[Proceedings of the *Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, August 14-16, 1985]

ON THE PRODUCTION ENVIRONMENT PROPOSED FOR THE EUROTRA PROJECT

D.BACHUT

Groupe d'Etudes pour la Traduction Automatique - Université Scientifique et Médicale de Grenoble
BP 68 - 38402 Saint Martin d'Hères FRANCE

Institut de Formation et Conseil en Informatique
27, **rue de Turenne - 38OOO Grenoble FRANCE**

I. ABSTRACT

We present the general architecture of a production environment which is specific for a M(A)T system, and give some proposals to Integrate new functionalities in this system. A good management of the results of the translation process may lead to an easier Improvement of the linguistic data.

We describe a possible organisation for the machine environment of such a system and for the management of the data base of texts. Finally, we give some general rules for the implementation of a monitor.

II. KEY WORDS

Production environment, translation process, local network, special purpose network, monitor,

III. INTRODUCTION

The main goal of the EUROTRA programme is the automation of the translation task. When "production" is mentioned, this means a wish to push on. It means that not only the translation as such should be automated, but also its whole environment.

(*) This work has been carried out as part of a contract with the Commission of the European Communities (in the framework of the EUROTRA Research and Development programme) and the CNRS (Centre national de la Recherche Scientifique). The ideas and proposals in this paper are those of the author and not necessarily shared or supported by the Commission, nor are they to be interpreted as part of the EUROTRA design. We are grateful to the Commission and the CNRS for agreement to publish this paper.

Our objective is, therefore, to create a computer environment encompassing all the subsidiary operations as well as the management of the texts submitted to translation and of the resulting texts. Examples of such operations are:

- the transcoding of the texts at input, and perhaps output, time;
- the "on-line" revision of the texts;
- the use of interchangeable supports (floppy disks) for "off line" revision operations;
- the use of external supports (such as magnetic tapes) for large corpuses.

The designer should take into account the classical problems of production environments: maintenance, multiplicity of versions, multiplicity of implementation sites...

In the particular case of M(A)T, the newest problem, which is also the most crucial one will be the maintenance of the linguistic models. Below, we try to present clearly its implications in the context of a translation sequence.

The essential constraint of a real production environment is the impossibility to modify the linguistic data, as modules are used, which are binary images of these data. These modifications are, however, feasible in parallel (if necessary on another site). When these versions, after sufficient testing, are considered operational, they can replace or be added to the modules which are active in the environment.

IV. GENERAL ARCHITECTURE OF A M(A)T SYSTEM

Let's first consider the functional organization of a typical M(A)T system.

```
+-------------+------------------------------------------------------------------------------------------------------+
!             !                                                                                                      !
!  TEXT       ! Entry ->- Input  ->- Exploration ->-Translation->-  Output   ->-  linguistic ->- technical ->-  output on      !
!             !         conversion !                                 Conversion    revision        revision        any support    !
+-------------+------------------------------------------------------------------------------------------------------+
!             !                    ^        v                                      v              v                 !
! LINGUISTIC  !                    !  Indexing of new                        Maintenance of   Dictionnaries         !
! DATA        !                    !  words, corrections and                 linguistic model maintenance           !
!             !                    !  compilation                            and, possibly,                         !
!             !                    !        v                                dictionaries                           !
!             !                    +--------+                                                                       !
+-------------+------------------------------------------------------------------------------------------------------+
```

Figure 5 : Interaction between translation sequence and linguistic data

The first line in this schema is the conventional translation sequence. The second represents the possible consequences at the level of the linguistic data.

1. PRIOR TO THE TRANSLATION PROCESS

The "Entry" and "Input conversion" operations (manual or automatic) have only one goal: to put any text into a form which is "readable" by the translation process. They have no influence whatsoever on the linguistic applications. But the text itself, with its content constituted by words which are perhaps unknown, must lead the linguist to enrich his vocabulary. This work can be done in two ways:

- either manually - the linguist looks up the words of the text in his dictionaries;
- or automatically, with the help of a program which does this research itself.

Except in the case of the very beginning of a linguistic application, only the second solution seems interesting and acceptable to us, all the more in a production framework.

This program (called here "Exploration") can be placed before or after the input conversion. As the dictionaries and the texts are not necessarily entered with the help of the same Input device, the first solution will certainly lead to incompatibilities in the transcription. Thanks to the unique character of the EUROTRA code we have proposed to use in the second case <1>, this problem can be avoided, and the exploration work can be carried out in a homogenous way.

Moreover, in order to recognize the unknown words, it is natural to use the morphological analysis of the M(A)T system. The text should therefore have the form of an input work structure.

The exploration must provide the user with all the unknown words (which can, in some cases, be different forms of the same LU). Except for those words which are intentionally outside the system, such as, for example, proper names, the new lexical units must be indexed and the mistaken words which the programme would have interpreted as unknown words must be corrected. However, the activation of this program should remain optional. Even with unknown words, a translation is not impossible; a linguistic strategy must be defined for their processing.

2. AFTER THE TRANSLATION

The "Output conversion" will not yield anything for the linguist.

The revision, on the contrary, will be for him an interesting source of error detection.

We distinguish two revisions which are, in fact, complementary:

1. a purely linguistic revision, aimed at the quality of the translated text. It concerns essentially work on the grammar with some lexical consequences;

2. a technical revision, aimed at the accurate translation of the terms proper to the subject field of the text. This work is, then, essentially lexical, although grammatical corrections are likely at this level.

3. COMPLEMENTARY TOOLS

We have thought it interesting to define several functionalities for the exploration process Introduced in the preceding section. Some of these functionalities have already been used in other M(A)T system.

- a system of automatic rectification of errors. With the help of very simple techniques, a significant number of errors can be rectified. Such a system is used in METAL <2>.

- the elaboration of temporary dictionaries, the access mode, the syntax and the semantics of which respect those of the current dictionaries of the linguistic applications written in the EUROTRA formalism.

  Their role would be to index the unknown words detected during the exploration and which are useless.

- a system of corpus exploration of the JEUDEMO <5> or DEREDEC <6> type used in the TAUM project <8>. Such a system should yield (in increasing order of complexity and, thus, of implementation):

1. statistical data and standard lists: occurrences, words, lexical units, concords, unknown words;
2. special lists: foreign language words, composed words, names, verbs...
3. complex lists: simple NPs, complex NPs (noun's complement...)...

V. GENERAL MACHINE ARCHITECTURE

We will give here a general description of the architectures which can be envisaged for a system which supports the production environment of the EUROTRA system. To that end, we distinguish between centralized and decentralized organizations on the one hand, and the characteristics of the work stations on the other hand, which can be specialized (special purpose machine) or general (access to all the available tools). These distinctions lead us to the table below, in which the difference between the project and the operational periods of EUROTRA is made clear.

The term "centralized" applies exclusively to the computer component of the EUROTRA system, because the linguistic component is shared between the various countries of the Community. This is one of the basic constraints of this project.

Machine -> Processing ↓	GENERAL	SPECIAL PURPOSE
CENTRALIZED	project	transition
NOT CENTRALIZED	transition	exploitation

When the system is really operational, it is better to decentralize and to specialize the tasks, for reasons of:

- safety: the volume of the linguistic applications, for example, will be huge and they will be very complex. Their access must be under strict control;

- volume: the size of the corpuses will also be more important and pre- and post-editing must be divided-up. It will be out of the question to keep all the corpuses and their corresponding translations on one machine only;

- efficiency: this reason follows from the preceding one. The volume of data to be processed will necessarily imply a specialization of the work: operator, translator revisor, system manager...

  It is obvious that most of the operations on texts are asynchronous and can be carried out in parallel. It seems, therefore, desirable, for efficiency reasons, to have different processors share this work. It would be a pity if the reading of a tape at input time would prohibit (or, in the case of a time-sharing/ multi-tasking system, delay) a translation. From this viewpoint, a decentralized environment is, therefore, a good solution.

This classification of architectures may seem rather simplistic, but in fact, shifting from one to the other may be carried out gradually. In the table above, the "transition" illustrates this shifting from one architecture to the other. This "transition" can be implemented either by a specialization of certain operations in the central work site, or by a decentralization, or by both simultaneously.

The proposed environment centers around a collection of "virtual machines" (users' spaces under UNIX, projects under MULTICS, intelligent terminals linked through local networks <7>...) with pre-established roles.

The structure of the virtual machine offers an automatic sequence of the tasks linked with the translation process. It takes charge of the means of communication between these machines, some of which are assigned to the actual translation process and the others to the so-called subsidiary operations.

Thus, the architecture adopted here appears as a "processing network", the components of which are special purpose machines for precisely defined tasks. In principle, a linear architecture would certainly have been sufficient for these relatively fixed operations, but the network structure has the advantage of enabling:

- the specialization of the virtual machines;
- a greater diversification of the processings;
- a greater flexibility in adaptation to the tasks to be carried out.

The synchronization of the various virtual machines is ensured by one "master machine" which manages these machines as resources for the sequencing of the desired processings.

Although the choice of a master machine is not an absolute necessity, it has nevertheless the advantage of enabling each virtual machine to communicate with a rapidly available partner and of centralizing the information concerning the progress of the tasks, allowing the search for a sequencing in function of the general organization.

The schema below illustrates a possible configuration of this network. ("Machine" includes proper core and mass memory; "telecom" means "telecommunications lines").

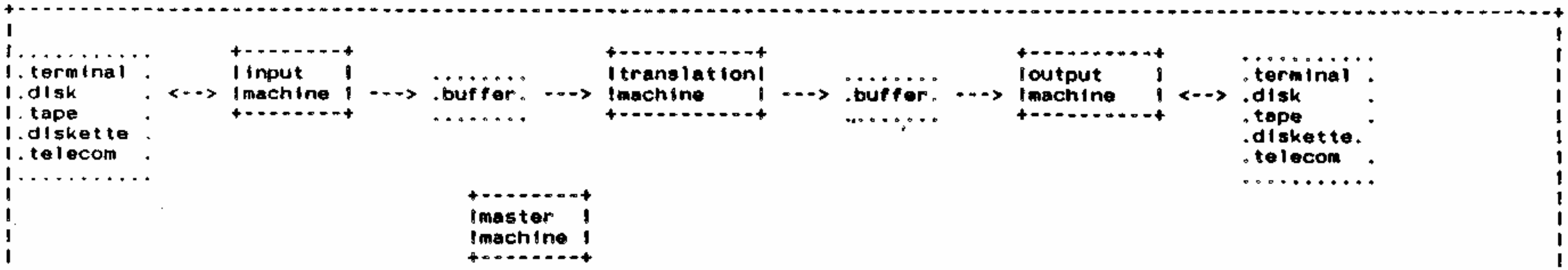

Figure 3 : Minimal configuration of the architecture

VI. CIRCULATION AND ORGANIZATION OF TEXTUAL DATA

The term "data" applies here to the texts submitted to translation or to those resulting from it. For the environment, a text constitutes a unit, from the processing point of view. A "corpus" is a finite and definite set of texts, homogenous both as to their nature and as to the possible processings in the context of the environment.

Thus, the user would have the possibility to start a process for all the texts in a corpus, and the environment would then take charge of the management of these texts, so as to ensure the desired processing (Including the entry, the input conversion, the translation, the revision, etc.).

A set of "input objects" would be associated with each text, which contains the datas to be processed, as well as a set of "output objects", containing the resulting data.

A particular object, called "bulletin", would be associated with each text and act as a descriptor of the sequencing of the processings planned for this text; so the bulletin accompanies the text all along this sequencing. It can be established directly by the user (by means of a component installed on an input machine, which carries out this operation interactively), or it can be derived from an analogous bulletin established for the whole corpus. It would be progressively completed as the various processings progress.

```
+----------------------------------------------------------------------+
| Text  : ............                                                 |
|                                                                      |
| Corpus :.........                                                    |
|                       Type of formatting  :  ..................      |
|                       External support    :  ..................      |
|                       Origin              :  .................       |
|                                                                      |
| Operator  : ........                                                 |
|                                                                      |
| Pre-editing :                                                        |
|                       Input conversion   :  ..................       |
|                       Entry machine       : ..................       |
| Translation :                                                        |
|                       Pair of languages   : ...................      |
|                       Output of results   :  .................       |
|                                                                      |
| Post-editing :                                                       |
|                       Type of formatting  : ..................       |
|                       Revision machine    : ................         |
|                                                                      |
| Statistics     : ..............                                      |
|                                                                      |
| Storage      : ..............                                        |
|                                                                      |
| Addressee : ...............                                          |
|                       External support    : ...................      |
+----------------------------------------------------------------------+
```

Figure 4 : Example of a bulletin

VII. MONITOR

The role of the environment is to ensure the adequate flow of data between processings carried out in parallel, taking into consideration the requests of the users, the available resources and the constraints imposed by the implemantation of this environment.

It is possible to ensure this complex management with the help of a monitor <3>, which, moreover, will offer a conversational interface for users who are (computational) linguists. Through this channel, they can communicate their wishes (in the form of parameters), thanks to which it will be able to manage the sequencing of actions.

Generally speaking, the basic principles of such a monitor should be:

- to facilitate the work of the user;
- to handle "all" the cases of error;
- to be able to communicate in different natural dialog languages;
- to ensure the coherence of the data base stored in the system and to manage it in way which is transparent for the user, who does not have to bother about calling and loading the "right" programme at the "right" moment.

The programming of such a monitor should be done with a specialized language, which offers flexiblity in the implementation. A command language should also be forseen to facilitate the work of the linguist.

VIII. CONCLUSION

The problem of the production environment is not the most important one for M(A)T system. Generally a solution to such problem is studied once the M(A)T system is running, and is often not adapted.

Studying this problem at the beginning leads to know the possible impacts on the functional requirements of the software system.

The proposal made here is general and give a framework for the choice of a given solution.

IX. BIBLIOGRAPHY

1. BOITET Ch., BACHUT D., VERASTEGUI N., "Various representation of text proposed for EUROTRA", ACL, Geneve, March 1984.

2. BENNETT W., SLOCUM J., "METAL: The LRC Machine Translation System", Linguistic research center, Austin, texas, USA, September 1984.

3. BOITET C., GUILLAUME P., QUEZEL-AMBRUNAZ M., "Implementation and conversational environment of ARIANE-78. An Integrated system for automated translation and human revision". Proceedings COLING82, North-Holland, Linguistic Series No 47, P 19-27, Prague, duly 82.

4. BOITET C., NEDOBEJKINE N., "Illustration sur le développement d'un atelier de traduction automatisée", Colloque "l'informatique au service de la linguistique", Université de METZ, France, Juin 1983.

5. OUELLETTE F., "JEUDEMO ou les textes «mot à mot»", Bulletin 896-01, Centre de calcul, Université de Montréal, Canada, Janvier 1979.

6. PLANTE P., "DEREDEC, manuel de l'usager", Service de l'informatique, Université du Québec, Montréal, Canada, Octobre 1980.

7. STALLINGS W., "Local Networks", ACM Computing Survey, pp 3-41, Vol 16 Number 1, March 1984.

8. TAUM, "TAUM-METEO, Description du Système". Groupe de recherches pour la Traduction Automatique, Université de Montréal, 47 p., janvier 1978.