TWO STEPS TOWARDS SELF-EXPLAINING DOCUMENTS

Hervé Blanchon[1], Christian Boitet[2]

Abstract *— In the LIDIA project, we have demonstrated that, in many situations, Dialogue-Based MT is likely to offer better solutions to multitarget translation needs than machine aids to translators or batch MT, even if controlled languages are used. First experiments have shown the need to keep a memory of the "author's intention" by means of "disambiguating annotations" transforming the source document into a "self-explaining document" (SED). We present ways to integrate these annotations into an arbitrary XML document (SED-XML), and to make them visible and usable to users for better understanding of the "true content" of a document. The very concept of SED might deeply change our way of understanding important or difficult written material. We also show that an SED, once translated into a target language L, might be transformed into an SED in L with no human interaction, if an analyzer and a disambiguator are available for L. Hence, the SED structure might be used in multilingual as well as in monolingual contexts, without addition of human work.*

Index Terms — interactive disambiguation, self-explaining document, active document, XML-based disambiguating representation

INTRODUCTION

In many situations, documents such as working notes, scientific abstracts, slides, calls for proposals, technical documentation, etc., should be translated into several languages. They are not translated, because they are ready at the last moment, and available translators have no time to do the job, or because there are simply no translators to do the job, and of course, in all cases, because no satisfactory MT solution is available.

Our first point is that interactive Dialogue-Based MT systems (DBMT), especially of the kind we have prototyped in the LIDIA project [4], offer a better hope to solve the problem than machine aids for translators and "black box" MT, even if controlled languages are used.

Our second point is that the DBMT approach also leads to a new and extremely interesting possibility, that of producing all versions of a document, that is, the source document and all its translations, as "self-explaining" documents [3]. Such a document consist of a normal document and its deep or (even better) multilevel disambiguated linguistic representation, augmented by a memory of the original ambiguities and of the disambiguation process.

Finally, we observe that the production of self-explaining documents might also be very useful in monolingual contexts, and perhaps lead to new ways of accessing and using documents of any kind: one could "click" on any part marked as ambiguous, and get clarifying presentations or paraphrases of it. Thus, an unrestricted self-explaining text would be less ambiguous than a text in a controlled language, which may be unambiguous for a machine, but not for a human, and access to texts written in foreign languages would also be facilitated. In this way, authors' true intentions would accompany their productions in other places, times and tongues.

In this article, we will first introduce the LIDIA project and its first implementation (LIDIA 1). Section 3 is dedicated to the presentation of the SED concept and the LIDIA-2 implementation. In Section 4 we will introduce our first SED visualizer and give some short-term improvement. The 5th Section is dedicated to some more research-oriented following steps.

LIDIA-1: INTERACTIVE DISAMBIGUATION FOR DBMT

The LIDIA project

Past efforts towards raising the quality of MT output have demonstrated that FAHQMT (Fully Automatic High Quality Machine Translation) is possible, but only for restricted typologies of texts (domain, style) such as weather bulletins (METEO, TAUM, English↔French), stock market flash reports (ALT/Flash, NTT, Japanese→English), or technical documents (BV/aéro/FE for airplane maintenance manuals, Systran for Xerox documents in controlled English), etc.

After having worked in this direction of "suboptimization" for 15 years, we turned to high quality Dialogue-Based Machine Translation. DBMT is a new paradigm for translation situations where other approaches, such as the Linguistic-Based (LBMT) and the Knowledge-Based (KBMT) approaches, are not adequate. In DBMT, although the linguistic knowledge sources are still crucial, and extralinguistic knowledge might be used if available, emphasis is on indirect pre-editing through a negotiation and a clarification dialogue with the author in order to get high quality translations without revision. Authors are distinguished from "spontaneous" writers or speakers by the fact that they want to produce a "clean" final message and may be willing to enter into such dialogues.

[1] Hervé Blanchon, CLIPS-GETA, BP 53, F-38041 Grenoble Cedex 9, herve.blanchon@imag.fr
[2] Christian Boitet, CLIPS-GETA, BP 53, F-38041 Grenoble Cedex 9, Christian.boitet@imag.fr

In the first phase the typical translational situation considered is the production of multilingual technical documentation in the form of HyperCard stacks. Notable points in the linguistic design include multilevel transfer with interlingual acceptions, properties and relations. The first mockup, LIDIA-1, demonstrates the idea on a HyperCard stack, presenting short ambiguous French sentences in context. This stack is translated into three stacks, German, Russian and English. Although this mockup does not implement all features of the general design, because a complete implementation would have called for considerably more human resources than were available, we feel it demonstrates the potential of the approach and is a first step towards a usable prototype, where the linguistic engineering aspects and the reactions of real users could be studied.

The understandability of the question asked to the user has been evaluated. The results are available in [2].

Some aspects of LIDIA-1's GUI

The user can trigger the most frequent treatments by using the LIDIA-1 palette. The first line contains the LIDIA tools (process the selected object, show the treatment progress, show the annotations and show the reverse translation), and the second line the most frequent browsing tools.

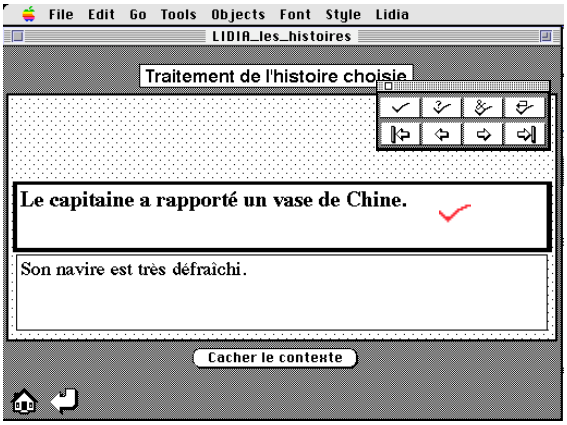

FIGURE 1
SELECTION OF AN ITEM TO BE TRANSLATED[3]

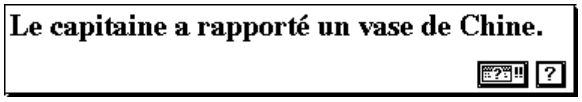

FIGURE 2
QUESTIONS ARE PENDING FOR THE TEXT

After analysis, the sentence may have to be disambiguated. A new button appears over the concerned object as in Figure 2. The user can choose to interact at once or later.

Suppose the user clicks on the button. A first question appears (Figure 3). In the context of this story, the user should choose to attach 'de Chine' to 'vase' (Chinese vase). A second dialogue appears (Figure 4) to ask about the word sense of 'capitaine'.

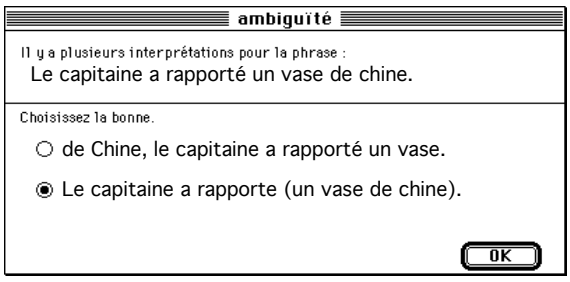

FIGURE 3
STRUCTURAL DISAMBIGUATION[4]

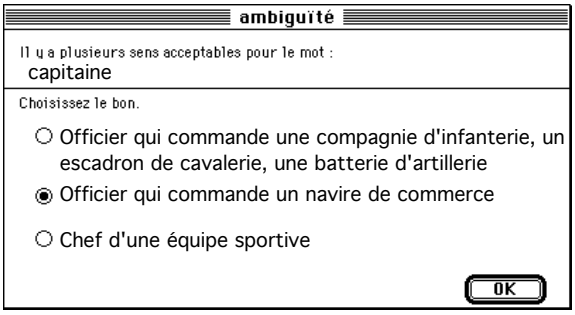

FIGURE 4
WORD SENSE DISAMBIGUATION FOR "CAPITAINE"[5]

Finally, the system produces the corresponding target language fragment, as shown in Figure 5.

FIGURE 5
FINAL GERMAN TRANSLATION FOR THE ORIGINAL SENTENCE IN TWO DIFFERENT CONTEXTS

[3] The chosen sentence translates as "*the captain brought back a vase from China*".

[4] The first paraphrase translates as "from China, the captain brought back a vase", the second one translates as "the captain brought back (a vase from China)".

[5] The first word sense proposed for "*capitaine*" is related to the military field, the second one to the shipping field, and the third one to the team field.

SED production workflow

We first proposed and motivated the concept of Self-Explaining in [3], let us give a brief account of the processes and data structures involved.

The source language text is analyzed and a source mmc-structure (Multisolution, Multilevel[6] and Concrete[7]) is produced. This mmc-structure is used to produce a question tree that will be displayed to the user. After interactive disambiguation, it becomes an unambiguous source umc-structure (Unisolution, Multilevel and Concrete) corresponding to the analysis chosen by the author. The source umc-structure is then abstracted to a source uma-structure (Unisolution, Multilevel and Abstract[2]).

The system then produces the *target gma-structures* (Generating, Multilevel and Abstract), using adequate transfer components. A gma-structure is in a way more "general" and "generative" than a uma-structure, because its surface-oriented levels (syntactic functions, syntagmatic categories…) may be empty, and if not are only preferences indicated by the transfer.

Paraphrase selection produces a *target uma-structure* homogenous with what would be the result of analyzing (and disambiguating) the target text to be generated. The translation process ends with syntactic generation and morphological generation.

During the translation (or analysis) the information to produce SED documents are kept. Figure 6 shows a functional diagram of this process.

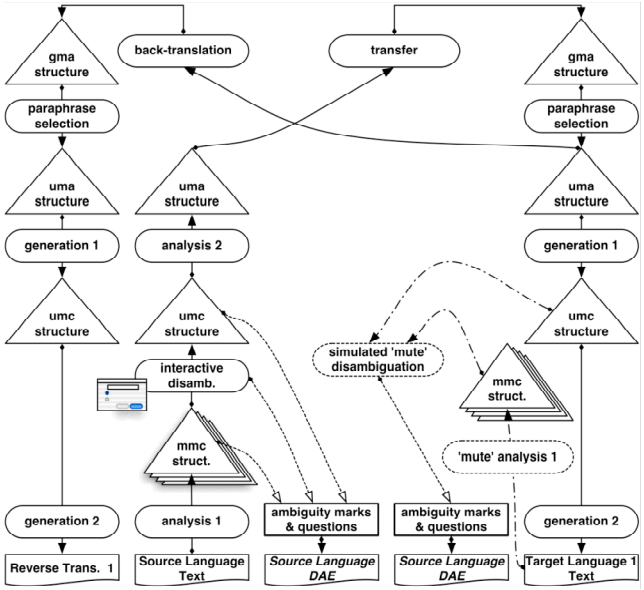

FIGURE 6
THE SED PRODUCTION ARCHITECTURE

[6] The structure consists of three levels of linguistic interpretation: the level of syntactic and syntagmatic classes, the level of syntactic functions and the level of logic and semantic relations.

[7] A "concrete" representation of a text is such that the corresponding text can be recovered from it by using a standard traversal algorithm. Otherwise we say that the representation is "abstract".

LIDIA-2 : PRODUCTION OF A SED (STEP 1)

In LIDIA-2, we first changed considerably the DBMT software architecture, in terms of communications and formats. In particular, we now produce a "mirror file" containing the history of (human) interactive disambiguation. Then, we developed a filter to produce the corresponding SED. Both files are produced in XML.
In this very first mockup we are using a disambiguation module that has been designed for English [1] based on a study reported in [6].

A simple session

The user first personalizes his environment. He can then create a new document or open an already existing one. The document window is divided into two sections: the editing window itself is located in the upper part and statistics about the current document status are given in the lower part.

After the user has requested the analysis (Figure 7), the ambiguous sentences are displayed in brown, and the unambiguous ones in green. In our example, the text contains 7 ambiguous sentences and 1 unambiguous sentence (the first).

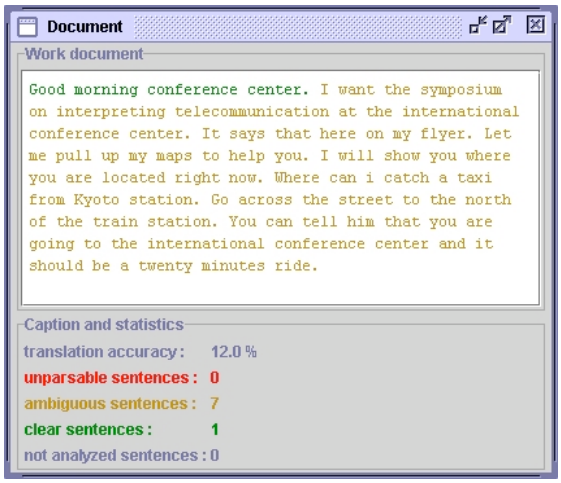

FIGURE 7
LIDIA-2 DOCUMENT WINDOW (AFTER PARSING)

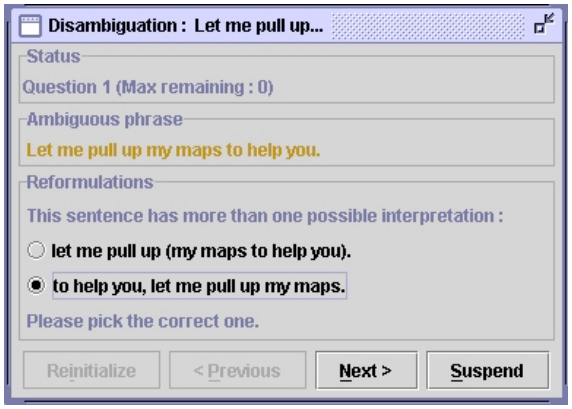

FIGURE 8
A DISAMBIGUATION QUESTION IN LIDIA-2

When the user double-clicks on some ambiguous sentence, the questions relative to this sentence are asked. Figure 8 shows the first (and here unique) question for the sentence “let me pull up my maps to help you”. The first proposed interpretation stands for “let me pull up my maps that have been designed to help you”, the second stands for “let me pull up my maps in order to help you”.

At that point, the sentence turns green, and the user can ask for the text to be translated into the available target languages. The translated text(s) may then be displayed side by side with the original text.

Document file

We chose the DOM API to handle the produced documents, and the SAX API to check the syntactic well-formedness of the documents to be opened.

The document contains a header, and its actual content (`support`). The description is made of a title, information about the author. The support is a set of paragraphs, each one being made of a set of sentences.

Each sentence has a source language and a unique transaction identifier that allows the environment to keep track of the ongoing treatments for each of them. The original content of the sentence, the answered question tree, and the produced translations are represented. As far as the question tree is concerned, it stores the answer path along the different reformulations and the umc-structure with its solution number associated with each terminal question.

Filtering to a SED

After disambiguation a source SED can be filtered out from the document. The sentences and the answered disambiguation tree without the umc-structure are kept.

VISUALIZATION OF A SED (STEP 2)

Our idea of a SED is that it should be a stand-alone document readable through a SED visualizer.

Goals and constraints

A SED visualizer should present a document and highlight its ambiguous segments. The reader should then be able to choose any ambiguous segment and exhibit its meaning when it seems necessary.

The SED visualizer we present in this article is fairly simple and the interaction with the document is still poor. As we want a SED to be usable locally and through the web, the SED visualizer is implemented as a Java application.

DOM is again is used to produce the GUI.

Screens from the present SED vizualizer

The user can open a SED document through the visualizer as shown in Figure 9.

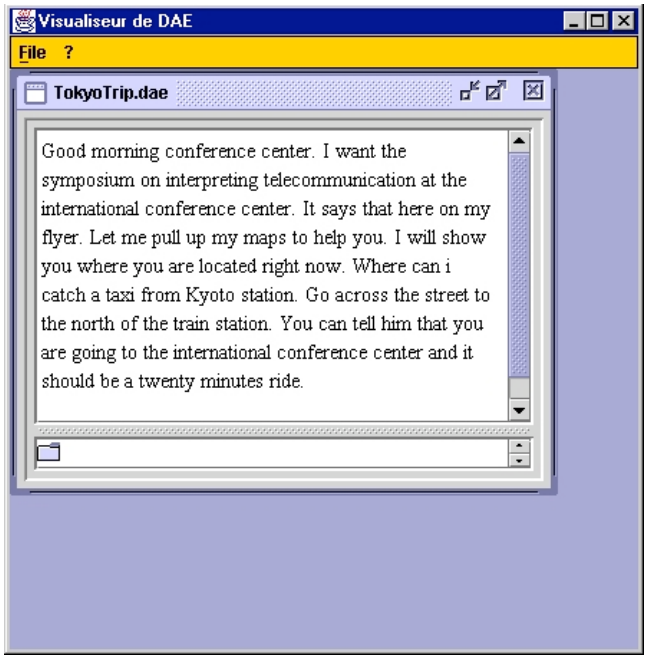

FIGURE 9
THE SED READER ENVIRONMENT

At this point, the ambiguous segments are not highlighted (see 5.2 below). To gather information about the different readings the reader has to double-click on a sentence. A dialogue box appears that allows the reader to browse through the questions answered by the writer of the document. While browsing, the rephrasing chose by the writer is highlighted.

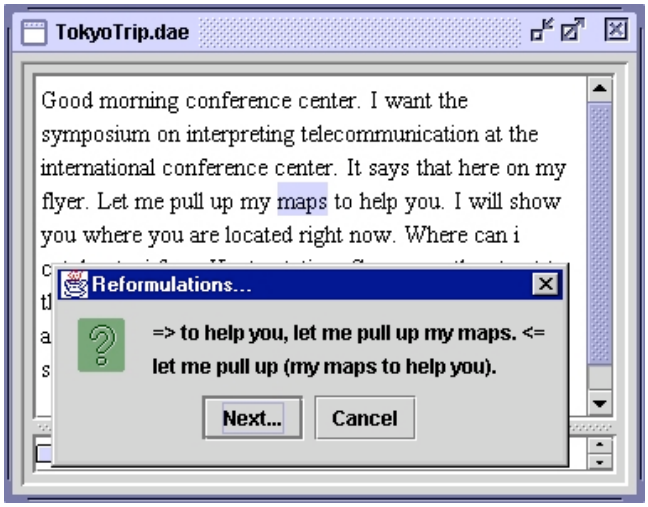

FIGURE 10
REVEALING THE RIGHT INTERPRETATION OF THE SELECTED SENTENCE

Short term planned improvements

In order to improve the current LIDIA-2 implementation, we have several short terms goals. Our long-term goals are described in section 5.

Integration with the French disambiguator

Our first short-term goal is to have the ARIANE HTL modules for LIDIA and the disambiguation module available through a ComSwitch. It would then be possible to offer a richer experiment platform for the project. We will show in section 5.2 that going further on showing the ambiguities implies several changes in the disambiguation module itself.

Integration with a multilingual editor

We have shown that the mirror file may be filtered and visualized through the LIDIA-2 client as a multilingual document. Such a file is not exported yet. It could be interesting to export such a file and have it read and manipulated with a multilingual editor.

Making clarification changeable

In some cases, it may be necessary to go through the disambiguation process again, either to correct a translation result (in this case, the disambiguation may have not been done properly), or to produce another translation (to demonstrate the use of ID).

All necessary information to propose the ID process is available in the mirror object. Thus this process can be done offline. If the new path followed in the question tree is equal to the stored path, then the translations already processed can be kept for future usage. If the path is new, then the already available translations have to be discarded.

FOLLOWING STEPS

Longer-term goals will impact on the HLT modules and/or the disambiguation module.

Handling ambiguity support

To improve practical usability of SEDs, the most important aspect seems to highlight each ambiguity, and hence to locate it, as precisely as possible. To do this, the ID preparation process must output the "support" of each ambiguity [5]. This can be achieved either by computing the support from the current ambiguity descriptors or by modifying the ambiguity descriptors so that they use only the support. Hence, this improvement may let us modify the question construction part of the ID module.

Allowing incomplete interactive disambiguation

In the context of real applications writer may not be willing to answer all questions but only the most crucial ones. Thus, given a partially disambiguated set of mmc-structures and maybe some user preferences or profile, the (automatic) HLT module should be further equipped with ambiguity-hadling rules such as the production of several possibilities ("slashed" translations) and some heuristic-based choice.

Creation of SEDs in target languages

In section 1, we have discussed why it would be very interesting to produce SEDs in target language. Reaching such a goal is very demanding as far as the HLT module development is concerned.

CONCLUSION

In this article we have shown our first implementation of the concept of self-explaining documents. This idea fits in the more general research field on active documents [7]. We are working towards a more sophisticated client embedded within an environment *a la* Thot (http://opera.inrialpes.fr/Thot.en.html).

Our first XML document structure is fairly simple and all the information is not fully "XML-ized". For example, the mmc-structure and the question tree use a lisp-like representation that necessitate a specific handling module although a DOM treatment would be more efficient and portable.

However, these two first steps represent original results in the aimed direction. The perspectives of this work are varied. More practical results will follow in the near future.

ACKNOWLEDGEMENTS

Ghislain Gressard has carried out the implementation of the new Java interface for LIDIA. Eugénie Schonek has realized the SED reader described in this paper.

REFERENCES

[1] Blanchon, H. (1995). *An Interactive Disambiguation Module for English Natural Language Utterances. Proc.* NLPRS'95. Seoul, Korea. Dec 4-7, 1995. vol. **2/2:** pp. 550-555.

[2] Blanchon, H. & Fais, L. (1997). *Asking Users About What They Mean: Two Experiments & Results. Proc.* HCI'97. San Francisco, California. August 24-29, 1997. vol. **2/2:** pp. 609-912.

[3] Boitet, C. (1994). *Dialogue-Based MT and self explaining documents as an alternative to MAHT and MT of controlled language. Proc.* Machine Translation Ten Years On. Oct. 12-14, 1994. pp. 7p.

[4] Boitet, C. & Blanchon, H. (1995). *Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mockup. in* Machine Translationl. vol. **9(**2): pp. 99-132.

[5] Boitet, C. & Tomokiyo, M. (1995). *Ambiguities & ambiguity labelling: towards ambiguity databases. Proc.* RANLP'95 (Recent Advances in NLP)). Tzigov Chark, Bulgaria. September 14-16, 1995. vol. **1/1:** pp. 13-26.

[6] Fais, L. & Blanchon, H. (1996). *Ambiguities in Task-oriented Dialogues. Proc.* MIDDIM'96. Le col de porte, Isère, France. 12-14 Août 1996. vol. **1/1:** pp. 263-275.

[7] Quint, V. & Vatton, I. (1994). *Making structured documents active. in* Electronic Publishing Origination, Dissemination, and Designl. vol. **7(**2): pp. 55-74.

ANNEX : XML DOCUMENTS

For the sake of saving room, the .ldi and .dae files corresponding to the example given in the paper are both given simultaneously. The whole structure of the .ldi and .dae files is given, with the analysis being discarded. The file content that is proper to the .ldi file is given in italics. This is the case for the source, the stamp, the analysis and the translation.

```
<?xml version="1.0" ?>
<work>
 <description>
   <title><![CDATA[A trip to Tokyo]]></title>
   <language><![CDATA[ENG]]></language>
   <auteur>
     <firstname><![CDATA[herve]]></firstname>
     <lastname><![CDATA[blanchon]]></lastname>
   </auteur>
 </description>
 <support>
   <paragraphe>
     <phrase source="ENG" stamp="11054803544635">
       <original><![CDATA[Good morning conference center.]]></original>
       <analyse><![CDATA[…]]></analyse>
       <traduction cible="FRA"><![CDATA[Bonjour ici le centre de conférences international.]]>
       </traduction>
     </phrase>
     <phrase source="ENG" stamp="21054803544655">
       <original><![CDATA[ I want the symposium on interpreting telecommunication at the international
                          conference center.]]></original>
       <question>
         <reformulation choix="NON"><![CDATA[the symposium (on interpreting telecommunications at the
                                          international conference center)]]>
           <question>
             <reformulation><![CDATA[the (international center) for conference]]>
               <analyse><![CDATA[…]]></analyse>
             </reformulation>
             <reformulation><![CDATA[the center for (international conference)]]>
               <analyse><![CDATA[…]]></analyse>
             </reformulation>
           </question>
         </reformulation>
         <reformulation choix="OUI"><![CDATA[at the international conference center, the symposium on
                                          interpreting telecommunications]]>
           <question>
             <reformulation choix="OUI"><![CDATA[the (international center) for conference]]>
               <analyse><![CDATA[…]]></analyse>
             </reformulation>
             <reformulation choix="NON"><![CDATA[the center for (international conference)]]>
               <analyse><![CDATA[…]]></analyse>
             </reformulation>
           </question>
         </reformulation>
       </question>
       <traduction cible="FRA"><![CDATA[Je veux le symposium sur la communication interprétée qui se
                                     déroule au centre de conférences international.]]></traduction>
     </phrase>
     …
     <phrase source="ENG" stamp="51054803544695">
       <original><![CDATA[ I will show you where you are located right now.]]></original>
       <question>
         <reformulation choix="NON"><![CDATA[I will show you (where you are located right now).]]>
           <analyse><![CDATA[…]]></analyse>
         </reformulation>
         <reformulation choix="OUI"><![CDATA[right now, I will show you where you are located.]]>
           <analyse><![CDATA[…]]></analyse>
         </reformulation>
       </question>
       <traduction cible="FRA"><![CDATA[Je vais tout de suite vous montrer où vous êtes.]]></traduction>
     </phrase>
     …
   </paragraphe>
 </support>
</work>
```