. In ARIANE-78, ATEF and ROBRA give even more possibilities of heuristic programming. This added power, however, requires more programming skill.

### *2.3.3 User-oriented programming environment*

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.

In ARIANE-78, this environment is implemented on any user space (a VM/CMS ‘virtual machine’) as a specialized database of lingware 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 database with the users (in French or in English).

Subenvironments are defined to permit the preparation, testing, debugging and maintenance of the lingware, to manipulate the texts, to check the spelling of a list of corpuses (or of individual texts), to produce mass translations, and to revise the translations. The database system ensures the coherence and integrity of all the applications and texts in a given user space (since the system is multilingual, it is perfectly possible to have several translation systems in the same user space, sharing one or several analysers or generators).

It is interesting to note that the needs of linguists led to the creation of such a ‘programming system’ before this type of system became a main theme of research and development in software engineering.

A parallel can be made between this sort of CAT system and compiler-

compiler systems for programming languages. The various SLLPs are in effect tools used to build morphological analysers, structural analysers, transfers and generators.

ATEF, for instance, may be compared with LEX, used to write lexical analysers for programming languages. Of course, the richness of the information contained in each word, and the inherent ambiguity of language, make such a tool more complex: it is necessary to handle large dictionaries (as opposed to small sets of reserved words and of identifiers with no *a priori* content) and to offer advanced control structures, such as non-deterministic programming with or without heuristic functions.

## 3. DEVELOPMENT OF (CLASSICAL) MT SYSTEMS

The ideas exposed above have been and are tested in a variety of MT systems, developed as laboratory experiments, in the framework of various academic cooperations, or for industrial purposes, in the context of the French M(A)T National Project (MAT-NP). The experience gained has also triggered an interesting new idea concerning a way to reduce the apparently inherent quadratic cost of transfer-based multilingual MT systems.

These linguistic developments are accompanied by parallel work on the basic software, which is in constant evolution.

### 3.1 Experiments with MT systems

### *3.1.1 MT systems developed as laboratory experiments*

*3.1.1.1 Types and aims of such systems*

MT systems are developed in the laboratory for four main reasons.

(1) To validate the linguistic methodology for multilingual systems by attacking various languages, preferably pertaining to different groups or families. This is why we started long ago with Russian-French, and are now working on French-Chinese.

(2) For training and testing purposes. This is or was the case for Portuguese-English (POR-ENG), French-English and English-French ‘for the example’ (BEX-FEX and FEX-BEX), English to Chinese and Japanese (IN1-HAN, JAP), Chinese into five other languages (HAN-ENG, FRA, GER, RUS, JAP), and now French-Russian.

(3) To prepare further large-scale development, by developing methods and tools for lingware engineering, and making real experiments. This has been the case for Russian-French since 1982.

(4) To support linguistic research on some language(s) or pair(s) of languages (e.g., German-French, English-Arabic).

Let us give some more details on the two most developed systems.

*3.1.1.2 Russian-French: a real-size operational prototype*

This system is being constantly developed, improved, and used on real texts, in the framework of an operational translation unit (since April 1982). An immediate project is to evaluate it in cooperation with an independent research institute.

The various dictionaries contain some 7500 lexical units in Russian and 5000 in French, which amounts to roughly 30 000 terms in usual dictionaries (remember that a lexical unit is a family of ‘lemmas’, which may correspond to simple or complex terms), because of the richness of the derivational systems used for the two languages.

The grammars cover a (perhaps too) wide range of typologies, ranging from titles and technical abstracts to scientific articles. Technical abstracts are by far the most difficult, owing to the poor quality of writing, the length of sentences, the abundance of apocopes (e.g. ‘the abund. of apoc.’), and the presence of figures and mathematical formulae.

As the texts do not come on magnetic media, it is necessary to type or read them in. In 18 months (April 1984-October 1986), around 200000 running words, or 1.5 million characters, were inputted in the textual database, half of them using an OCR (in cooperation with the Paris-based CERTAL research group).

In one month (September 1986), around 835 abstracts and texts, or 97 000 running words, were translated or retranslated on a shared minicomputer (IBM 4331-2 with 4Mb under VM/CMS), to present a set of coherent results in the final report of a contract with the Ministry of Defence.

This system is also used as support for contrastive studies by its main author, N. Nedobejkine.

Some examples of machine translations with manual on-screen revisions are given in the appendix. In the first, long example, some words appear between brackets, e.g. ⟨"m_AMX-30⟩. This is because their lexical unit was not in the Russian dictionaries. However, in most cases, the morphological sub-grammar for unknown words has analysed them correctly. Here, the special prefix "m_ introduces a trademark, hence, an inanimate proper noun.

The second, shorter example, exemplifies the improvement obtained by modifying the lingware. Here, three or four dictionary items have been corrected in transfer and generation. For example, 'introduction dans' is replaced in the second translation by ‘introduction à’, and ‘golografia’, having been indexed, is no longer decomposed in ‘golo-’ (nude) and ‘grafia’ (graphy), and is correctly translated as ‘holography’.

The virtual CPU time used for translating one word is about 1.4 Mipw (million operations per word). In terms of elapsed time on a shared 4331-2 (4 Mb, 0.4 Mips), it amounts to 15 min per page of 250 words, or $3.50 if taking all computer-related costs in consideration. On-screen revision of the long example (TANK2) has taken less than 15 min, including terminological discussion, and using the standard ARIANE-78 REVISION subenvironment (XEDIT in two or three-windows configuration, with some useful macros associated with certain keys).

### *3.1.1.3 German-French: a feasibility study supporting linguistic research*

This system uses the same generation of French as the former one. The German side (analysis and transfer) is still a prototype, covering a restricted typology and based on a small lexicon (around 2000 lexical units, or 4000 terms for German).

A particular feature is its development by two independent researchers, one in Paris (G. Stahl) and the other in Grenoble (J. Ph. Guilbaud). The first author has developed the structural analysis, and the second the morphological analysis and the transfer.

No large-scale development is planned for the moment. Rather, J. Ph. Guilbaud is now using this system as support of a study on the possibility of integrating some results of the linguistic research pursued by J. M. Zemb (Collège de France, Paris), mainly on the contrastive French-German grammar, but also on the fundamental notions underlying the grammatical descriptions.

### *3.1.2 MT systems developed in academic cooperation*

#### *3.1.2.1 Aims of such systems*

B. Vauquois has always sought international cooperation, in order to confront different points of view on natural language processing, and to try them experimentally, MT being perhaps the best benchmark.

In the 1960s, permanent contacts were established with scholars from the USA, the USSR, Japan, Czechoslovakia, and almost all West-European countries. Long stays by Czech and Japanese colleagues strengthened those links, but no common systems, or even mockups, were built.

In the 1970s, GETA developed a truly language- and theory-independent software environment for building multilingual MT systems, ARIANE-78. This tool (or its preceding versions) supported the development of a series of experiments, all done in cooperation with foreign colleagues: several analyses of French (J. Weissenborn and E. Stegentritt, Saarbriicken), an analysis of Portuguese with a mockup transfer and generation into English (P. Daun Fraga, Campinas), a structural analysis of Japanese (R. Shimamori), and prototypes from or into Chinese (Feng Zhi Wei, Yang Ping, Beijing).

From 1979 onwards, a long-term cooperation was started with Malaysia and Thailand, producing two prototypes sharing the same analysis (English--Malay and English-Thai). We give more details about these systems below.

#### *3.1.2.2 English-Malay*

This effort started in 1979, after a visit of Professor B. Vauquois to Malaysia, at the initiative of Professor Tan Wang Seng (USM, Penang). The outline of the project was defined and some common understanding on the methodology was reached during a one-month stay of Professor Tong Loong Cheong and Dr Chang May See. The ARIANE-78 system was installed at USM.

In 1980, B. Vauquois, P. Daun Fraga and Ch. Boitet stayed at USM for two months. Starting only from previous desk research (specifications), B. Vauquois, P. Daun Fraga and our two Malaysian colleagues produced a working English-Malay prototype in 6 or 7 weeks, while the author was busy producing an English version of ARIANE-78. At the end of August, an international seminar convened at USM, and the prototype was used extensively for demonstrations and experiments.

Since then, the group at USM has grown and become permanent. By the end of 1985, the English-Malay system had reached the stage of laboratory prototype. It was systematically evaluated, with a resulting acceptability rate of 76% (Tong, 1986). The stage of operational prototype should be reached at the end of 1988s. The system aiming mostly at translating computer-related technical material.

#### *3.1.2.3 English-Thai*

Initiated during the 1980 USM seminar, this cooperation started effectively in 1981. Several Thai universities participate in this effort (Rakhamhaeng, Chulalongkorn, Prince of Songkhla, etc.). The stage of laboratory prototype should be attained at the end of 1987.

Of course, the peculiarities of the Thai writing system have been a challenge. However, the computer scientists from Chulalongkorn have connected the ARIANE-78 system to special I/O (input/output) devices, so that translations can be produced in Thai characters.

### ***3.1.3 MT systems developed for industrial purposes***

#### *3.1.3.1 The French MAT-NP (National Project)*

The French Machine (Aided) Translation National Project (MAT-NP) started in November 1983, and ended in February 1987. Financing of the project was 50% public and 50% private. Public financing and control was centralized by ADI (Agence de l'Informatique), while the private firm SG2 and its subcontractors (including the SONOVISION and B'VITAL firms) invested the rest and built the system. The first official presentation of CALLIOPE-AERO was made at EXPOLANGUES (Paris) in February 1986.

For the first development, it has been decided to build a French-English system tailored to aviation manuals of the kind produced by SONOVISION, which are in machine-readable form, and for which the appropriate terminology exists in both languages.

After EXPOLANGUES, it has been decided to begin the development of CALLIOPE-INFO (English to French for computer-related material), which was until then only an option of the project. A first version was obtained at the end of the MAT-NP, and should be expanded, if adequate funding can be found. The first translations were produced in February 1987.

The core of the architecture of the lingware and the software comes from previous work done at GETA, but new tools and techniques have been added. The use of SCSGs is quite notable.

*Use of SCSGs for static specification of dynamic grammars*
B. Vauquois and S. Chappuy developed a formal model **'static grammars'** (Chappuy, 1983; Vauquois and Chappuy, 1985) before the start of the MAT-NP.

A SCSG (structural correspondence static grammar) describes the correspondence between the strings of a natural language and the corresponding interface structures. Such a description is neutral with respect to analysis and generation, and does not express any particular strategy for computing the correspondence.

During the first phase of the MAT-NP, from November 1983 until November 1984, only SCSGs of French and English were developed, and no procedural grammars. Special care was taken to describe a reasonable core grammar and to study in detail the particularities of the considered typology. As for any sublanguage, it offered grammatical constructions which would be judged ungrammatical in other contexts.

These SCSGs have been used later as reference and documentation while writing the very large dynamic grammars.

### *3.1.3.2 CALLIOPE-AERO (French-English)*

The size of grammars and dictionaries is obviously heavily dependent on the considered application. In the case of CALLIOPE-AERO, the typology of the manuals includes almost all normal syntactic constructions, with the exception of interrogative clauses, relative clauses introduced by 'dont' and imperative forms of verbs (replaced by the infinitive form), and a lot of special phenomena.

As far as the lexicon is concerned, a preliminary study of the corpus had led to the estimation that 6000 general terms and 15 000 terminological terms would be necessary for the system to be usable.

The dictionaries comprise now around 8000 lexical units in the running system (more in the lexical database), or about 12000 terms, in both languages. Counting the source lines (written in ATEF for morphological analysis, TRANSF for lexical transfer and SYGMOR for morphological generation), we arrive at a total of about 55 000 lines.

As far as the grammars are concerned, there are about 175 rules for morphological analysis (AM), 600 for structural analysis (AS), 90 for structural transfer (TS), 200 for syntactic generation (GS), and 20 for morphological generation (GM). In terms of source lines, we find, for the grammatical part of the same phases, a total of around 4500 (AM), 18000 (AS) 2300 (TS), 5600 (GS) and 470 (GM).

If we compare this with the size of a compiler for some programming language, written in metalanguages such as LEX and YACC, we see that the lingware engineering effort required to create and maintain such an MT system exceeds by far what is required for a compiler. This is made even worse by the fact that natural language is not fixed by decree, but changes, and is not defined by our grammars, but only approximated. Contrary to the case of a compiler, the grammars and dictionaries of an MT system must be easily modifiable, by linguists and not by computer scientists. Hence, modularity in the SLLPs and user-friendliness of the programming environment are essential.

#### *3.1.3.3 CALLIOPE-INFO (English-French)*

This system aims at translating computer manuals. The SCSGs of French and English are of course reused, and enriched for two reasons:

- — the typology changes, hence, more grammatical phenomena must be accounted for;
- — ambiguity 'boards' ('planches', or two-dimensional representations of rules in a SCSG) are being constructed for English, as they have been for French. They are useful for analysis, where they help design the disambiguation (dynamic) rules.

The dynamic grammars for the analysis of English and for the generation of French are offshoots from those developed by GETA, in-house or in cooperation.

Indexing of the terminology has been done by the SONOVISION firm, as for CALLIOPE-AERO. Around 3000 specialized terms have been incorporated in the first version.

### *3.1.4 A way to reduce the cost of transfer-based multilingual systems*

In the context of the Russian-French, German-French and English-French systems, a unification of the generators of French has been attempted, as they had diverged somewhat from their common root, the generator of Russian-French. This unification is accompanied by a deep restructuring of the syntactic generation phase, with the aim of making composition of machine translators possible in the context of multilingual transfer-based MT systems.

The main disadvantage of the transfer approach is that $N(N$-1) transfers must be written to translate between $N$ languages, as opposed to the $N$ analysers and $N$ generators.

Of course, one might envisage attempting to translate everything into a natural language, which would act as 'pivot'. However, structural ambiguities would multiply. An artificial 'natural' language such as Esperanto is even worse, because of the need to build a complete technical vocabulary. A satisfactory 'logical' language, in which everything would be disambiguated, has yet to be devised and equipped with the appropriate 'universal' vocabulary.

An idea, then, is to 'compose' transfers. However, this cannot be done immediately. As a matter of fact, the input to a generator is a target interface structure which is not in general of exactly the same type as the source interface structure produced by an analyser. This is because the final form of the text to be generated is not yet fixed (paraphrases are possible), because polysemies not reduced by the transfer may appear as a special type of enumeration, and because the transfer may transmit to the generator some advice or orders (relative to the possible paraphrases), by encoding them in the structure.

Instead of producing directly the surface tree (to be passed to the morphological generator), the new technique consists of **producing a source interface structure of the target text** as an intermediate result, which may be sent to a transfer going from the target language to still another language.

For example, consider the four main Romance languages (French, Spanish, Italian, Portuguese). Taking any one of them as 'stopover', it would only be necessary to build 6 transfers instead of 12.

Note that, in general, a minimum of *N* transfers would be enough. However, the associated 'ring' organization leads to an average of (*N*-2)/2 stopovers (and a maximum of *N*-1) for a language pair. With a unique 'stopover', 2 (*N*-1) transfers are necessary, but the number of stopovers in a given translation is always 0 or 1 (an average of 1-2/*N*).

A pragmatically better organization can be envisaged. For instance, consider the nine languages of the European Community, and divide them in three groups: four Romance languages, four Germanic languages, and Greek. Instead of constructing 72 transfers, it might be enough to construct only 16:6 in each group of four, and 4 between the 'pivots' of the groups, e.g. English-Greek, Greek-French, French-English, English-French. No translation would then require more than two stopovers.

### *3.1.5 Projects in basic software for MT*

#### *3.1.5.1 ARIANE-78.4 and ARIANE-85.2 on mainframes and micros*

ARIANE-78.4 has been chosen as support for all developments of the MAT-NP (CALLIOPE) project. Its successor ARIANE-85.2 is beginning to replace it, with the advantage of added modularity and better handling of big dictionaries (Boitet *et al.,* 1985).

Until recently, this system ran only on mainframes and minis. However, the compactness and relative speed of the code have made it quite easy to adapt the complete ARIANE-78.4 system to the IBM PC/AT-370 (under VMPC). ARIANE-85.2 will soon follow. Exactly the same programs run on the micro and on the mainframe. The minimal configuration uses a fixed 20Mb disk, the 370 kit (two cards), and a 3278/79 emulation card. A memory extension of at least 1.5 Mb may be added and used for paging, which speeds it up considerably.

Of course, the speed of the 370 card (0.1 Mips) does not yet make it possible to consider such a configuration for the **production** of translations. However, it is quite adequate to build MT prototypes, and may be used as low-cost hardware for academic cooperation.

In order to use the PC for producing translations, it would be necessary for IBM to modify VMPC in such a way that the memory extension could be used **directly** as real memory (as Lotus does with the EEMS extension). Now, only the basic 512 Kb can be used as real memory. Available memory cards might then be used to get a real memory of 4 Mb (VMPC limit for the virtual memory). Also, it is likely that faster 370 cards will be produced, and perhaps adapted to the faster hardware of the new PS/2 series.

### *3.7.5.2 Work on ARIANIX*

In the framework of the French MAT-NP, it was decided to push ahead a project which had been prepared at GETA over recent years, but advanced slowly, due to the lack of resources. This new basic software for MT is constructed in Le_lisp, a French dialect of LISP produced by INRIA and converging toward Common Lisp. Here are the main features of this future system:

— A unified SLLP, called TETHYS for the moment, will replace the four SLLPs of the previous systems. It will contain several 'rule engines', in order to make it upward-compatible with ARIANE.85, and also to open it to rule systems developed elsewhere (such as Q-systems).

— The system will be basically multilingual, down to the level of characters.

— Access to the implementation language will be possible from within the rules written in TETHYS.

A very large set of characters has been defined. Several properties are attached to each character. Among them, the natural language (neutral, or Math, French, English, Spanish, Japanese, ...), the name of the (usual) character set (keywords, special, Roman, Greek, Cyrillic, Thai, hiragana, kanji, hanze, hangul, ...), its case and diacritics, if any, and some information about ‘stress’ (italic, bold, underlined).

These properties have been selected because of their importance for linguistic processing. They are all important for defining the dictionary order of words. For example, the information on the natural language permits one to consider ‘ch’ as one letter in Spanish, and also to display a TETHYS keyword, considered as one character, in the appropriate language.

Two preliminary implementations have been prepared on a SM-90 under SMX (a version of UNIX), one in Le_lisp, using property lists, and one in Pascal, using a 32-bit representation.

Also, a norm for representing multilingual documents has been defined, in order to be able to use information relative to the logical structure of a text in the linguistic processors. For example, a table containing textual elements in its cells should not be represented as a sequence of lines, each made of fragments of textual elements separated by tab characters, but as a construct of matrix type, where each textual element appears contiguously.

## 4. BUILDING A LINGUISTIC WORKSTATION

A lot of work has been done on translator workstations (Melby, 1982). Such stations always appear as extensions of classical text-processing systems, with extremely limited linguistic capabilities. By contrast, a linguistic workstation would be centred around non-trivial linguistic capabilities, and offer powerful extensions to existing, programmable text processors of various kinds.

The basic capabilities envisaged relate to the specification of dynamic grammars by static grammars, to the construction of multitarget integrated dictionaries, and to the structural study of corpuses, translator's aids being seen as subproducts.

### 4.1 Grammatical specification with SCSGs

Recently, a first environment for building SCSGs has been designed and implemented on a Macintosh+ by Y. F. Yan in 1987, under the guidance of F. Peccoud. It incorporates a methodology for writing the different components of an SCSG (attributes, axioms, boards), while handling at the same time the corpus investigated and the examples from the corpus appearing in the boards.

There is still much work to be done in this direction, before a complete environment can be offered to linguists. In particular, it would be useful to relate directly a dynamic grammar being executed on a text to the boards containing the specification of the partial correspondences computed by the executed dynamic modules or individual rules of the dynamic system. For this, AI workstations and software tools are envisaged.

### 4.2 Construction of lexical databases (MIDs)

Ultimately, the cost of MT systems lies essentially in their dictionaries, which are quite difficult to construct and to maintain. Since 1982, GETA has been working on fork integrated dictionaries, now called multitarget integrated dictionaries (MID). They integrate the terminological and the integrated dictionaries' grammatical aspects, as well as the 'usual' and 'coded' information used in computer applications. A given dictionary contains terms in one language, with the information concerning that language, and the translations of the different meanings in one or several languages.

In the prototype version, an item is a tree, structured according to a grammar, from which an analyser is derived automatically, using a tool developed by Y. Lepage, as well as the definition of the image database structure, in the DDL (data definition language) of a commercial DBMS (now CLIO from SYSECA, but others could be added). From a given item, a DML (data manipulation language) program is generated automatically, to load the image of the item in the DBMS. There is an ongoing project to build a complete user interface, written in Prolog, and easy to interface with any reasonably powerful DBMS.

In order to get some practical experience, 3000 terms in telecommunications have been written in MID format in 1986, in three languages (French, Japanese, English), in cooperation with the French Telecommunications (DGT) and with KDD (Japan). The corresponding 9000 entries have been keyed in on a Macintosh+, using Kanji-Talk.

The work to be done remains immense, in practice and in theory. It is comparatively easy to generate MT dictionaries from MIDs. However, the reuse of existing lexical information encoded in machine dictionaries developed for computer applications is a very difficult task.

We have begun to study how to extract the lexical information from our Russian-French system and to put it in MID format. The preliminary results obtained so far show this to be more difficult than we imagined at the beginning, because some information is implicit or absent from the codes, and has to be added (in the form of comments) in order to be accessible by a program.

### 4.3 Structural study of corpuses

This line of research has been hardly touched by GETA until now. Some elementary tools (concordances) have been constructed, and some methods used elsewhere in particular at the Bureau of Translations (Ottawa), have been tested in experiments.

However, we are trying to get researchers to explore this further, in the context of such a linguistic workstation. The idea is to use a tree editor equipped with pattern-matching facilities to explore linguistic trees (source interface structures, for example) associated with the units of text (paragraphs, sentences,...) under consideration.

A powerful tree transformational editor, TTEDIT, has recently been developed by J. C. Durand in REXX-XEDIT. The rules look like simplified ROBRA rules, with an addition to move a cursor in the edited forest. Commands may be grouped in packages analogous to usual editor macros. The fact that left-hand sides are schemas with variables gives TTEDIT considerably more power than graphic editors with which one manipulates only the drawing of a tree (or a graph).

Such a tool might also be used for exploring and modifying intermediate results produced by an existing analysis or transfer step, in the context of the validation of dynamic grammars constructed from SCSG specifications.

## 5. DESIGNING NEW TYPES OF MT SYSTEMS

Finally, several projects concern new types of MT systems. Grafting an expert system onto a second-generation MT system can produce an expert translator system of the third-generation type. Replacing the usual purely automatic analyser by a linguistic editor interacting with the author should lead to fourth-generation systems. Finally, second-generation systems might be completely reshaped as expert systems relying only on static linguistic knowledge. These three approaches are or have been followed concurrently by small teams.

### 5.1 Going to third generation by accessing domain-specific expertise

The linguistic and paralinguistic knowledge incorporated in modern second-generation MAT systems is quite enormous. Unless the knowledge of a given domain is extremely limited, it is not feasible to put this information directly into such systems. We have proposed (Boitet and Gerber, 1984) to graft corrector expert systems onto existing MT systems. The corrector system would detect 'problem patterns' in the source interface structure, convert them into questions OB the domain (represented as a knowledge base, an expert system, or as a simple database), and modify the structure according to the answer.

A small prototype has been built by Gerber (1984) connecting a small

English-French system to such a corrector system, written in Prolog. The problem here is that developers of MT systems have no time and no competence to build knowledge bases for technical domains. However, significant improvements in translations cannot be obtained if the knowledge base is too small. To continue this line of research, we are looking for a situation where both an adequate knowledge base and texts to be translated would be available.

### 5.2 Going to fourth generation by interacting with the author

Even if a large knowledge base is available, no machine analysis of a text can be 100% correct, because new knowledge is usually introduced by the translated text. However, no adequate learning method has yet been devised to modify and enrich the knowledge base dynamically. Even if one did exist, the communicative character of texts, their pragmatic aspect, would not be handled satisfactorily.

As an alternative or complement to the method above, we propose to return to the interactive approach. The essential difference from previous schemes such as that of ITS (BYU, Provo) is to consider an interaction with the author of a document, and not with a specialist of both the domain and the MT system, without imposing a too restricted controlled language, as in the TITUS system (Ducrot, 1982).

In his thesis, Zajac (1986a) has proposed an organization of the dialogues with the author, in an MT system using a traditional morphological analyser, but where the structural analyser would be replaced by a syntactico-semantic editor parametrized by the SCSG used to specify the usual purely automatic editor-analyser. The structure of the dialogues is partly an elaboration of Tomita’s methods for handling ambiguities (reduction of the number of questions asked (Tomita, 1984)), and partly original (paraphrases are proposed, and no questions are asked in terms of the grammar).

There are many situations where documents are produced within a computerized environment, and where the editor must follow some norm of writing (e.g., the AECMA norm for aircraft manuals written in English). Such an approach might lead to very good translations, transfer and generation being done with no interaction, as now. Of course, not all polysemies and ambiguities can be reduced in this way, because some are of a contrastive nature. Hence, revision would still be necessary to obtain 'guaranteed' translations, as is the case with very good human translations of high importance.

### 5.3 Organize the linguistic knowledge as in expert systems

The new types of MT systems just presented pertain respectively to the third and fourth generations. However, it is also interesting to investigate new approaches to the purely linguistic part of the MT process, even of the second generation. We have already mentioned a new design of the generators permitting us to compose translators. A more ambitious goal is to reshape completely the grammars and dictionaries, in order to represent the linguistic knowledge as in an expert system.

In all existing systems, much procedural knowledge is included in the rules and in the control of their application. The main part of the procedural knowledge is concerned with solving class and structure ambiguities. Usually, ambiguities are eliminated or ignored rather than handled.

In first-generation systems, they are eliminated as soon as they are encountered. In second-generation systems using the ‘filter’ technique, their number is reduced by the combinatorial application of the rules, and one final solution is arbitrarily chosen (Veillon, 1970). Sometimes (Slocum, 1984) a weighting device is used to rank them, but it is extremely difficult to assign weights in a meaningful way. In other systems (GETA, or Kyoto), heuristic programming is used to follow one or a few solutions at a time, with the possibility to backtrack (locally) or to ‘patch’ later (Vauquois, 1979, 1983). Programming in this way is quite delicate, although it makes it possible to get some direct handling of the ambiguities.

What seems to be needed is a way to represent ambiguities such that the major part of the linguistic programming could be expressed without bothering about them, through rules of a declarative nature (like the ‘boards’ of the SCSGs), and that ambiguities might appear as problems treated by a separate mechanism of metarules which would describe solutions to individual problems and be used by the linguists at appropriate points.

A first effort in this direction has been made (Verastegui, 1982), before SCSGs were introduced. A promising line of research consists of using the boards of the SCSG as the declarative rules of a (combinatorial) analyser, and to precompute as many ambiguity cases as possible, much in the same way as is done when testing that a context-free grammar is LL(1). Then, there would be some variant of the SLLP to describe how to solve those identified problems. Some default solution might be used in the absence of a good enough expert rule, and for the ambiguities which could not be precomputed.

Note that this scheme might be used in second-, third- and fourth-generation systems, as the solutions described might involve a call to some knowledge base or an interaction with a human.

## 6. CONCLUSION

The tradition of GETA has always been to pursue at the same time fundamental and applied research on machine translation or related topics. Big experimental systems have been built, and techniques linked with AI and modern linguistics are being investigated.

However, and this is also true of almost all work in computational linguistics (CL), they are just techniques. CL, of which MT is a part, could only attain the status of an experimental science if experiments would be made in order to prove or disprove scientific hypotheses, or to discover new phenomena, calling for new hypotheses, etc., very much like in physics. However, things built nowadays, such as MT systems or natural-language interfaces, do not seem to give any new insight into scientific questions, even if they are useful in practical settings.

From this point of view, building a linguistic workstation of the kind described above would be a very important goal.

## REFERENCES

Bachut, D. and Verastegui, N. (1984) Software tools for the environment of a computer-aided translation system, *Proc. of COLING-84, ACL, Stanford, July 2-6,1984,* pp. 330-334.

Bennett, W. and Slocum, J. (1984) *METAL: The LRC Machine Translation System,* Linguistic Research Center, Austin, Texas, USA. September 1984.

Boitet, Ch. (1976) Un essai de réponse à quelques questions théoriques et pratiques liées à la Traduction Automatique. Définition d’un système prototype. *Thèse d’Etat,* Grenoble.

Boitet, Ch. (1984) Research and development on MT and related techniques at Grenoble University (GETA). *Lugano Tutorial on Machine Translation,* April 1984.

Boitet, Ch. and Gerber, R. (1984) Expert systems and other new techniques in MT. *Proc. of COLING-84, ACL, Stanford, July 2-6, 1984,* pp. 468-471.

Boitet, Ch. and Nedobejkine, N. (1981) Recent developments in Russian-French machine translation at Grenoble. *Linguistics* **19,**199-271.

Boitet, Ch. and Nedobejkine, N. (1983) Illustration sur le développement d’un atelier de traduction automatisée. *Colloque "l'informatique au service de la linguistique", Université de Metz, France, juin 1983.*

Boitet, Ch., Guillaume, P. and Quezel-Ambrunaz, M. (1978) Manipulation d’arborescences et parallélisme: le système ROBRA. *Proc. of COLING-78, Bergen, August 1978.*

Boitet, Ch., Guillaume, P. and Quezel-Ambrunaz, M. (1982) ARIANE-78: an integrated environment for automated translation and human revision. *Proceedings COLING-82, Prague, July 1982,* North-Holland, Linguistic Series No. 47, pp. 19-27.

Boitet, Ch., Guillaume, P. and Quezel-Ambrunaz, M. (1985) A case study in software evolution: from ARIANE-78 to ARIANE-85. *Proc. Conf. on Theoretical and Methodological Issues in MT, Colgate University, Hamilton, N.Y., August 1985.*

Chappuy, S. (1983) Formalisation de la description des niveaux d’interprétation des langues naturelles. *Thèse de 3è cycle,* Grenoble.

Chauche, J. (1974) Transducteurs et arborescences. Etude et réalisation de systèmes appliqués aux grammaires transformationnelles. *Thèse d’Etat,* Grenoble, décembre 1974.

Chauche, J. (1975) Les langages ATEF et CETA. *AJCL Microfiche* 17, 21-39.

Clemente-Salazar (1982) Etudes et algorithmes liés à une nouvelle structure de données en TA: les E-graphes. *Thèse de Docteur-ingénieur,* USMG & INPG, Grenoble, May 1982.

Clemente-Salazar (1982) Une structure de données intéressante en TA: les E-graphes. *Proc. of COLING-82, Prague, 1982,* North-Holland.

Ducrot, J. M. (1982) TITUS IV. Proc. of the EURIM-5 Conf., Versailles. In: P. J. Taylor (ed.), *Information Research in Europe,* ASLIB, London.

Gerber, R. (1984) 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, January 1984.

Guilbaud, J. Ph. (1984) Principles and results of a German-French MT system. *Lugano Tutorial on Machine Translation,* April 1984.

Melby, A. (1982) Multi-level translation aids in a distributed system. *Proceedings COLING-82, Prague, July 1982,* North-Holland, pp. 215-220.

Mozota (1984) Un formalisme d'expressions pour la spécification du contrôle dans les systèmes de production. *Thèse de 3è cycle,* Grenoble, juin 1984.

Slocum, J. (1984) METAL: The LRC Machine Translation System. *Lugano Tutorial on Machine Translation,* April 1984.

Tomita, M. (1984) Disambiguating grammatically ambiguous sentences by asking. *Proc. of COLING-84, ACL, Stanford, July 2-6, 1984,* pp. 476-480.

Tomita, M. and Carbonell, J. (1986) Another stride towards knowledge-based machine translation. *Proc. of COLING-86, IKS, Bonn, August 25-29,* pp. 639-642.

Tong, L. C. (1986) English-Malay translation system: a laboratory prototype. *Proc. of COLING-86, IKS, Bonn, August 25-29, 1986.*

Vauquois, B. *(1975) La Traduction Automatique à Grenoble,* Document de Linguistique Quantitative 29, Dunod, Paris.

Vauquois, B. (1979) *Aspects of Automatic Translation in 1979.* IBM-Japan, Scientific Program, July 1979.

Vauquois, B. (1983) Automatic Translation. *Proc. of the Summer School on the Computer and the Arabic Language, Rabat, October 1983,* Chapter 9.

Vauquois, B. and Chappuy, S. (1985) Static grammars. *Proc. Conf. on Theoretical and Methodological Issues in MT, Colgate University, Hamilton, N.Y., August 1985.*

Vauquois, B. and Boitet, Ch. (1985) Automated translation at GETA (Grenoble University). *Computational Linguistics* 11(1) 28-36.

Veillon, G. (1970) Modèles et algorithmes pour la traduction automatique. *Thèse d'Etat,* Grenoble, 1970.

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

Woods, W. (1970) Transition network grammar for natural language analysis. *CACM* 13(10) 591-606.

Zajac, R. (1986a) Etude des possibilités d'interaction homme-machine dans un processus de traduction automatique. Spécification d'un système d'aide à la rédaction en vue d'une traduction par machine. Définition d'un langage de spécification linguistique. *Proc. of COLING-86, IKS, Bonn, August* 25-29,*1986, pp. 393-398.*

Zajac, R. (1986b) SCSL: a linguistic specification language for MT. *Proc. of COLING-86, IKS, Bonn,* August 25-29, 1986, pp. 393-398.

## APPENDIX: EXAMPLES OF TRANSLATIONS

Russian-French is designed to produce ‘crude’ translations, not necessarily revised, but good enough to give the content of the source text in an intelligible and reliable way.

We give two examples, as they are produced by the ARIANE-78 system, on our IBM mainframe, using the SCRIPT-DCF formatter. ARIANE's result has been transmitted to the Macintosh used to produce this document by Kermit, over an SM-90. This explains why the columns are not as well aligned as on the IBM 3262 printer used in the laboratory, in spite of Yan Yong Feng's valuable aid.

### A long example, with source text, machine translation and human revision

LANGUES DE TRAITEMENT: RUB-FRB

— ( TRADUCTION DU 24 SEPTEMBRE 1986 9H31MN26S)—

VERSIONS : ( A : 21/07/86 - T : 21/07/86 - G : 21/07/86 )

--(REVISION DU 6 NOVEMBRE 1986 10H58MN54S) —

-TEXTE SOURCE--

Na tanke ustanovlen 12-cilindrovyij mnogotoplivnyij dizelq s turbonadduvom i zhidkostnyi oxlazhdeniem "m_HS-110. Mexanikheskaya transmissiya "m_5SD_20 vklyukhaet v sebya pyatistupenkhaluyu korobku peredakh, avtomatikheskoe centrobezhnoe sceplenie s yelektroprivodom, mexanizm povorota. tormoz s gidroprivodoni i planetarnyie bortovyie peredakhi. Podveska opornyix katkov torsionnaya. Na pervom i pyatom katkax ustanovlenyi gidroamortizatoryi . Tank oborudovan sistemoj zathilyi ot oruzhiya massovogo porazheniya , avtomatikheskoj sistemoj pozharotusheniya. OPVT pozvolyaet tanu preodolevalq po dnu vodnyio pregradyi glubinoj do 4 m

-TEXTE TRADUIT -

Sur le char on a installé un diesel polycarburant à 12 cylindres avec la suralimontation par turbosufflante et le refroidissement par liquide <'m_HS-110>. La transmission mécanique <'m_SSD_200D> comprend la boite à cinq étages de vitesses, embrayage centrifuge automatique avec la commande électrique, mécanisme de direction, le frein avec la commande hydraulique et les engrenages de bord planétaires La suspension de galets porteurs est à barre de torsion. Sur les galets premier et les galets cinq on a installe les amortisseurs hydrauliques. Le char est équipé du système de protection contre l'arme de destruction massive, le système automatique de lutte contre incendie. Le schnorchel permet au char

-TEXTE REVISE-

Sur le char, on a installé un diesel polycarburant à 12 cylindres avec suralimentation par turbosoufflante et refroidissement par liquide HS-110. La transmission mécanique SSD_200D comprend une boite de vitesses à cinq étages, un embrayage centrifuge automatique avec commande électrique, un mécanisme de direction, un (rein avec commande hydraulique et des engrenages de bord planétaires.

La suspension de galets porteurs est à barre de torsion. Sur les premier et cinquième galets, on a installé des amortisseurs hydrauliques. Le char est équipé d'un système de protection contre l'arme de destruction massive, et d'un système automatique de lutte contre

Na baze tanka “m_AMX-30 sozdanyi mostoukhladkhik “m-AMX-3OPP remontno-yevakuacionnaya mashina “m_AMX-30D, samoxodnaya zenitnaya ustanovka “m_AMX-30SA. samoxodnyij zenitnyij raketnyij kompfeks “m_AMX-30R, samoxodnaya puskovaya ustanovka raketyi "Ptuton" i samoxodnoe orudie “m_AMX30GT.

S 1982 g. v vojska nakhal postupatq modernizirovannyij obrazec tanka, polukhivshij oboznakhenie "m_AMX-30B2, V otlikhie ot svoego predshestvennika on snabzhen vmesto 12,7-mm pulemeta 20-mm avtomatikheskoj pushkoj, kotoraya po uglu vozvyisheniya takzhe imeet nezavisimyij privod. Tank “m_AMX-30B2 osnathen sovremennoi sistemoj upravleniya ognem "m_APX-M581. V sostav eevxodyat lazemyij pricel-dalqnomer, yelektronnyij ballistikheskij vyikhislitelq, yelektrogidravlikheskij stabilizator voonuzheniya leplovizionnyie priboryi nokhnogo videniya. V boekomplekt pushki vklyukhen novyij bronebojnyij podkalibernyij snaryad , broneprobivaejnostq kotorogo na dalqnosti 2000 m sostavlyaet okolo 350 mm po normali. Podvizhnostq modernizirovannogo tanka ulukhshena blagodarya ustanovke gidromexanikheshoj transmissii “m_ENC-200. V xode rabot po dalqnejshemu sovershenstvovaniyu tanka “m_AMX-30 byil sozdan osnovnoj tank

de franchir sur le lond les obstacles fluviaux de la profondeur jusqu'à *4 m.*

Sur la base du char < “m_AMX-30>on a créé pontonnier <“m_AMX-30PP>. véhicule de dépannage <"m_AMX-30D>. canon antiaérien automobile <"m_AMX-30SA>, un ensemble de fusée antiaérien automobile <“m_AMX-30R>. rampe de lancement automobile de la fusée <"Pluton"> et canon automobile <"m_AMX-30GT>.

Dès 1982 dans l'armée a commencé à entrer modèle modernisé du char qui a reçu le nom <“m_AMX-30B2> Contrairement à son prédécesseur il est équipé au feu de la mitrailleuse de 12,7 millimètres du canon mitrailleur de 20 millimètres qui sur l'angle d'élévation aussi a une transmission indépendante. Le char <“m_AMX-30B2> est équipé du système moderne? actuell (Genre?)? de commande du feu <"m_APX M581>. De sa composition font partie les viseurs telemètre a laser , un ordinateur balistique, un stabilisateur électrohydraulique de l'armement, les instruments infrarouges d'une vision nocturne. Un nouvel obus sous-calibre perforant dont la force de pénétration sur la distance de 2000 m constitue près 350 mm selon les normes est incorporé? branché? dans la dotation en munitions du canon La mobilité du char modernisé est amelioree grâce à

l'incendie. Le schnorchel permet au char de franchir sur le fond les obstacles fluviaux de profondeur jusqu'à 4 m.

Sur la base du char AMX-30, on a créé le pontonnier AWX-30PP, le véhicule de dépannage AMX-30D, le canon antiaérien automobile AMX-30SA, l'ensemble de fusée antiaérien automobile AMX-30R, la rampe de lancement automobile de fusée "Pluton" et le canon automobile AMX-30GT.

Dès 1982, l'armée a commencé à être dotée d'un modèle modernisé du char qui a reçu le nom AMX-30B2. Contrairement à son prédécesseur, il est équipé, au lieu de la mitrailleuse de 12.7 millimètres, du canon mitrailleur de 20 millimètres, qui a aussi une transmission indépendante selon l'angle d'élévation. Le char AMX-30B2 est équipé du système moderne de commande de feu APX-M581. De sa composition font partie des viseurs télémètre à laser, un ordinateur balistique, un stabilisateur électrohydraulique d'armement, des instruments infrarouges de vision nocturne. Un nouvel obus sous-calibre perforant dont la force de pénétration à une dislance de 2000 m constitue près 350 mm selon la normale est incorporé à la dotation en munitions du canon. La mobilité du char modernisé est améliorée grâce à l'installation de la *m_AMX-32. On osnathen 120mm gladkostvolqnoj pushkoj, imeet nezavisimuyu stabilizaciyu polya zreniya panoramnogo pricela komandira, bolee sovershennyie priboryi nokhnogo videniya. Krome togo, blagodarya primeneniyu mnogoslojnogo bronirovaniya perednej khasti korpusa i bashni, a takzhe falqshbortov znakhitelqno povyishen urovenq zathihennosti

l'installalion de la transmission hydraulique et mécanique <"m_ENC-200>. les travaux selon le perfectionnement ultérieur du char <"m_AMX 30> on a créé un char de base <"m_AMX-32>. Il est équipé d'un canon à ame lisse de 120 millimètres, il a une stabilisation indépendante champs de la vue de la lunette panoramique du chef champs les instruments plus parfaits? perfectionnes (Genre?)? d'une vision nocturne. En outre grâce à l'utilisation du blindage multicouches de la partie avant de la carcasse et de la tour ainsi que des carènes considérablement on a augmenté le niveau de protection.

transmission hydraulique et mécanique Pendant ENC-200

Pendant les travaux sur te perfectionnement ultérieur du char AMX-30, on a créé le char de base AMX 32. Il est équipé d'un canon à âme lisse de 120 millimètres, il a une stabilisation indépendante du champ de vision de la lunette panoramique du chef, des instruments plus perfectionnés de vision nocture. En outre, grâce à l'utilisation du blindage multicouches de la partie avant de la carcasse et de la tour ainsi que des carènes, on a considérablement augmenté le niveau de protection.

## Improving the lingware: a short example

### *Source text and translation before correcting the dictionaries*

LANGUES DE TRAITEMENT: RUB - FRB

--( TRADUCTION DU 6 NOVEMBRE 1986 8H 40MN 41S ) ----

VERSIONS : ( A : 21/07/86 - T : 21/07/86 - G : 21/07/86 )

-TEXTE SOURCE-

Cifrovaya obrabotka signalov v optike i gologralii. Vvedenie v cifrovuyu optiku.

Izlagayutsya osnovyi naukhnogo napravlenîya, izukhayuthego ispolqzovanie cifrovyix processorov v optikheskix i golografikheskix sîstemax Rassmatrivayutsya voprosyi optimalqnogo cifrovogo predstavleniya i modelirovaniya optîkheskix signalov i îx preobrazovanij, yeffrktivnyie vyikhislitelqnyie algoritmyi i adaptivnyie metodyi obrabotki izobrazhenij, gologramm i interferogramm, sinteza gologramm i yelementov optikheskix sistem

**-TEXTE TRADUIT-**

Traitement numéral des signaux dans l'optique et la graphie nue. Introduction dans une optique numérale.

On expose les bases de la direction scientifique qui étudie l'utilisation de processeurs numéraux dans des systèmes optiques et nu (Genre-Nombre?) graphiques. On examine les problèmes de la représentation numérale optimale et du modelage de signaux opaques et de leurs transformations, algorithmes de calculateur efficaces et méthodes adaptables du traitement des représentations, des grammes nus et des interférogrammes, de la synthèse des grammes nus et des

Dlya naukhnyix rabotnikov, specializiruyuthixsya v oblasti informatiki, v khastnosti zanimayuthixsya obrabotkoj izobrazhenij, optikoj, golografiej i cifrovoj obrabotkoj signalov.

elements de systèmes optiques

Pour les chercheurs spécialisés dans le domaine de l'informatique, en particulier les représentations qui s'occupent au traitement, optique, graphie nue et le traitement numeral des signaux.

## *Translation after correcting the dictionaries and revision*

LANGUES DE TRAITEMENT: RUB - FRB

— (TRADUCTION DU 6 NOVEMBRE 1980 14H 27MN 22S ) —

VERSIONS : { A : 9/10/86 - T : 6/11/86 - G : 9/10/86 )

----(REVISION DU 6 NOVEMBRE 1986 14H29MN54S)------

-TEXTE TRADUIT-

Traitement numérique des signaux dans l'optique et l'holographie. Introduction à une optique numérique.

On expose les bases des axes de recherche scientifiques qui étudie l'utilisation de processeurs numériques dans des systèmes optiques et holographiques. On examine les problèmes de ta représentation numérique optimale et de la modélisation de signaux optiques et de leurs transformations, les algorithmes de calculateur efficaces et des méthodes adaptables du traitement des représentations, des hologrammes et des interferogrammes, de la synthèse des hologrammes et des éléments de systèmes optiques.

Pour les chercheurs spécialisés dans le domaine de l'informatique, en particulier les representations qui s'occupent au traitement, l'optique, l'holographie et un traitement numérique des signaux.

- TEXTE REVISE-

Traitement numérique des signaux en optique et en holographie. Introduction à l'optique numériques.

On expose les bases des axes scientifiques de recherche pour l'élude de l'utilisation de processeurs numériques dans des systèmes optiques et holographiques. On examine les problèmes de la représentation numérique optimale et de la modélisation des signaux optiques et de leurs transformations, les algorithmes de calcul efficaces et des méthodes adaptatives du traitement des représentations, des hologrammes et des interférograrnmes, de la synthèse des hologrammes et des éléments des systèmes optiques.

Destiné aux chercheurs spécialisés dans le domaine de l'informatique, en particulier à ceux qui s'occupent du traitement des représentations. de l'optique, de l'holographie et du traitement numérique des signaux.