

Perspectives of DBMT for monolingual authors on the basis of LIDIA-1, an implemented mock-up

Hervé BLANCHON

GETA, Institut IMAG (UJF & CNRS)
BP 53, 38041 Grenoble Cedex 9, France
e-mail : Herve.Blanchon@imag.fr

Abstract

DBMT is researched here in the context of future systems for the general public, where a monolingual author wants to translate into several languages. We have produced a complete mock-up, LIDIA-1, which demonstrates how a French HyperCard™ stack could be translated into German, Russian and English. We present the computational, linguistic and ergonomic aspects of the mock-up, and discuss them in the perspective of building an operational prototype in the future.

Keywords

Interactive MT, DBMT for monolingual author, Interactive disambiguation, Production of disambiguation dialogues, Distributed architecture, Whiteboard approach

Introduction

Our LIDIA project aims at studying the concept of 'Personal Machine Translation', or more precisely, DBMT for monolingual authors [Boitet & Blanchon 1993], in a multilingual setting.

We have now completed the first implementation of a mock-up, LIDIA-1. Working on a mock-up first, and not on a prototype, has made it possible to tackle all aspects of such future systems, computational, linguistic and ergonomic. Even if we could not solve all problems, we felt they are put in perspective. Almost all other attempts in the direction of DBMT have considered only some aspects of the paradigm, leading to unbalanced and sometimes inadequate architectures.

Before a demonstration which shows the principles of the translation process, we present an overview of the context of the mock-up. Then, we give some more details about the mock-up itself, the implementation techniques used and the principles of the interactive disambiguation process. Finally, we discuss some important points (interface, implementation techniques and tools, and disambiguation process) of the mock-up in the perspective of building an operational prototype in the future.

1. Framework

1.1. The DBMT

Interactive MT was first proposed in the sixties by M. Kay for the MIND system [Kay 1973], and several projects experimented with variations of this design, notably the ITS project [Melby 1981] at Provo (75 - 81), the Alvey N-tran project [Wood 1989] at Manchester (85 - 87), the DLT project [Sadler 1989] at Utrecht (82 - 88), the I.MT project [Rimon, *et al.* 1991] from 1989 at several IBM

research centers, and the JETS project [Tsutsumi, *et al.* 1993] from 1989 at IBM Tokyo Labs.

In KBMT-89 [Goodman & Nirenburg 1991] at CMU-CCI, questions were also asked by the "augmentor" if ambiguities could not be solved by the ontology.

Among those projects where an interactive disambiguation component was integrated, we were inspired by:

- the interface proposed in KBMT-89,
- the pattern-based disambiguation process used for several ambiguities in I.MT,
- the distributed architecture of JETS.

1.2. The LIDIA-1 mock-up

We have chosen a well-defined situation as regard to the profile of the task and the profile of the user. We have integrated the use of an interactive disambiguation process at the very beginning of the design. This means that the whole set of constraints was well established before we started the implementation. The translation process organization is described in [Boitet & Blanchon 1993].

In the scenario we propose, a monolingual French engineer creates technical documentation, in the form of an HyperCard stack, on a middle-range Macintosh, and helps the system translate it into English, German and Russian. We have opted for a distributed architecture (author workstation on a Macintosh and MT server on a mini-IBM-4361).

We have produced a demonstration stack about the linguistics ambiguity we have chosen to cope with in French.

1.3. The demonstration stack

Our demonstration stack, called 'LIDIA les histoires' is made of story cards (Fig. 2) and treatment cards (Fig. 1).

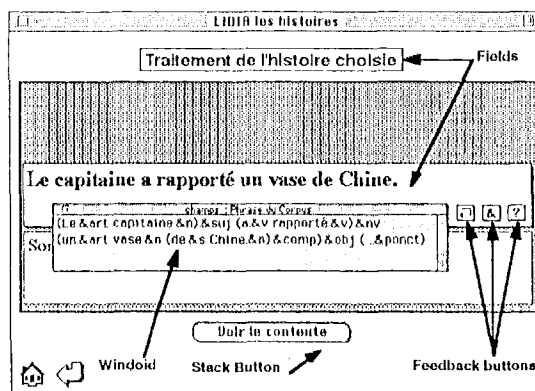


Figure 1: a card and its objects

A story card is a collection of two or three stories sharing an ambiguous sentence. The author is supposed to solve the ambiguities through his understanding of the stories. Here is an example of a story card (see translations¹).

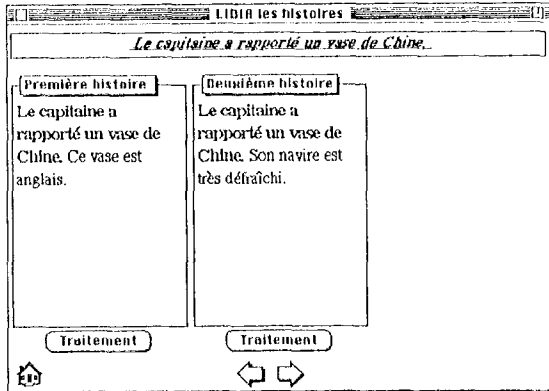


Figure 2: a story card

For the purpose of the demonstration, each story is presented in a treatment card, where the context of the ambiguous sentence may be shown or hidden. Here is the card for the story on the right.

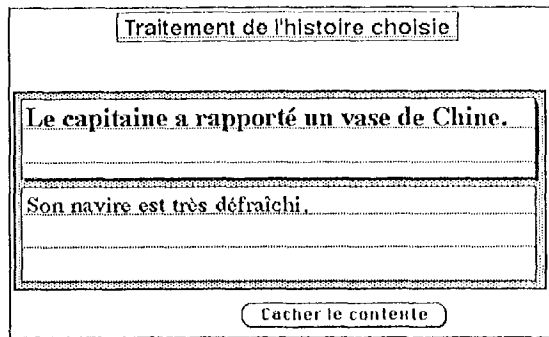


Figure 3: a treatment card

To have the story translated, the user will ask for the translation of the fields of the treatment stack. Note that the user is never interrupted by a question. Objects show they are waiting for answers, and the user decides when and which question to answer.

2. Demonstration

The user can choose the selection tool (✓) and select the object to be translated (Fig. 4).

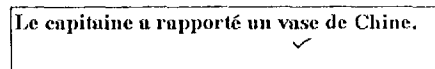


Figure 4: the selection of an object

The button of treatment state then appears. When clicking on it (fig 5), a windoid or pop-up window appears (Fig. 7).

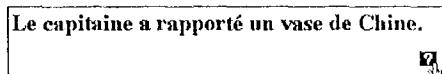


Figure 5: the user asks for the treatment state

¹ Left story: 'From China, the captain has bring back a vase. This vase is English'. Right story: 'The captain has bring back a Chinese vase. His boat is soiled.'

The task in progress is displayed in bold, the previous ones in plain, and the following ones in italic. Thus, in figure 7 the system is currently analyzing the text fragment.

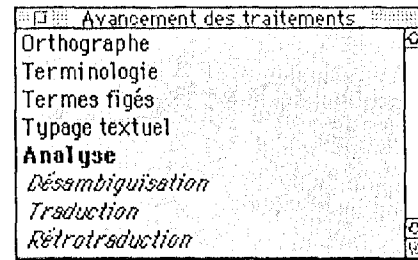


Figure 6: windoid for the treatment state

If the sentence has to be disambiguated, the author is asked to answer some questions. The author is advised a new question is ready by a new item in the menu **Message** and by a button which appears over the concerned object as in figure 7. The user can choose to interact at once or later

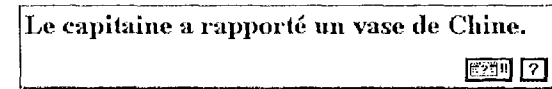



Figure 7: the object have a question of terminology

Suppose the user clicks on the  button. A first question appears (Fig. 8).

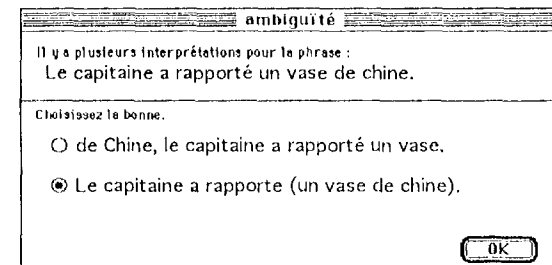


Figure 8: attachment disambiguation (story 2)

With the dialogue (Fig. 8), the author select the attachment of 'de Chine' (Chinese). Then, a second dialogue appears (Fig. 9) where the author chooses the sense of 'capitaine'. The senses are found in Parax, a multilingual lexical database mock-up [Sérasset 1993].

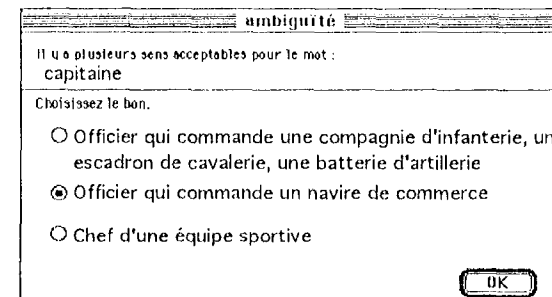


Figure 9: word sense disambiguation (story 2)

Once the disambiguation step has been performed, the user can ask for the annotated form of the text (Fig. 10) which contains the syntactic class of each occurrence and the syntactic function of each phrase.

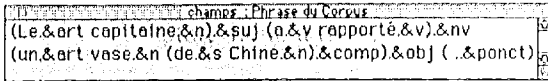


Figure 10: annotated form

These annotations should help the user understand the structure produced by the analyser. We think that experienced users would like to shortcut some dialogues by inserting some disambiguation marks themselves.

To check the translation produced in each target language, the user can ask for the "reverse translation". From German and for the second interpretation of the example, he get :

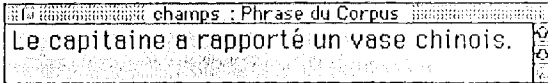


Figure 11: reverse translation²

Finally, the system produces a translated story card.

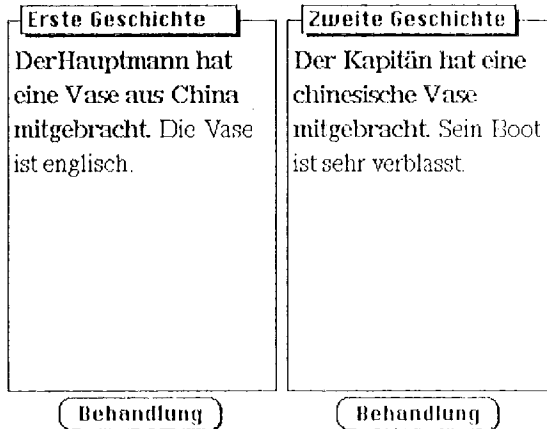


Figure 12: translation of the two stories into German

3. Other aspects

As a demonstration can not show all external aspects of the mock-up, let us now give more details about the interface, the implementation techniques, and the methodology for disambiguation.

3.1. Interaction tools

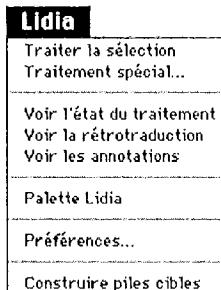


Figure 13: the menu

Once the preferences have been defined, the author uses a menu and a palette to interact with LIDIA.

The interaction with the author is made through the LIDIA menu (Fig. 13), the Messages menu, a palette (Fig. 14), feedback buttons (Fig. 1) and windoids (Fig. 1).

The menu shown here offers 8 choices: process the selected object according to the set of preferences, process some object with a particular preference set, show the

² "The captain has brought back a Chinese vase."

treatments' progress, show the reverse translation, show the annotations, show the palette, modify the preferences and build the target stacks.

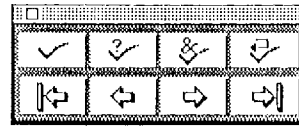


Figure 14: the palette

The user can also ask for the frequent treatments with a palette. In the first line are displayed the LIDIA tools (process the selected object, show the treatment progress, show the annotations and show the reverse translation). In the second line are the browsing tools.

The translation process is divided into two steps: the standardisation and the clarification. We have seen the clarification process during the demonstration, let us have a look on the standardization step.

3.2. Implementation

The implementation is characterized by the use of a distributed architecture, a whiteboard approach, and object-oriented techniques.

a. Distributed architecture

Three machines (Fig. 15) are involved in the translation process.

On the author's workstation the HyperCard Kernel sends and receives messages from the LIDIA kernel which organises the translation process for each object. The LIDIA Kernel sends translation jobs to the Translation server via the Communication server. The LIDIA Kernel also asks to prepare the disambiguation questions.

b. Whiteboard approach

For each object to be translated, the LIDIA Kernel creates a *mirror object* (a text file) in which are stored all information required by the translation process and necessary for the construction of the target stack. We distinguish between *static* and *dynamic information*. Static information is what is attached by HyperCard to each object. It is necessary to construct target stacks. Dynamic information is any information used by LIDIA to translate the content of an object.

These files can be considered as *whiteboards* as defined in [Seligman & Boitet 1994]. Unlike the blackboard, the whiteboard is accessed only by a *coordinator* (the LIDIA Kernel), and not by the components (Disambiguation kernel and RemoteMacMain-Frame). The main advantage of this architecture is to allow easy integration of existing new components without having to modify them.

c. Object oriented techniques

Except the lingware, all components use object-oriented programming. The module for the Terminology, the idioms and the Typage as the kernel of the Communication server are written in HyperTalk the HyperCard scripting language.

The LIDIA serveur is written with CLOS (MCL). Although encapsulated within the same environment, the LIDIA Kernel and the Disambiguation Kernel communicate by exchanging messages and can then be distributed.

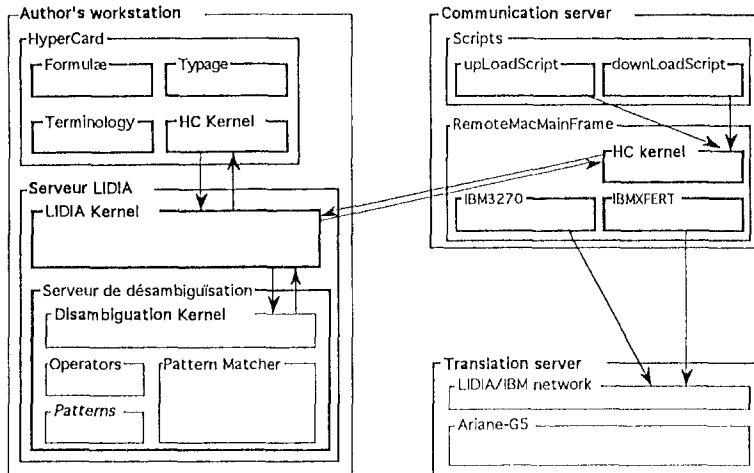


Figure 15: the architecture of LIDIA-1

The disambiguation process is organized around a pattern matcher [Blanchon 1992]. For five out of the eight classes of ambiguity considered in the mock-up, we use a mechanism of pattern matching with unification of variables which allows to recognize the ambiguity and produce the disambiguation dialogue. A dialogue construction method is associated with each pattern. These methods rely on a set of thirteen operators.

Figure 16 shows the trees produced for the sentence 'Le capitaine a rapporté un vase de Chine.'

The patterns (Patron 12 & Patron 13) used to recognize the ambiguity in our example are shown in Fig. 17.

The use of messages and object-oriented programming techniques is close to the actor model used in the context of distributed cooperative systems.

3.3. disambiguation

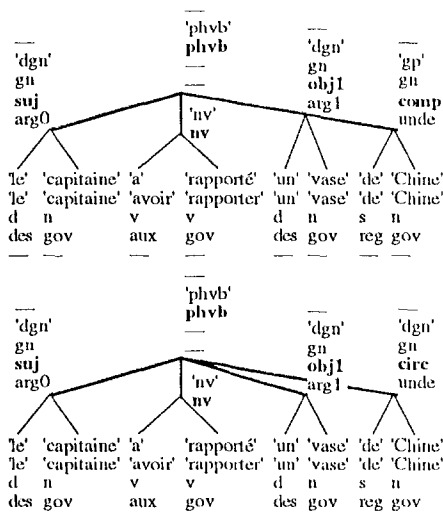


Figure 16: multisolution, multilevel and concrete structure

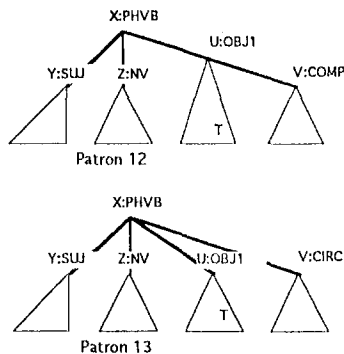


Figure 17: 2 patterns

12 is:

`Texte(Y) Texte(Z) Parenthese(Texte(T), Texte(V))`

which produces the following text:

Le capitaine a rapporté (un vase de chine.)

The method associated with pattern 13 is:

`Texte(V) , Texte(Y) Texte(Z) Texte(T)`

which produces the following text:

de Chine, le capitaine a rapporté un vase.

4. Towards an operational prototype

4.1. Interface

For a prototype, the modules for the terminology and the idioms should use, at least, a lemmatizer, and with the text categorization module they should not rely on HyperCard any more (Fig. 17).

Our implementation of the 'guided languages' idea is still very primitive. We hope to develop working techniques from our studies on 'utterance styles' and 'text genres'.

The interfaces of the standardization modules are only a first sketch. The iconic buttons used to ask for the user intervention have to be redesigned (we haven't found a good solution yet). On the other hand, the cursors for the LIDIA tools and the feed-back buttons are homogenous and could be kept (Fig. 1 & 7).

In a future work, it will be necessary to adapt the dialogue type to the skills of the author. The kind of dialogue we have developed allows only the user to select the right analysis. A new dialogue type could allow the user to get information and examples about the ambiguity currently solved. The user could then change its text or insert disambiguating marks.

4.2. Implementation techniques

The current implementation in terms of software and hardware may be characterized as integrated, distributed and extensible.

Using four servers (LIDIA, Disambiguation, Communication, and Translation) collaborating through messages and text files as made it relatively easy to integrate tools running in different hardware and/or software environments.

For using DBMT at home, a simple communication server could pilot a modem to request services from a LIDIA server, exactly as a mail utility. With such an architecture, a low-cost personal computer would be usable for authoring and translating.

Using object-oriented programming techniques makes the system easy to customize.

4.3. Implementation tools

The dictionaries used by the Ariane-G5 lingware are build from Parax [Sérasset & Blanc 1993]. For a prototype we need a more powerful and flexible tool, as also described in [Sérasset & Blanc 1993].

For developing the lingware, we have used Ariane-G5, designed for heuristic programming in the context of sublanguages. We plan to develop some new Specialized Languages for Linguistic Programming, thereby working in the direction of 'ambiguous programming' [Boitet 1993].

4.4. Disambiguation process

It has been clear from the beginning that we would not be able to find, for each class of ambiguity we have chosen to solve, a unique resolution method. Keeping in mind the kind of dialogues we wanted, we have examined a large quantity of ambiguity configurations and have arrived at 9 problem patterns.

The use of a strategy, organizing the disambiguation process, the use of patterns and methods implemented with a set of basic operators make the process highly customizable. That's why we think about an environment for the description of disambiguation process.

This environment integrates three modules: a module for the patterns definition, a module for the definition of the dialogue production methods, and finally a module for the description of the disambiguation strategy.

Conclusion

The implementation of our mock-up LIDIA-1, first concrete experiment towards the DBMT 'for everybody', has been done 'in breadth' at first, and 'in depth' on certain points. It was very important to tackle all the aspects. Previous experiments have shown the necessity of a broad conception for a MT system to succeed. During our work we have seen that the ergonomics goals can trigger computational and linguistic choices. The situation is the same for the computational or linguistic goals.

The idea of the interactive clarification approach in the context of natural language processing seems now to interest a real community. For MT, the current work of [Wehrli 1993], [Yamaguchi, *et al.* 1993], and the ongoing work on JETS [Tsutsumi, *et al.* 1993] are some good examples. For speech systems, the interactive clarification approach is also a solution as shown in [Frankish, *et al.* 1992] and proposed in [Ainsworth & Pratt 1992] and [Saito 1992].

As far as the future is concerned, we have begun to study multimodal interactive disambiguation with ATR-JIT in a more general framework than LIDIA-1. We hope to get adequate support for developing a more larger-scale prototype in the next few years.

References

- Ainsworth, W. A. & Pratt, S. R. (1992). *Feedback strategies for error correction in speech recognition systems*. in International Journal of Man-Machine Studies. vol. 36(6) : pp. 833-842.
- Blanchon, H. (1992). *A Solution to the Problem of Interactive Disambiguation*. Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. 4/4 : pp. 1233-1238.
- Boitet, C. (1993). *Crucial Open Problems in Machine Translation and Interpretation*. Proc. Symposium on Natural Language Processing in Thailand. Bangkok. 17-21 March 1993, vol. : pp. 1-29.
- Boitet, C. & Blanchon, H. (1993). *Dialogue-based MT for monolingual authors and the LIDIA project*. Proc. NLP'93. Fukuoka, Japon. 6-7 décembre 1993, vol. 1/1 : pp. 208-222.
- Frankish, C., Jones, D. & Hapeshi, K. (1992). *Decline in accuracy of automatic speech recognition as a function of time on task: fatigue or voice drift?* in International Journal of Man-Machine Studies. vol. 36(6) : pp. 797-816.
- Goodman, K. & Nirenburg, S. (eds), (1991). *The KBMT Project: A case study in knowledge-based machine translation*. Morgan Kaufmann. San Mateo, California. 331 p.
- Kay, M. (1973). *The MIND system*. in Courant Computer Science Symposium 8: Natural Language Processing. Algorithmics Press, Inc. New York. pp. 155-188.
- Melby, A. K. (1981). *Translators and Machines - Can they cooperate?* in MFTA. vol. 26(1) : pp. 23-34.
- Rimon, M. & al. (1991). *Advances in Machine Translation Research in IBM*. Proc. Machine Translation Summit III. Washington, D.C. 1-4 juillet 1991, vol. 1/1 : pp. 11-18.
- Sadler, V. (1989). *Working with analogical semantics : Disambiguation techniques in DLT*. Floris Publications. Dordrecht, Holland. 256 p.
- Saito, H. (1992). *Interactive Speech Understanding*. Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. 3/4 : pp. 1053-1057.
- Seligman, M. & Boitet, C. (1994). *A "whiteboard" architecture for automatic speech translation*. Proc. International Symposium on Spoken Dialogue. Waseda University, Tokyo. 1-12 novembre 1993, vol. : pp. 4.
- Sérasset, G. & Blanc, É. (1993). *Une approche par acception pour les bases lexicales multilingues*. Proc. T-TA-TAO 93. Montréal, Canada. 30 septembre-2 octobre 1993, vol. : pp. À paraître.
- Tsutsumi, T. & al. (1993). *Example-Based Approach to Machine Translation*. Proc. Premières journées franco-japonaise sur la traduction assistée par ordinateur. Ambassade de France au Japon, Tokyo, Japon. 15-16 mars 1993, vol. 1/1 : pp. 161-169.
- Wehrli, É. (1993). *Vers un système de traduction interactif*. in La traductique. Les presses de l'Université de Montréal, AUPÉL/URÉF. pp. 423-432.
- Wood, M. M. (1989). *Japanese for speakers of English: The UMIST/Sheffield Machine Translation Project*. in Recent Developments and Applications of Natural Language Processing. Kogan Page Limited. London. pp. 56-64.
- Yamaguchi, M. & al. (1993). *An Interactive Method for Semantic Disambiguation in Sentences by Selecting Examples*. Proc. NLP'93. Fukuoka, Japon. 6-7 décembre 1993, vol. : pp. 208-222.