An Interactive Hypertextual Environment for MT Training

Etienne Blanc

GETA, CLIPS-IMAG
BP 53, F-38041 Grenoble cedex 09
Etienne.Blanc@imag.fr

Abstract

An interactive hypertextual environment for MT training is described. It combines the ARIANE MT system with an hypertextual control interface implemented on the learner's personal computer, and communicating with the ARIANE server through e-mail.

Introduction

The main tool used for MT Research at GETA (Groupe d'Etude pour la Traduction Automatique) is the ARIANE-G5 MT shell, a generator of MT systems. Using ARIANE-G5, one may develop MT systems for any pair of languages, using specialized languages for linguistic programming developed at Geta (Boitet, 1997).

The flexibility of the ARIANE-G5 shell and of the methodology developed at GETA made it possible to undertake or to take part in various multilingual projects , as the LIDIA multilingual project of Dialogue Based Machine Translation (Blanchon, 1994), the 15-language Universal Networking Language project initiated by the Institute of Advanced Studies of the United Nations University (http://www.ias.unu.edu) , or the CSTAR project of multilingual speech translation (http://www.is.cs.cmu.edu/cstar).

Numerous trainees, were involved in these projects, and had to be acquainted with the MT technique,. An interactive hypertextual environment for the control of ARIANE-G5, CASH (for Control of Ariane Supported by Hypercard), initially developed as an tool for the MT developer (Blanc & Guillaume, 1997 ; Blanc, 1999) proved very useful for the training of beginners in the MT field.

The fact that ARIANE-G5, although particularly well suited for the transfer approach, doesn't impose it, and that the CASH interface runs on a small personal computer (presently only an Apple Macintosh one), and controls ARIANE through e-mail messages, offers a large range of training applications

Before describing the use of the CASH interface as a training tool, a brief presentation of the ARIANE generator of MT systems will be useful.

ARIANE-G5, a Generator of MT Systems

Fig.1 shows ARIANE-G5 when used for generating a classical bilingual transfer MT system. Each of the main steps (analysis, transfer, generation) is achieved by several lexical and structural phases, each written by the linguist using specialized languages for linguistic programming. Not all the phases are compulsory. An elementary training transfer MT system will only include two phases for each step: morphological and structural analysis, lexical and structural transfer, structural and morphological generation.

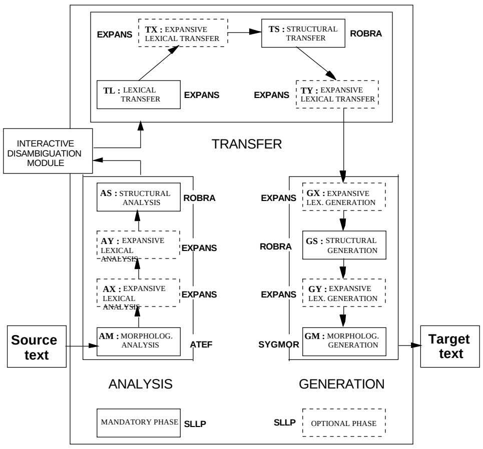

EXPANS
TX : EXPANSIVE LEXICAL TRANSFER
TS : STRUCTURAL TRANSFER
ROBRA
TL : LEXICAL TRANSFER
EXPANS
EXPANS
TY : EXPANSIVE LEXICAL TRANSFER
INTERACTIVE DISAMBIGUATION MODULE
TRANSFER
AS : STRUCTURAL ANALYSIS
ROBRA
EXPANS
GX : EXPANSIVE LEX. GENERATION
AY : EXPANSIVE LEXICAL ANALYSIS
EXPANS
ROBRA
GS : STRUCTURAL GENERATION
AX : EXPANSIVE LEXICAL ANALYSIS
EXPANS
EXPANS
GY : EXPANSIVE LEX. GENERATION
Source text
AM : MORPHOLOG. ANALYSIS
ATEF
SYGMOR
GM : MORPHOLOG. GENERATION
Target text
ANALYSIS
GENERATION
MANDATORY PHASE
SLLP
SLLP
OPTIONAL PHASE

Figure 1 : the Ariane-G5 system as used for generating a transfer MT system.

When using GETA's methodology , the unit of translation (which is not restricted to a sentence, but may include several paragraphs) is processed as a decorated "multilevel" tree.

The name "multilevel tree" means that the complete representation of a sentence, as obtained after the analysis phase, is a tree bearing the three levels of information : syntagmatic, syntactic and logico-semantic, the geometry of the tree being the syntagmatic one.

For the transfer from the source to the target language, only the deep level logico-semantic information of the multilevel tree supplied by the analysis step is normally used to build the multilevel tree in the target language (the other levels being available as a « safety net »).

But, as mentioned before, ARIANE-G5 is not devoted to an unique MT methodology. The only strong constraint is that the structures representing the unit of translation must be decorated trees.

Fig.2 shows an example of how ARIANE is used in the UNL project. In the UNL pivot language, a unit of meaning is represented as a hypergraph. (A hypergraph is a graph were nodes themselves may be graphs). A transfer module external to ARIANE gives an equivalent representation in the form of a dependency tree, which is the input of the transfer and generation ARIANE processing.

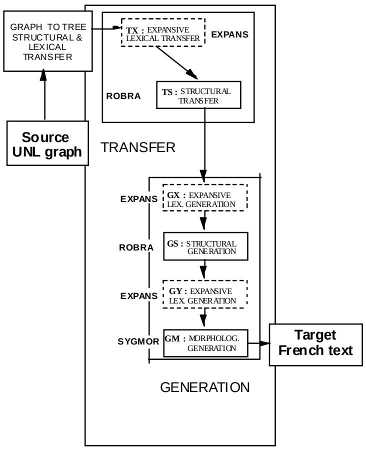

GRAPH TO TREE STRUCTURAL & LEXICAL TRANSFER
TX : EXPANSIVE LEXICAL TRANSFER
EXPANS
ROBRA
TS : STRUCTURAL TRANSFER
Source UNL graph
TRANSFER
EXPANS
GX : EXPANSIVE LEX. GENERATION
ROBRA
GS : STRUCTURAL GENERATION
EXPANS
GY : EXPANSIVE LEX. GENERATION
SYGMOR
GM : MORPHOLOG. GENERATION
Target French text
GENERATION

Figure 2 : The ARIANE-G5 system as used for generating an UNL deconverter.

We will restrict this presentation to an insight into the use of the CASH control interface as a training tool in usual tranfer MT.

Principle of the CASH Interface

The basic principle of the CASH control interface is that the user's personal computer stores hypertextual copies of the lingware implemented in the mainframe, and that file exchanges with ARIANE through the e-mail offer then an almost complete control of ARIANE (fig.3).

The user may for instance :

- edit and send a text for processing (complete translation, or partial processing ; plain result or trace through the various processing steps)

- receive lingware modules from ARIANE for updating his copies

- modify or create lingware modules on his personal computer and send them for implementation in the ARIANE mainframe.

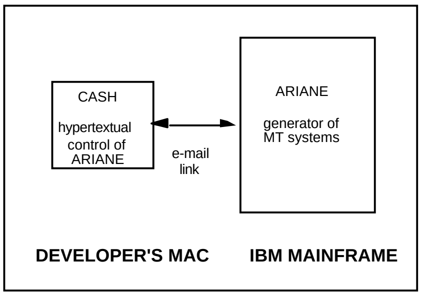


Figure 3 : Basic principle of the CASH interface.

A given MT system for a given pair of language may be shared between several users, or each user may have his own system on his virtual machine on the mainframe.

All this makes ARIANE, used through the CASH interface, a very powerful and flexible tool for MT training, from the very first introductory steps to high level training, for a wide and open rangeof languages. Let's take as a simple example an introduction to the structural analysis of English.

Tracing a Structural Analysis

Fig.4 shows the main control screen of the CASH interface. By clicking on this screen, the user defines the various parameters of the command to be executed: his virtual machine on the ARIANE mainframe (here the machine named « DEMOCASH »), the MT system (here the English to French BA5-FAC MT system), the processing step and the lingware module of this step to be examined or edited (here a grammar of the analysis step), the processing chain (defining the parameters to be used for the processing of a text as will be explained below), and the text to be processed.

The main commands are listed in the « Commandes » item of the menu bar. For instance clicking on the « execut » command will result on the processing of the selected text with the selected parameters (the selection of the AS phase means that the processing will stop after the structural analysis step). The « execut/ctrl » will supply the trace of the processing through a given step.

Fig.5 shows a fragment of the trace of the structural analysis of the English sentence « This little shy boy misses his mother » as received from ARIANE after sending the « execut/ctrl » command with the chosen parameters. This fragment shows the integration of an adjectival phrase into a noun phrase.

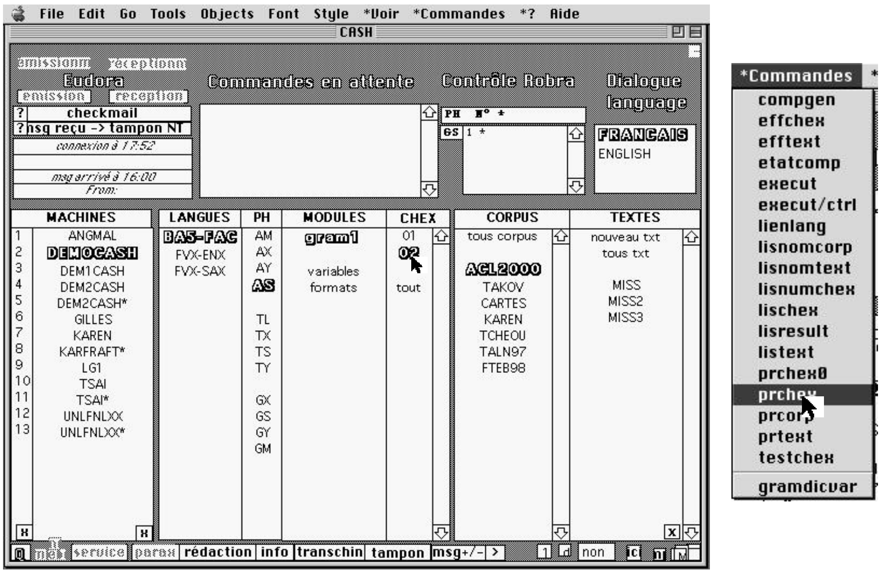


Figure 4 : The main control screen of CASH and the item « Commandes » of the menubar.

As may be seen on this figure, the trees commonly used in the ARIANE processing are very close to dependency trees, the main difference being that each node is splitted into a couple, the “syntagmatic” node (e.g. NP for noun phrase, AP for adjectival phrase), bearing the information relative to the group, and the “governor” node, bearing the information relative to the word itself. But, as already mentioned, other tree structures, as pure dependency trees, could as well be used.

Browsing in the structural analysis module until accessing the rule for the integration of an adjectival phrase into a noun phrase is achieved as following.

A structural analysis (or generation) module comprises one or several « transformational systems ». The structural analysis module of the English to French « BA5-FAC » MT system comprises only one such system, named « gram1 » on the screen of figure 4.

Selecting « gram1 » on the screen of figure 4 and activating an item of the « voir » menu gives access to the « control graph » of the transformational system (upper window of fig 6). This control graph is a graph where each node bears the name of a transformational grammar (eg « DISAMB6 », «JUXTN », « NOUNP1 » on fig.6) or the exit symbol &NUL, and the edges bear tree conditions.

Clicking on the name of a given transformational grammar (« NOUNP1 » for example) of the upper window of fig.6 gives access to the second window of this figure, where the the rules of the grammar are listed (« DINGNP », « APNP »…)

Clicking on the name of the rule gives access to a third window where this rule is described. The lower left part of the window shows the structure an input subtree should have to be processed by the rule ; the lower right part shows the structure of the subtree after application of the rule ; the upper left part describes the detailed rule as written in the ROBRA specialized language for linguistic processing (clicking on a symbol like $CP in the rule opens another window explaining the symbol); the upper right part may contain a comment to the rule, a comment which is local to the user’s CASH interface.

There are indeed two kinds of comments. The first one is a classical one, and is an integral part of the lingware, shared by all its users. The second one is local to the CASH interface : the user may comment each object of each level of the module (the control graph, each transformational grammar of the control graph, each rule of a transformational grammar). These local comments are not altered when the module is updated by receiving a new version from the ARIANE mainframe (except of course if the concerned object has disappeared in the new version). Such personal comments are very useful for MT learners.

The Tree Editor

The whole of the processing steps of a MT system may be studied as shown in the preceding section. It is also possible to examine how a given processing step works, independently from any preceding ones. This is achieved by inputting an appropriate tree at the entry of that processing step. Such a tree is created by using the tree editor of figure 7.

The tree shown on this figure is an entry tree for a generator of English. The structure appears in the upper field, the node decoration in the third field. After processing, the result appears in the bottom

Conclusion

The use of ARIANE associated to the CASH interface offers thus a very flexible tool for MT training :

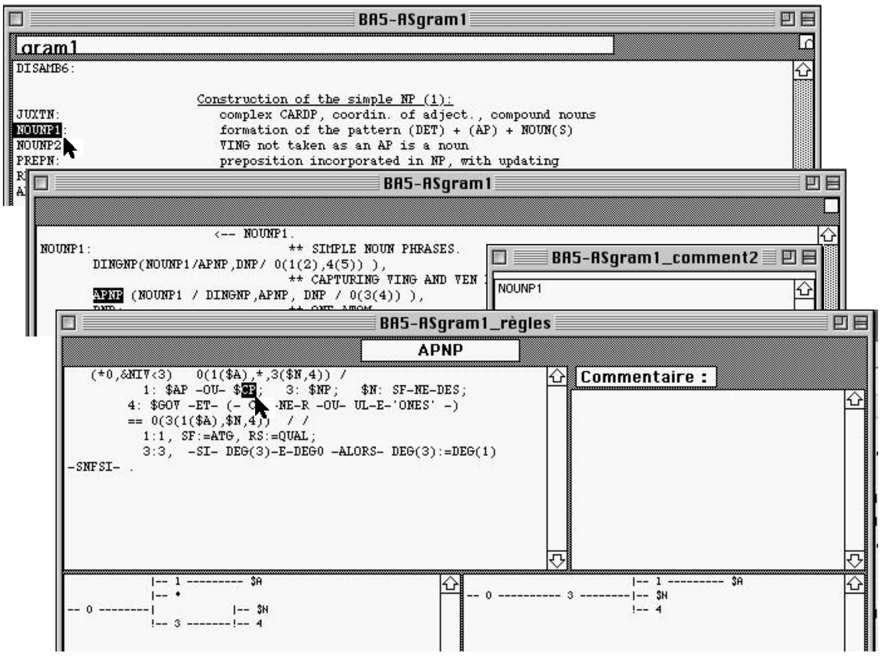
```
Input tree :
                              |-- 3:*SEG
                              |-- 4:ULOCC ----- 5:THIS
                              |-- 6:*AP ------- 7:LITTLE
                              |-- 8:*AP ------- 9:SHY
                              |-- 10:*NP ------ 11:BOY
-- 1:ULTXT ------ 2:ULFRA ----|-- 12:*VCL ----- 13:MISS ------ 14:MISS
                              |-- 15:ULOCC ---- 16:HIS
                              |-- 17:*NP ------ 18:MOTHER
                              |-- 19:.
                              !-- 20:*SEG

       8 '': UL('*AP'),K(AP),CAT(A),SUBA(ADJ).
         9 'SHY': UL('SHY'),SF(GOV),CAT(A),SUBA(ADJ).
      10'': UL('*NP'),K(NP),CAT(N),NUM(SIN),SEM(ANIM),SEMAN(HUMAN), VL1(N).
         11 'BOY': UL('BOY'),SF(GOV),CAT(N),NUM(SIN),SEM(ANIM),SEMAN(HUMAN).

Output tree :
                              |-- 3:*SEG
                              |-- 4:ULOCC ----- 5:THIS
                              |-- 6:*AP ------- 7:LITTLE
                              |              |-- 9:*AP ------- 10:SHY
                              |-- 8:*NP -----!-- 11:BOY
                              |
-- 1:ULTXT ------ 2:ULFRA ----|-- 12:*VCL ----- 13:MISS ------ 14:MISS
                              |              |-- 16:HIS
                              |-- 15:*NP ----!-- 17:MOTHER
                              |
                              |-- 18:.
                              !-- 19:*SEG

         8 '': UL('*NP'),K(NP),CAT(N),NUM(SIN),SEM(ANIM),SEMAN(HUMAN), VL1(N).
            9 '': UL('*AP'),RS(QUAL),K(AP),SF(ATG),CAT(A),SUBA(ADJ).
               10 'SHY': UL('SHY'),SF(GOV),CAT(A),SUBA(ADJ).
           11 'BOY': UL('BOY'),SF(GOV),CAT(N),NUM(SIN),SEM(ANIM),SEMAN(HUMAN).
```

Figure 5 : Fragment of the trace of a structural analysis : integration of an adjectival phrase into a noun phrase.



```
DISAMB6:

                        Construction of the simple NP (1):
JUXTN:                     complex CARDP, coordin. of adject., compound nouns
NOUNP1:                    formation of the pattern (DET) + (AP) + NOUN(S)
NOUNP2                     VING not taken as an AP is a noun
PREPN:                     preposition incorporated in NP, with updating
R
A
```

```
                <-- NOUNP1.
NOUNP1:                       ** SIMPLE NOUN PHRASES.
       DINGNP(NOUNP1/APNP,DNP/ 0(1(2),4(5)) ),
                              ** CAPTURING VING AND VEN
       APNP (NOUNP1 / DINGNP,APNP, DNP / 0(3(4)) ),
```

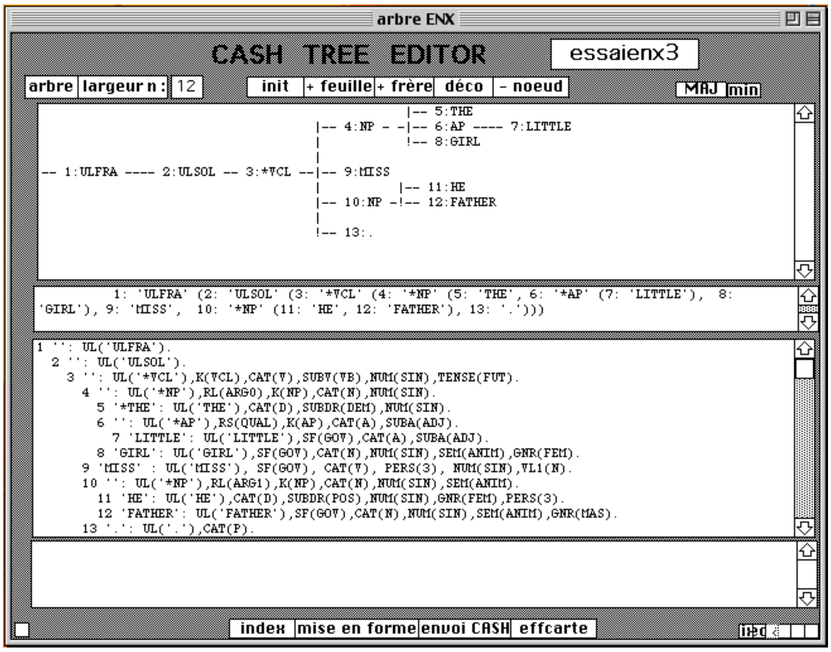
```
NOUNP1
```

```
   (*0,&NIV<3)    0(1($A),*,3($N,4)) /
         1: $AP -OU- $CP;    3: $NP;    $N: SF-NE-DES;
       4: $GOV -ET- (- C  -NE-R -OU- UL-E-'ONES' -)
       == 0(3(1($A),$N,4))  / /
         1:1, SF:=ATG, RS:=QUAL;
         3:3,  -SI- DEG(3)-E-DEG0 -ALORS- DEG(3):=DEG(1)
-SNFSI- .
```

```
            |-- 1 --------- $A
            |-- *
-- 0 -------|          |-- $N
            !-- 3 -------!-- 4
```

```
                        |-- 1 --------- $A
-- 0 ---------- 3 --------|-- $N
                        !-- 4
```

Figure 6 : Browsing through a structural analysis module : the rule achieving the integration of an adjectival phrase into a noun phrase



```
                                                |-- 5:THE
                                    |-- 4:NP - -|-- 6:AP ---- 7:LITTLE
                                    |           !-- 8:GIRL
                                    |
-- 1:ULFRA ---- 2:ULSOL -- 3:*VCL --|-- 9:MISS
                                    |          |-- 11:HE
                                    |-- 10:NP -!-- 12:FATHER
                                    |
                                    !-- 13:.
```

```
          1: 'ULFRA' (2: 'ULSOL' (3: '*VCL' (4: '*NP' (5: 'THE', 6: '*AP' (7: 'LITTLE'),  8:
'GIRL'), 9: 'MISS',  10: '*NP' (11: 'HE', 12: 'FATHER'), 13: '.')))
```

```
1 '': UL('ULFRA').
  2 '': UL('ULSOL').
    3 '': UL('*VCL'),K(VCL),CAT(V),SUBV(VB),NUM(SIN),TENSE(FUT).
      4 '': UL('*NP'),RL(ARG0),K(NP),CAT(N),NUM(SIN).
        5 '*THE': UL('THE'),CAT(D),SUBDR(DEM),NUM(SIN).
        6 '': UL('*AP'),RS(QUAL),K(AP),CAT(A),SUBA(ADJ).
          7 'LITTLE': UL('LITTLE'),SF(GOV),CAT(A),SUBA(ADJ).
        8 'GIRL': UL('GIRL'),SF(GOV),CAT(N),NUM(SIN),SEM(ANIM),GNR(FEM).
      9 'MISS' : UL('MISS'), SF(GOV), CAT(V), PERS(3), NUM(SIN),VL1(N).
      10 '': UL('*NP'),RL(ARG1),K(NP),CAT(N),NUM(SIN),SEM(ANIM).
        11 'HE': UL('HE'),CAT(D),SUBDR(POS),NUM(SIN),GNR(FEM),PERS(3).
        12 'FATHER': UL('FATHER'),SF(GOV),CAT(N),NUM(SIN),SEM(ANIM),GNR(MAS).
      13 '.': UL('.'),CAT(P).
```

Figure 7 : Editing an input tree for a generator of English

- the different steps of the MT processing may be studied in detail and independently.

- one is not limited to a given MT methodology (but decorated trees as support of linguistic information, and Geta’s languages for linguistic programming have to be used).

- analysers and generators are presently available for various languages.

A present limitation is due to the writing of the present version using Hypercard®, which imposes the use of a Macintosh personal computer.

References

- Blanc E, Guillaume P (1997), Developing lingware through internet : Ariane and the Cash interface. PACLING' 97, Meisei University, Ohme, Japan, sept 97, Proceedings 15-22.
- Blanc E (1999) An interactive hypertextual environment for MT development. Communications of COLIPS 9 (1) :67-81
- Blanchon,H (1994) Perspectives of DBMT for monolingual authors on the basis of LIDIA-I, an implemented mock-up. 15th International Conference on Computational Linguistics, COLING-94, Kyoto, August 5-9,1994.
- Boitet C. (1997) GETA' s methodology and its current developments PACLING' 97, Meisei University, Ohme, Japan, sept 97, Proceedings 23-57.