CHRISTIAN BOITET & HERVÉ BLANCHON
GETA, IMAG (UJF& CNRS)
GRENOBLE
*Langages n° 116, 20—47,1994*

# PROMESSES ET PROBLÈMES DE LA "TAO POUR TOUS" APRÈS LIDIA-1, UNE PREMIÈRE MAQUETTE

## Résumé

Seule la technique de "Traduction Automatique Fondée sur le Dialogue", ou TAFD, permettra d'aboutir à des systèmes de "TA pour tous". Grâce à un dialogue homme-machine de normalisation et de clarification, un auteur pourrait être traduit de façon très correcte dans plusieurs langues, sans les connaître, et sans investir plus de son temps que pour traduire dans une seule langue qu'il connait très bien. L'idée de la TA interactive n'est pas neuve, mais a échoué jusqu'ici pour des raisons ergonomiques et technologiques. Une première maquette, LIDIA-1.0, a été implémentée pour illustrer une partie des solutions que nous proposons pour que cette approche permette d'arriver à des systèmes grand public réellement utilisables. Le résultat est encourageant, même si l'ampleur des efforts et investissements à consentir pour construire et maintenir les bases de connaissances linguistiques nécessaires apparaît de plus en plus comme un problème crucial. Un autre problème, plus théorique, est de trouver des moyens simples et efficaces pour définir des "styles d'énoncés" permettant de réduire la "perplexité" des analyseurs, et donc le nombre de questions à poser pour clarifier un énoncé, et de guider l'auteur de façon naturelle vers le genre de texte, puis vers le style d'énoncé le plus approprié.

## Abstract

Dialogue-Based Machine Translation", or DBMT, will be the only way to achieve "MT for all". Thanks to a normalization and clarification man-machine dialogue, an author could be translated quite accurately into several languages, without knowing them, and without investing more time than to translate into a single language s/he would know quite well. The idea of interactive MT is not new, but has failed until now for ergonomical and technological reasons. A first mockup, LIDIA-1.0, has been implemented to illustrate some of the solutions we propose in order for this approach to lead to systems really usable by the general public. The result is encouraging, even if the sheer size of the efforts and investments necessary to build and maintain the appropriate linguistic knowledge bases appears as a crucial problem. Another problem, more theoretical, is to find simple and efficient ways to define "utterance styles", allowing to decrease the "perplexity" of analyzers, and hence the number of questions to be asked in order to clarify an utterance, and to guide the author in a natural way towards the most appropriate "text genre", and then utterance style.

## Introduction

Situons d'abord rapidement notre approche actuelle de la TAO dans l'évolution du domaine. Vers 1949, les USA, puis l'URSS, ont lancé des programmes motivés par le besoin de renseignements. C'est la *TAO du veilleur,* toujours actuelle. Il s'agit de traduire totalement automatiquement, pour obtenir des traductions "grossières", rapidement, en grand volume et à bas coût. La qualité de ces traductions n'est pas essentielle, car elles servent à filtrer des documents, dont les plus intéressants peuvent, si nécessaire, être traduits ou communiqués à des spécialistes bilingues. Préédition et postédition doivent être absentes ou très limitées (ex : séparer les phrases, les formules, les figures…)[1]. Ce besoin est toujours actuel. Cependant, il s'agit maintenant plus de veille scientifique, technique, économique et financière que de renseignement militaire[2].

Une quinzaine d'années plus tard, les chercheurs se sont orientés vers la *TAO du réviseur*, dans laquelle on veut produire des traductions "brutes", destinées à être révisées. Dans cette optique, la

[1] Les systèmes Systran sont essentiellement de ce type (par exemple, le système russe-anglais installé depuis 20 ans à la Wright-Patterson Air Force Base traduit, d'après nos informations, environ 18 millions de mots par an, avec une qualité tout à fait satisfaisante pour l'usage visé).

[2] À titre d'exemple, on peut citer l'accès en anglais à des bases de données japonaises depuis la Suède [39], ou depuis la CEE/UE (service Japinfo utilisant des traductions "grossières" d'ATLAS-II de Fujitsu, à peines "arrangées").

machine doit remplacer le traducteur, promu réviseur[3]. Typiquement, en moyenne industrielle, une page technique de 250 mots est traduite en 1 h et révisée en 20 mn. Avec 4 personnes, on passerait donc de 3p/h à 12p/h, et on multiplierait donc la productivité par 4 ? Il s'agit d'une limite, le chiffre le plus vraisemblable étant plutôt de 8 p/h, en comptant une révision plus lourde, de 30 mn/p.

Que faire pour la plus grande partie des textes dont on veut obtenir de bonnes traductions ? La bureautique apporte maintenant des solutions variées, sous forme d'outils de *TAO du traducteur*. Il s'agit de traduction humaine assistée par la machine, ou THAM, et pas de traduction automatique. C'est l'utilisateur qui traduit, en s'aidant de dictionnaires bilingues, de bases terminologiques, de thésaurus de "bitextes" (textes + traductions) etc., accessibles depuis un traitement de texte, le tout formant un "poste de travail pour le traducteur", réalisé sur microordinateur ou station de travail[4]. Pour la majorité des besoins, et en particulier pour la traduction de manuels d'enseignement dans des pays où la langue nationale ne s'est que récemment affirmée comme support de l'enseignement secondaire et universitaire, la THAM est actuellement la seule voie réaliste.

Enfin, on commence à voir apparaître des outils de *TAO de l'auteur*, destinés à des personnes ignorant la langue cible. Cela correspond à des besoins croissants. Pour l'instant, il s'agit en fait de textes multilingues préenregistrés, personnalisables grâce à des parties variables. Par exemple, Ambassador™, disponible en anglais-japonais, anglais-français, anglais-espagnol et français-japonais, offre environ 200 "formats" de lettres et formulaires, et 450 "moules" de texte (phrase ou paragraphe)[5]. Nous nous intéressons plutôt à la traduction de textes libres. Dans ce contexte, il n'y a pas encore de produits commerciaux, bien que le système JETS d'IBM-Japon [27] soit prêt au moins depuis 1992. En tout état de cause, il ne pourra s'agir que de systèmes fondés sur le dialogue (TAFD), comme nous le verrons plus loin.

Voici le développement de quelques sigles ou acronymes que nous emploierons fréquemment.

| | | |
|---|---|---|
| BDLM | Base de données lexicales multilingue | |
| TA | Traduction automatique | C'est la machine qui traduit |
| TAO | Traduction automatisée par Ordinateur | Regroupe TA et THAM |
| TAFC | TA fondée sur la connaissance (du domaine) | Utilise une "ontologie" en sus de grammaires et dictionnaires |
| TAFD | TA fondée sur le dialogue | Normalisation et désambiguïsation interactives |
| TAFL | Traduction automatique fondée sur la langue | Technologie classique, reposant sur des dictionnaires et des grammaires |
| THAM | Traduction humaine assistée par la machine | Outils pour traducteurs humains |

# I. La TAFD pour auteur monolingue

## I.1 Opportunité et nécessité de la TAFD

### 1.1. Motivations

Les recherches et développements en TAFD sont motivés par la limitation des paradigmes actuels, par l'importance croissante des langues nationales dans le contexte de l'internationalisation, et par de récents progrès méthodologiques et technologiques.

*a. Limitation des paradigmes actuels*

La TAFL est très bien adaptée à la TAO du veilleur. Des systèmes portables de qualité tout à fait convenable pour l'usage visé commencent à se répandre au Japon, à des prix abordables (environ 5 M¥ pour un système logiciel et matériel complet[6]).

Par contre, elle est loin de pouvoir répondre à tous les besoins en TAO du réviseur. Outre le fait qu'elle demande évidemment autant de révisions que de langues cibles, elle reste trop chère pour des usages légers. En effet, selon les chiffres donnés par les producteurs de systèmes de TA [22] la

[3] La plupart des systèmes commerciaux récents visent ce créneau. On peut citer des systèmes japonais (AS-Transac de Toshiba, ATLAS-II de Fujitsu, PIVOT de NEC, HICAT de Hitachi, DUET de Sharp, SHALT d'IBM-Japon, PENSÉE de OKI, Majestic de l'Université de Kyoto et du JICST,…), ou américains (LOGOS, METAL), ou français (Ariane/aéro/F-E de SITE-B'VITAL, fondé sur les outils informatiques et les méthodes linguistiques du GETA). Il est aussi possible d'adapter des systèmes de conception plus ancienne à cet usage, si on les spécialise à un langage fortement contrôlé, comme cela a été fait chez Xerox avec Systran pour la traduction de notices d'anglais en 4 langues.

[4] Il s'agit d'accessoires de bureau (MTX™ sur PC, WinTool™ sur Macintosh), ou d'outils intégrés et plus riches (Trados d'Ink, Translation Manager d'IBM, et récemment EuroLang Optimizer™ de SITE-EuroLang).

[5] Disponible sur Macintosh et PC, produit par Language Engineering Corp., Belmont, MA 02178.

[6] Station Sparc, lecteur optique, système avec 80.000 termes, dictionnaire utilisateur, et options de personnalisation.

création *ex nihilo* d'un système opérationnel de TAFL coûte entre 200 et 300 hommes-années, avec des développeurs spécialisés, et l'adaptation d'un système de TAFL existant à un nouveau domaine et à une nouvelle typologie de textes coûte de l'ordre de 5 à 10 hommes-années. Un utilisateur n'a intérêt à s'équiper d'un tel système que s'il doit traduire de gros flux de textes homogènes et informatisés, comme des manuels d'utilisation ou de maintenance[7]. Adapter un système disponible à des besoins ponctuels n'est pas non plus une solution viable[8].

D'autre part, une condition essentielle de succès de la TAO du réviseur est de constituer une équipe de développement et de maintenance des linguiciels (dictionnaires, grammaires) qui soit en liaison constante avec l'équipe de révision, et si possible avec les auteurs des documents à traduire[9]. C'est ce qu'a réussi la PAHO (Pan American Health Association), avec ses systèmes ENGSPAN et SPANAM [44]. On peut faire un parallèle avec les systèmes experts, qui peuvent être développés par des tierces parties, mais qui doivent ensuite être totalement maîtrisés par leurs utilisateurs, seuls à même de les faire évoluer de façon adéquate.

En ce qui concerne la TAFC [15, 31, 32], elle est totalement inapplicable en TAO du veilleur, et elle est moins applicable que la TAFL en TAO du réviseur, car, tant qu'il faudra construire les "ontologies" spécialement pour la traduction, elle restera beaucoup plus onéreuse, et ce pour un résultat guère meilleur.

*b. Importance croissante des langues nationales et de l'internationalisation*

De plus en plus, nous désirons rédiger dans notre langue, et transmettre nos textes à l'étranger, qu'il s'agisse de messages électroniques, de lettres, d'articles, de manuels techniques, voire de livres. Contrairement à ce que d'aucuns prédisaient il y a une cinquantaine d'années, l'internationalisation croissante ne s'est pas accompagnée d'une uniformisation linguistique vers l'anglais, mais au contraire d'un renforcement considérable de l'usage scientifique et technique de langues déjà importantes de ce point de vue, comme le japonais, et d'une promotion volontariste de bien d'autres, comme le malais-indonésien ou l'arabe, pour les amener au même niveau. À notre sens, et même si l'anglais continue de se renforcer en tant que volapük scientifique moderne, cette évolution ne fera que se renforcer, les langues étant, pour reprendre une expression de C. Hagège dans un article paru dans *Le Monde* début 1990, les "drapeaux des identités nationales".

Il ne s'agit pas seulement de politique, mais d'efficacité. Dans les projets coopératifs européens (Esprit, Eureka), par exemple, la communication est gênée par la nécessité de lire et d'écrire en anglais. Pour la grande majorité des participants, lire en anglais pose des problèmes de compréhension et prend beaucoup de temps. Quant à écrire, si même c'est envisageable, le résultat est souvent difficile à comprendre, voire illisible. De grandes sociétés comme Thomson chiffrent à plusieurs dizaines de MF par an les pertes dues aux problèmes de communication multilingue.

Les trois types de TAO "classique" ne peuvent évidemment répondre à ce nouveau besoin. En effet, la TAO du veilleur, sans préédition ni postédition, ne peut donner une qualité suffisante, et la TAO du réviseur comme la TAO du traducteur s'adressent par définition à des spécialistes au moins bilingues, et non à des rédacteurs supposés ne connaître aucune des langues cibles, ou au plus une, et ce imparfaitement.

*c. Progrès méthodologiques et technologiques*

L'idée de la TAFD date des années soixante [23], et a été incorporée à plusieurs maquettes ou prototypes dans les années soixante-dix et quatre-vingt [17, 21, 30, 41, 42, 49, 54, 55]. Si ces travaux n'ont pas donné lieu à des systèmes utilisables en pratique, c'est à notre avis que les

[7] En prenant l'hypothèse d'un système coûtant 1 MF (400 KF de base et 600 KF de spécialisation au vocabulaire et au type de texte) et d'un amortissement sur 2 ans, il faut un flux de 10.000 p/an (en comptant 10%/an de maintenance, 60 F/page de coût machine, et 100 F/page de révision, contre 150 F/page de traduction et 70 F/page de révision pour la méthode manuelle, soit 60 F/page de gain pour amortir 1,2 MF). À coût machine nul, il faudrait encore 5.000 p/an. Une autre façon de comparer est de dire qu'il faut de 5 à 10 hommes-années pour spécialiser un système, et 8 pour traduire et réviser 10.000 pages (par exemple, on peut le faire en un an avec 6 traducteurs et 2 réviseurs travaillant 1700 h/an).

[8] Ce serait comme réoutiller une usine pour produire quelques dizaines de voitures. En effet, sans compter la saisie optique ou manuelle, entraînant toujours un coût important de vérification, ni la maintenance, ni même l'achat du système de base, mais seulement sa spécialisation (600 KF) et les coûts de traduction et de révision, on arrive avec les hypothèses précédentes à 632, 680, 760 et 920 KF pour 200, 500, 1.000 et 2.000 pages, contre 44, 110, 220 et 440 KF pour la méthode classique manuelle, soit environ 14,5, 6, 3,5 et 2 fois plus, respectivement (chiffres de 1991).

[9] Dans le "contre-rapport ALPAC" du JEIDA [22]comme au MTS-II à Munich en août 1989, Fujitsu reconnaissait clairement avoir fait une erreur en distribuant largement ATLAS-II : seules étaient en effet rentables les traductions effectuées chez Fujitsu, soit pour sa documentation, soit dans le cadre d'un contrat avec la CEE (Japinfo) concernant la veille technologique et ne demandant donc qu'une révision minimale.

dialogues devaient être conduits par des spécialistes[10], que la couverture linguistique était trop limitée, et que l'on ne disposait pas encore d'environnements interactifs conviviaux.

La méthodologie s'est affinée ces dernières années. Tout d'abord, l'utilisateur envisagé n'est plus un spécialiste, mais un rédacteur, ou plutôt un *auteur* [1, 6, 9, 20, 27, 36, 40]. Nous préférons ce dernier terme. D'un côté, en effet, "auteur" est moins restrictif que "rédacteur" : un auteur est quelqu'un qui veut créer un texte, et peut le faire en l'écrivant, en le dictant, ou encore en le construisant interactivement. De l'autre, "auteur" est plus restrictif que "rédacteur", "locuteur", ou "commentateur", car "auteur" désigne quelqu'un qui désire créer un produit final "propre", alors que les autres termes peuvent renvoyer à des personnes désirant seulement produire un message écrit ou parlé de façon "spontanée", en vue d'une communication immédiate, et non disposées à conduire un dialogue éventuellement lourd pour rendre leur message "propre"[11].

D'autre part, l'informatique personnelle a fait des progrès gigantesques. On dispose maintenant d'ordinateurs personnels très puissants et bon marché, d'environnements conviviaux, de l'intégration du multimédia, et d'outils de télécommunication permettant le recours à des serveurs. Enfin, les techniques et outils de génie logiciel modernes (essentiellement la programmation par objets), permettent de construire des systèmes complexes et interactifs bien plus rapidement et sûrement que par le passé.

### 1.2. Critères de choix de l'approche par dialogue

Nous proposons quatre critères pour choisir la TAFD :

- la qualité visée doit être élevée, et la révision impossible ou très coûteuse ;
- le contexte doit être fortement multilingue (1—>n, comme pour la dissémination de documentation technique, ou n<—>n, comme dans des projets internationaux) ;
- l'entrée ou le domaine ne doivent pas être trop contraints ou contrôlés (sinon, mieux vaut utiliser la TAFL ou la TAFC) ;
- les utilisateurs doivent être prêts à participer à des dialogues de normalisation et de désambiguïsation.

Dans toute situation, il faut de plus pouvoir rendre les dialogues acceptables, en les laissant à l'initiative de l'utilisateur, en lui fournissant des moyens de les contrôler et de les réduire (en jouant sur des paramètres, en insérant directement des marques de désambiguïsation, etc.), et en lui laissant si possible le choix entre plusieurs média.

### 1.3. Situations traductionnelles adaptées à la TAFD

#### *a. Entrée écrite*

Parmi les situations favorables avec entrée écrite, on peut mentionner :

- la traduction de volumes relativement faibles de documentation technique en plusieurs langues, typiquement 5.000 à 8.000 pages à distribuer sur un Disque Optique Numérique, par exemple dans les 9 langues officielles de la CEE/UE (et peut-être dans d'autres aussi, comme le russe, l'arabe, le japonais, ou le chinois).
- la diffusion d'information dans plusieurs langues (sur la circulation, sur la météo, ou dans des congrès, des manifestations sportives, des situations d'urgence…), qui demande une sortie orales aussi bien qu'écrite ;
- l'échange télématique de notes et de documents de travail dans des projets internationaux.

#### *b. Entrée orale*

Comme l'état de l'art en reconnaissance de parole ne permet pas de traiter à la fois un grand vocabulaire, de la parole continue, et un locuteur arbitraire, il semble n'y avoir que très peu de situations favorables à la TAFD avec entrée orale :

- la production de commentaires ou de résumés à partir de scènes visuelles et auditives (par exemple, pour le sous-titrage d'émissions de télévision en langues étrangères) ;

---

[10] ITS [30] demandait même *plusieurs* intervenants, un linguiste spécialiste du système pour l'analyse, et un bilingue pour chaque langue cible.

[11] "Propre" signifie ici conforme à une certaine grammaire (permettant éventuellement des constructions incorrectes, pourvu qu'elles soient attestées) *et* ne comportant pas de parties "réflexives", ou "automodificatrices", si fréquentes dans la parole et même l'écriture spontanée (au moins manuscrite), comme des hésitations, des faux départs, des reprises, des répétitions, des corrections, des abréviations arbitraires (apocopes), etc.

- l'interprétation de dialogues bilingues très contraints, tels que les appels téléphoniques de politesse entre parents d'enfants échangés entre familles pour apprendre les langues, ou l'assistance téléphonique à des voyageurs étrangers (consultation médicale, réservation…). Ici, le dialogue doit être le plus réduit possible, et une combinaison entre TAFD et TAFC (analogue à l'architecture finale de KBMT-89 [31]) semblerait indiquée.
- la production de présentations vidéo multilingues.

*c. Création dans une langue d'un message dans une autre*

Il y a enfin beaucoup de situations intéressantes où le message source n'est créé que pour vérifier le contenu du (ou des) message cible. Le message est créé par interaction avec le système. C'est par exemple le cas de lettres officielles ou formelles, qui ont des structures très différentes dans différentes cultures. Somers & Tsujii [40] ont ici proposé le terme de "traduction sans texte source", mais il serait plus exact de parler de génération interactive multilingue que de traduction.

### I.2 Architecture linguistique et voies intermédiaires nouvelles

Le fait que TAFD suppose un dialogue avec l'auteur permet d'envisager de nouvelles possibilités au niveau des traitements linguistiques. Il ne s'agit pas de solutions radicalement nouvelles, mais, comme souvent dans un domaine technique, de nouveaux compromis, de voies intermédiaires nouvelles, avec ça et là des innovations intéressantes.

Dans la majorité des situations adaptées à la TAFD, il faut un système de couverture lexicale et grammaticale très large. D'où des questions théoriques importantes :

- Sachant qu'on n'obtient de bons résultats que sur des langages restreints, *comment construire une base de connaissances linguistiques utilisable comme une union de sous-langages ?* Est-il possible de séparer les aspects grammaticaux et lexicaux ?
- *Comment atteindre la large couverture lexicale nécessaire ?* Typiquement, un système de TAFL contient de $3.10^4$ à $3.10^5$ termes, en 2 langues. Le cas de METEO ($3.10^3$ termes) est atypique, à cause de son domaine très restreint [16]. Mais un système de TAFD visant le grand public et non restreint à un domaine particulier demandera de $3.10^5$ à $3.10^6$ termes, et ce en plusieurs langues !
- Dans des situations fortement multilingues, l'approche par interlingua est séduisante. Mais. *comment surmonter les difficultés d'ingénierie rencontrées dans la construction d'un grand lexique interlingue* par les récents projets japonais ATLAS, PIVOT, EDR, et CICC ?
- Il est crucial que des non-spécialistes puissent facilement comprendre les questions du système, éventuellement lui demander les raisons de certaines questions, et comprendre ses réponses. Une question importante (et nouvelle) est donc de trouver *comment rendre la base de connaissances linguistiques d'un système de TAFD accessible à un utilisateur naïf.*

Bien que les réponses proposées ici ne soient ni complètes ni définitives, elles sont basées sur une longue pratique de la TAFL et sur une étude approfondie de beaucoup de systèmes de TA passés ou présents, en particulier ceux qui utilisent une étape de désambiguïsation interactive.

#### 2.1. Transfert multiniveau à acceptions, propriétés et relations interlingues

Les systèmes modernes de TAO sont fondés sur un *transfert sémantique*, le *passage par un interlingua*, ou un *transfert multiniveau.* "Transfert sémantique" est un terme introduit par les Japonais pour désigner exactement l'approche du CETA entre 1960 et 1970, que B. Vauquois, reprenant un terme dû à Shaumjan, appelait approche par "pivot hybride" : la structure interface source fournie au transfert contient des éléments lexicaux propres à la langue source, et des propriétés et relations interlingues (traits sémantiques, cas profonds, traits abstraits de nombre, aspect, modalité, détermination…). Dans un véritable "interlingua", les éléments lexicaux doivent eux aussi être interlingues ("concepts" s'il y a une véritable ontologie, "acceptions" sinon).

Le *transfert multiniveau*, au sens de Vauquois [45, 46] diffère du transfert sémantique en ce que, outre les attributs et relations interlingues, on garde des attributs et des relations spécifiques à la langue source (classe syntagmatique, genre, nombre, détermination, temps, mode, fonction syntaxique…). Cela donne des "filets de sécurité", et permet de faire interagir les niveaux.

En TAFD, nous proposons de rajouter aux représentations multiniveaux un niveau lexical, celui des *acceptions interlingues*, sans aller jusqu'à introduire des concepts, puisqu'il faudrait alors construire une ontologie. La base lexicale multilingue sous-jacente (BDLM) contiendra alors un dictionnaire monolingue pour chaque langue traitée par le système, et un dictionnaire interlingue pour les acceptions interlingues. Chaque acception interlingue a une image dans chaque dictionnaire

monolingue, avec une définition appropriée dans la langue correspondante, utilisée lors de la désambiguïsation interactive du sens[12].

Qu'on nous permette d'insister sur le fait que le terme "interlingue" ne signifie pas pour nous "indépendant de toutes les langues", mais "intermédiaire entre les langues connues du système". Par exemple, si le système travaille avec le fançais, l'anglais et le russe, il y aura une seule acception pour "mur" en tant qu'objet concret. Dès qu'on ajoutera l'allemand ou l'italien, il faudra ajouter les raffinements "mur vu de l'extérieur" (Mauer, muro) et "mur vu de l'intérieur" (Wand, parete).

### 2.2. Approche par "langage guidé"

*a. Séparation lexique/grammaire*

La notion de "sous-langage" a été introduite et étudiée par R. Kittredge [24, 25] à la suite de son expérience en TAO (il fut directeur du groupe TAUM de l'Université de Montréal au début des années 70). Il a donné un certain nombre de critères, essentiellement lexicaux et syntaxiques, pour évaluer la difficulté d'un sous-langage pour les techniques de TAO du réviseur, et pour déterminer si l'approche dite de "deuxième génération avec sous-langage" (ou par "sous-optimisation") était prometteuse, et les a appliqués à un certain nombre de types de textes.

Kittredge distingue les aspects lexicaux et grammaticaux, la liaison plus ou moins forte avec un domaine sémantique clos, et la possibilité d'écrire une grammaire textuelle dépassant le niveau de l'énoncé. Il introduit la notion formelle de "clôture lexicale", qui signifie *grosso modo* que le nombre de nouveaux termes rencontrés dans une nouvelle page diminue rapidement et tend vers zéro ou une valeur très faible quand le nombre de pages augmente.

Mais il ne propose aucune notion formelle analogue pour l'aspect grammatical : un sous-langage est défini comme l'ensemble des phrases (ou des énoncés) produisibles dans des conditions fixées (par exemple, bulletins météo, appels d'offres du Secrétariat d'État, manuels de maintenance de tel avion, etc.), sans qu'on sache comment choisir un échantillon convenable et comment généraliser autrement qu'intuitivement. De plus, le terme choisi est gênant : Kittredge fait lui-même remarquer qu'un sous-language d'une langue n'est pas un sous-ensemble de cette langue (cela est dû au fait que "le français" signifie en général le français standard des manuels, et pas l'union de tous ses jargons et parlers régionaux). Dans notre contexte, il serait cependant très souhaitable de pouvoir définir des sous-langages à partir d'autres sous-langages par des opérations ensemblistes.

D'autre part, si l'usage de la TA se répand grâce à la TAFD, il sera impossible de produire et de maintenir une collection très variée de bases lexicales et grammaticales de grande taille correspondant à des sous-langages au sens de Kittredge, une pour chaque type d'utilisation. À terme, il faudra donc un dictionnaire total aussi exhaustif que possible, et cela ne sert à rien d'utiliser la notion de "clôture lexicale" pour le limiter. La même chose est vraie de la grammaire.

Pour réduire le problème (diviser pour régner !), il y a une voie intermédiaire, qui consiste à séparer les deux aspects, puis à définir chacun d'eux en deux temps, d'abord de façon grossière à l'aide d'un formalisme symbolique simple, puis de façon plus fine en ajoutant des paramètres numériques. Un type de texte donné pourra alors être défini comme un ensemble de poids relatifs à des arcs et à des nœuds d'une BDLM structurée en réseau sémantique, son "profil lexical", et comme un style d'énoncés ou un genre de texte vérifiant certaines contraintes numériques.

*b. Préférences lexicales*

La BDLM d'un système de TAFD doit contenir des relations entre acceptions qui permettent d'en extraire des thésaurus, en particulier la synonymie, la quasi-synonymie et l'hypéronymie, et des relations analogues aux relations lexico-sémantiques entre termes (comme entité —> qualité, action —> argument 1 de l'action…).

Les poids attachés aux acceptions, aux termes et aux relations entre eux constituent ce qu'on pourrait appeler un *profil lexical.* Au fur et à mesure que le temps passe, le système de TAFD peut les faire varier en fonction de l'interaction, ce qui permet un certain réglage automatique, et la définition de nouveaux profils lexicaux reflétant les préférences lexicales courantes. Les poids sur les termes reflètent leur "degré de pertinence" par rapport à la tâche en cours, et peuvent être utilisés pour indiquer à l'auteur le terme préféré parmi un ensemble de (quasi-)synonymes (comme par exemple avion, appareil, aéronef). Associés aux poids sur les acceptions et sur les relations entre termes et acceptions, ils peuvent aussi être utilisés pour lever des ambiguïtés de sens, comme cela a été montré dans [48]. En TAFD, cela peut servir à présélectionner le sens le plus probable.

[12] On peut aussi *expliquer* à l'auteur pourquoi une telle question est posée, et même montrer les mots en question dans les autres langues. L'introduction d'aspects d'auto-apprentissage dans ce genre de systèmes les rendrait plus acceptables par les utilisateurs potentiels.

Remarquons que, dans le contexte d'un système de "TAFD pour tous", la base lexicale devra contenir une très grande variété de termes, même incorrects ou douteux, alors que les bases de données terminologiques ne contiennent d'habitude que des termes normalisés ou recommandés.

*c. Types de textes : "styles d'énoncés" et "genres de textes"*

En ce qui concerne l'aspect grammatical, on peut proposer de diviser encore le problème en deux, en définissant des *styles d'énoncés* pour les phrases et autres énoncés traduisibles individuellement (titres, éléments homogènes de longues énumérations…) et des *genres de textes* pour les fragments plus longs.

Pour cela, nous supposons que nous avons recensé toutes les constructions d'une langue, y compris les constructions rares ou n'apparaissant que dans des types de textes très techniques (ex. : "Mettre interrupteur sécurité train avant sur OFF"), et que nous les avons représentées au moyen d'un très grand ensemble R de règles déclaratives. Tout formalisme déclaratif simple convient[13].

Un *style d'énoncé* est alors un sous-ensemble de R vérifiant certaines restrictions numériques (par exemple, sur le degré d'imbrication, d'ellipse, ou de coordination). Par exemple, M1 pourrait être le style des phrases simples sans sujet ("permet de sauver sur disque une copie de votre fichier"), fréquentes dans les documents techniques, et M2 le style des phrases explicatives simples.

En ce qui concerne les *genres de textes*, il est souhaitable qu'on puisse dans le futur les prendre en compte dans des éditeurs de documents structurés fondés sur SGML[14], qui peuvent traiter des fragments beaucoup plus longs que les outils linguistiques actuels, le plus souvent limités à la phrase ou au paragraphe. On peut même proposer de définir un genre de texte directement en SGML. Par exemple, le genre de texte S1 des paragraphes commençant par une phrase de style M1 suivie par une suite (éventuellement vide) de phrases de style M2 pourrait être défini par[15] :

```
<!ELEMENT     S1         - 0   (M1, M2*)                          >
```

Pour décrire le genre de textes d'un "document" commençant par un "titre" (de style d'énoncé M3), et se poursuivant par une liste non vide de paragraphes (de styles d'énoncé M2 et/ou M4) et/ou de sections de même structure qu'un document, on pourrait de même écrire :

```
<!ELEMENT     Document   - 0   (Titre, Contenu)                   >
<!ELEMENT     Titre      - 0   (M3)                               >
<!ELEMENT     Contenu    - 0   (Paragraphe | Document)+           >
<!ELEMENT     Paragraphe - 0   (M2 | M4)+                         >
```

Pour l'instant, nous ne savons pas très bien comment guider les auteurs dans la sélection d'un style d'énoncés ou d'un genre de textes particulier, de façon à ce que la critique textuelle, la standardisation et la clarification puissent être menées avec efficacité.

### 2.3. Accessibilité des connaissances

Dans la plupart des systèmes de TAO, les informations linguistiques sont très détaillées, et codées de façon compréhensible uniquement par des spécialistes. Dans certains, qui se présentent plutôt comme des aides au traducteur humain (Weidner, ALPS), on a au contraire cherché à n'utiliser que des informations très simplifiées, pour que des utilisateurs naïfs puissent coder eux-mêmes les dictionnaires. Le résultat a simplement été que les traductions étaient trop mauvaises pour servir de base à une révision[16].

En TAFD, il nous semble que l'information accessible à l'utilisateur ne doit pas être simpliste, mais peut-être moins fouillée que dans le cas de la TAFL. Par exemple, il est sans doute inutile d'avoir un système de codes sémantiques trop riche. Une hiérarchie à 3 ou 4 niveaux, avec au plus une dizaine de codes par niveau, est sans doute le maximum si l'on veut que des utilisateurs puissent compléter les dictionnaires, qui, même très grands, ne seront jamais exhaustifs.

Un second aspect, très délicat et intéressant, est de trouver comment exprimer les notions linguistiques obscures pour le commun des mortels d'une façon compréhensible. Pour certaines,

---

[13] Par exemple, les grammaires hors-contexte (CFG), avec ou sans attributs, les TAG, les STCG [47, 56, 57] Pour le moment, nous utilisons le langage spécialisé ROBRA dans LIDIA-1, malgré son caractère procédural, et testons dans chaque règle transformationnelle si le style d'énoncé attendu est l'un des styles contenant cette règle.

[14] Standard Generalized Markup Language [43].

[15] Un symbole suivant "<!ELEMENT" désigne un genre de texte, "|" l'alternation, "*" et "+" la répétition. Les symboles entre parenthèses désignent des genres de texte ou des sytles d'énoncé. On pourrait ajouter des conditions sur des attributs associés aux symboles principaux.

[16] En 1985, le Bureau des Traductions d'Ottawa mena une étude consistant à comparer la productivité de traducteurs humains utilisant le système Weidner sans, puis avec l'option de TA, en leur faisant traduire le même livre sur l'odontologie à 3 mois de distance. Dans le second cas, la productivité baissa de 40% !

comme l'aspect (ex.: "le courrier est arrivé" —> "the mail has arrived" / "the mail arrived" ?), il faut sans doute utiliser des exemples, ou de la reformulation, et éviter de parler de la notion elle-même.

Enfin, il faut arriver à organiser le système de façon à ce que l'utilisateur, même monolingue, puisse contrôler ce que produit le système dans les langues cibles. Nous proposons pour cela un mécanisme de "rétrotraduction" (cf. infra).

### 2.4. Vers un codage des textes adapté à la préédition directe en contexte multilingue

L'idée de base en TAFD est de remplacer les postéditions (une par langue cible) par une seule préédition. Ainsi, le résultat de l'interaction avec l'auteur se traduit par des annotations sur le texte soumis au processus automatique : c'est une préédition indirecte. Mais nous pensons que les utilisateur expérimentés voudront aussi prééditer directement, en insérant eux-mêmes certaines annotations, pour réduire le dialogue. Il convient donc de pouvoir présenter le texte enrichi par des marques de désambiguïsation, ainsi que les résultats d'analyse, comme une chaîne de caractères *portable* et *lisible*[17]. Nous partageons cette idée (de préédition indirecte et/ou directe) avec d'autres projets, comme le projet LMT d'IBM [35][18].

#### *a. Encodage interne de textes multilingues dans un jeu de caractères universel*

Le premier problème est de traiter des textes dans des langues écrites avec des systèmes d'écriture variés. Jusqu'aux années quatre-vingt, les systèmes informatiques n'offraient qu'un choix très réduit de jeux de caractères (comme romain sans diacritiques, mélange cyrillique/romain sans minuscules), de sorte qu'il était impératif d'utiliser des transcriptions. Dix ans après, presque tous les constructeurs d'ordinateurs commencèrent à offrir des jeux de caractères étendus et les systèmes d'exploitation "localisés".

Mac.OS-7.1, disponible depuis fin 1992, est le premier système d'expoitation complètement multiscript[19] : avec n'importe quel texteur utilisant le "Script Manager" standard, on peut composer un document contenant des parties dans presque toutes les langues d'Europe, ainsi qu'en arabe, japonais, chinois, etc. Cela nous a conduit à choisir le Macintosh comme station de rédaction. Cependant, Mac.OS-7.1 est encore un cas unique[20], et le problème reste entier si l'on veut échanger du texte entre ordinateurs de différentes marques, ou simplement transmettre du texte via les réseaux télématiques.

Notre solution est d'utiliser des transcriptions romaines pour les représentaitons internes des textes, des grammaires et des dictionnaires. Une transcription consiste en un jeu de caractères et une méthode pour représenter le matériau textuel considéré (pas seulement les mots, mais aussi la structure logique, la mise en page, et éventuellement d'autres informations), en n'utilisant que ces caractères. Le jeu de caractères et la méthode à utiliser dépendent de l'importance relative accordée à la portabilité, à la lisibilité et à la compacité.

Pour la TA "classique", nous avons depuis longtemps utilisé un jeu de caractères de transcription presque identique à celui de PL/I (ni minuscules[21] ni diacritiques, mais seulement les majuscules, les signes de ponctuation usuels et quelques signes spéciaux). Cela donne une portabilité totale, aux dépens de la lisibilité.

Par exemple, `'*A!2 *NOE!4L , *MAC*ALLISTER VA AUX **USA'`
code : `'À Noël, MacAllister va aux USA'.`

Pour la TAFD, la lisibilité est plus importante, et nous proposons d'utiliser un sous-ensemble plus grand de l'ISO-646, contenant les majuscules et les minuscules, mais pas les diacritiques, et de représenter les diacritiques par des "séquences spéciales" introduites par "!". En effet, dans la documentation technique, les pièces sont souvent référencées par des identificateurs où lettres et chiffres sont mélangés (par ex.: XA1). L'information relative à la structure ou à la mise en page du texte devra alors être représentée par des marques, ou "balises", dans l'esprit de SGML et de la TEI

---

[17] Les étiquettes, entités et transcriptions devraient être définis dans l'esprit de la TEI (Text Encoding Initiative).

[18] «The user can mark the input string selectively with brackets <…> (to any degree) to force parsing choice and desambiguate the input. "User" in this context can also apply tools (such as the interactive disambiguator) which may introduce such marks in their output.»

[19] Mac.OS-7.1 n'est pas encore lui-même multilingue : bien que tous les programmes soient indépendants d'une langue particulière, chaque version distribuée contient les messages et les ressources spécifiques de la langue de localisation.

[20] Le "documenteur" Star™ de Xerox a été le seul outil vraiment multilingue jusqu'à la sortie de WinText™ sur Macintosh en 1987, mais les systèmes d'exploitation sous-jacents étaient respectivement strictement monolingues, ou seulement localisables.

[21] En Thaïlande, par exemple, les minuscules sont remplacées par les caractères thaïs dans les terminaux bilingues.

(<parag>, <section>, <greek>, etc.). On indiquera de façon analogue un changement de langue et/ou de système d'écriture (certaines langues en utilisent plus d'un).

*b. Encodage d'informations linguistiques pouvant provenir d'une préédition*

Les syntagmes figés spéciaux seront transformés en des occurrences spéciales, de façon à aider l'analyse et la traduction. Par exemple, "Cacher les bulles" donnera `&FXN_Cacher_les_bulles,` qui pourra être traité par une sous-grammaire morphologique appropriée.

Après la désambiguïsation lexicale, nous pourrions attacher à chaque occurrence le numéro du sens dans la BDLM, par exemple `glace.1` pour "glace à manger" et `glace.2` pour "miroir". Mais cela n'est pas très lisible et interdit la préédition directe. Nous permettons donc d'ajouter au numéro de sens un fragment de la définition qui donne la distinction, la "clé sémantique", d'habitude un autre mot ou terme, ce qui donne `glace.1=aliment` ou `glace.2=miroir`. Comme en général plus d'une clé sémantique peut être associée à une même acception, par exemple `glace.1=a_manger` ou `glace.1=dessert`, il faut aussi permettre à l'utilisateur d'entrer `glace=dessert.` Dans ce cas, le système devrait consulter la BDLM, trouver que l'acception de "glace" la plus proche de "dessert" est le sens 1, et transformer `glace=dessert` en `glace.1=dessert` ou même en `glace.1=aliment`.

Les annotations concernant les informations grammaticales relatives aux mots, comme la catégorie morphosyntaxique (verbe, nom…), le nombre, le genre, le temps, le mode, etc., seront attachées aux occurrences de façon analogue.

Le dernier type d'annotations concerne les structures (arbres) produites par l'analyseur. Pour délimiter les groupes syntagmatiques, on pourra utiliser des parenthèses spéciales, comme `{&rel…}` pour une proposition relative[22], ou simplement `{…}` si la catégorie syntagmatique n'est pas connue ou trop ésotérique pour les utilisateurs naïfs (par exemple, "groupe adjectival" ou "groupe cardinal"). Pour représenter un lien anaphorique, on attachera au pronom une copie de son référent. En cas d'élision, on rajoutera des occurrences "cachées" `(centrale .&eld=inertielle).`

D'autres informations grammaticales et sémantiques peuvent être attachées aux terminaux et aux non-terminaux. Par exemple, "Devant cette somme, il ne rend pas sa glace" pourrait avoir la représentation intermédiaire suivante (avant d'avoir fini la désambiguïsation) :

```
{ {&grd,cause *devant.&vrb cette somme.2&nf , } il ne rend.2 pas=ne sa glace.1 }
```

qui lèverait l'ambiguïté sur "devant" (verbe/préposition) et "rendre" (vomir/restituer/faire devenir).

Le point principal est que *la vue du système d'annotations dont dispose l'auteur concerne tous les niveaux de description linguistique, mais doit être incomplète à chaque niveau*, parce qu'aucune notion non familière ne doit apparaître. Par exemple, "verbe" est une notion familière pour presque tout adulte instruit, mais pas "verbe modal". Au niveau des fonctions syntaxiques, "sujet", objet" et "complément" sont familiers, mais sans doute pas "attribut", "épithète", "tête" (ou "gouverneur"). Il en va de même au niveau des cas profonds. Prenons par exemple les trois phrases suivantes.

```
Un processus de traduction en ARIANE-G5 se compose d'une suite de trois étapes
(analyse, transfert et génération). Chaque étape est constituée d'une suite de
différentes phases de traitement. Chaque phase est relative à l'emploi d'un LSPL
précis.
```

Voici une vue simplifiée de la structure arborescente obtenue après la préédition optionnelle, l'analyse et la désambiguïsation interactive. On verra plus loin une vue légèrement différente.

```
{ { *UN.&art PROCESSUS.&n,suj { DE.&prep TRADUCTION.&n,comp { EN.&prep **ARIANE-
G5.&np,comp } } } SE.&refl COMPOSE.&v,phvb { D'UNE.&art SUITE.&n,obj1 { DE.&prep
TROIS.&card E!1TAPES.&n,comp { (.&lp ANALYSE.&n,app ,.&ponc TRANSFERT.&n,coord
ET.&cjcoord GE!1NE!1RATION.&n,coord ).&rp } } } ..&ponc } { { *CHAQUE.&art
E!1TAPE.&n,suj } EST.&v,aux CONSTITUE!1E.&v,phvb { D'.&prep UNE.&art
SUITE.&n,comp { DE.&prep { DIFFE!1RENTES.&adj,epit } PHASES.&n,comp { DE.&prep
TRAITEMENT.&n,comp } } } ..&ponc } { { *CHAQUE.&art PHASE.&n,suj } EST.&v,phvb {
RELATIVE.&adj,atsubj { A!2.&prep L'.&art EMPLOI.&n,obj1 { D'.&prep UN.&art
**LSPL.&np,comp { PRE!1CIS.&adj,epit } } } } ..&ponc }
```

Ces questions de codage interne ne sont pas secondaires, comme on le pense trop souvent. Elles sont importantes en TAO classique, et encore plus en TAFD, puisqu'il faut qu'un utilisateur non-spécialiste puisse y accéder (éventuellement à travers une vue simplifiée). C'est d'ailleurs un défi pour les linguistes d'organiser les informations linguistiques pour permettre de telles vues.

[22] Ou bien on ajoute "rel" à l'information attachée à sa tête, comme dans l'exemple suivant ("compose.&v,phvb").

*c. Vers des documents "auto-explicatifs"*

Il y a d'autres utilisations potentielles aux formes annotées que nous proposons. Premièrement, les annotations correspondantes peut être facilement générées dans chaque langue cible. Cela répond à l'une des objections à la TAFD, à savoir qu'il pourrait être impossible de générer des phrases ou des fragment de texte reflétant exactement les choix de désambiguïsation de l'auteur.

Dans un contexte plus général, en conservant la forme annotée d'un document, on obtiendrait un document "auto-explicatif". Les lecteurs pourraient demander le sens de n'importe quel fragment du document. Cela serait extrêmement intéressant dans toutes les situations où l'on veut obtenir des textes non-ambigus, ou les moins ambigus possibles, et où la seule solution actuelle est l'emploi — oh combien délicat et difficile à imposer !— de langages contrôlés. En effet, on pourrait ainsi produire des documents *complètement désambiguïsés*, ce qui est très difficile, même avec des langages contrôlés très contraignants, et ce en laissant à l'auteur toute sa liberté d'expression.

### I.3 Aspects ergonomiques

Les aspects ergonomiques sont bien sûr importants en TAO classique. En TAO du veilleur ou du réviseur, il y a ainsi une notion de qualité *apparente,* liée à la présentation. La même traduction paraît bien meilleure si elle est formatée comme un texte que si elle est présentée phrase par phrase. De même, un réviseur accepte plus volontiers des choix portant sur de petits groupes de mots que sur des phrases complètes. Les constructeurs de systèmes ont aussi beaucoup travaillé sur les éditeurs bilingues, sur les outils de paramétrage, et sur les aides à la création de dictionnaires.

En TAFD, l'ergonomie est un aspect encore plus crucial, et les choix ergonomiques influencent directement l'architecture de tout le système. Les choix principaux sont les suivants :

- Le système doit-il tourner sur des ordinateurs personnels bon marché ?
- Doit-il fonctionner en temps réel, ou l'asynchronisme est-il préférable ?
- Une architecture avec serveur(s) est-elle possible ?
- Comment les dialogues doivent-ils être organisés ? Est-il nécessaire et/ou possible de les conduire dans un environnement multimédia et multimodal ? Par exemple, peut-on améliorer l'efficacité et la convivialité des dialogues de désambiguïsation grâce à l'utilisation de synthèse de parole et d'interactions graphiques ?

Si l'on vise vraiment la "TAO pour tous", il est impératif de pouvoir utiliser les systèmes de TAFD sur des des ordinateurs personnels, ou sur des stations de travail de bas de gamme. Pour les autres questions, nous répondons : asynchronisme, distribution, interaction non-préemptive.

#### 3.1. Asynchronisme

Nous sommes partisans d'une organisation asynchrone, analogue à celle de CRITIQUE [30], pour des raisons ergonomiques et pratiques. D'abord, une organisation en temps réel serait nécessaire seulement si nous souhaitions que l'utilisateur réponde immédiatement aux questions posées par le système, ce que nous ne voulons pas. Ensuite, il serait contre-productif de poser au rédacteur des questions "à la volée"sur un texte non achevé. Enfin, si l'on prend en compte le degré de complexité inhérent à n'importe quel système de TAFD général et de bonne qualité, nous ne pouvons espérer ni une exécution en temps réel, ni une implantation des ressources nécessaires (base lexicale, linguiciel) sur le genre de matériel envisagé.

#### 3.2. Architecture distribuée

Une exécution en temps réel n'est pas non plus possible dans le cadre de l'architecture distribuée que nous préconisons, car (a) l'utilisation d'un serveur puissant pour prendre en charge tout ou partie des traitements non-interactifs de traduction est préférable au stockage de gros systèmes sur chaque station de travail, même si l'on envisage une exécution en tâche de fond ; et (b) on peut ainsi mieux résoudre les problèmes de maintenance des ressources linguistiques.

#### 3.3. Désambiguïsation interactive multimodale non-préemptive

Il nous paraît essentiel de permettre à l'auteur de décider quand il souhaite entamer le dialogue avec le système. En d'autres termes, le système propose et l'utilisateur dispose. En effet, les précédents essais en TAFD nous semblent avoir échoué en partie à cause du caractère modal des dialogues de désambiguïsation.

## II. LIDIA-1.0, une maquette illustrative

La maquette LIDIA-1.0 est un premier essai de réalisation des idées exposées plus haut. Nous avons choisi HyperCard comme environnement de rédaction. La maquette permet de traduire vers

l'angais, l'allemand, et le russe une pile en français qui présente, en contexte, des phrases comportant les ambiguïtés que nous avons choisi de traiter.

## II.1 Contexte d'utilisation de la maquette

### 1.1. Pile de démonstration

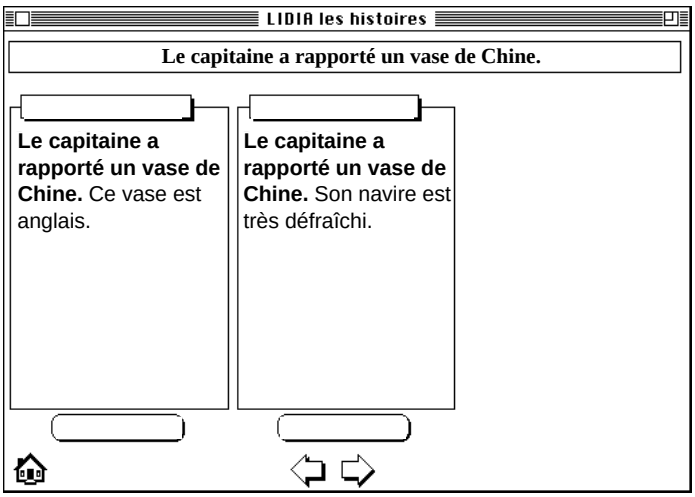


*Figure 1 : une carte d'histoires*

HyperCard permet de construire une pile de façon interactive et incrémentale. Une pile est constituée de cartes. Sur une carte, on peut placer deux types d'objets, des champs et des boutons. Les champs contiennnent du texte. Les boutons sont des éléments actifs sur lesquels on "clique" avec la "souris" pour déclencher des actions.

Notre pile de démonstration, intitulée "LIDIA les histoires", contient deux types de cartes, des cartes d'histoires et des cartes de phrase.

Une carte d'histoires rassemble plusieurs histoires qui partagent une même phrase, ambiguë hors contexte. La ou les ambiguïtés sont solubles par un humain, qui comprend le contexte global, mais ne le seraient pas par un système expert.

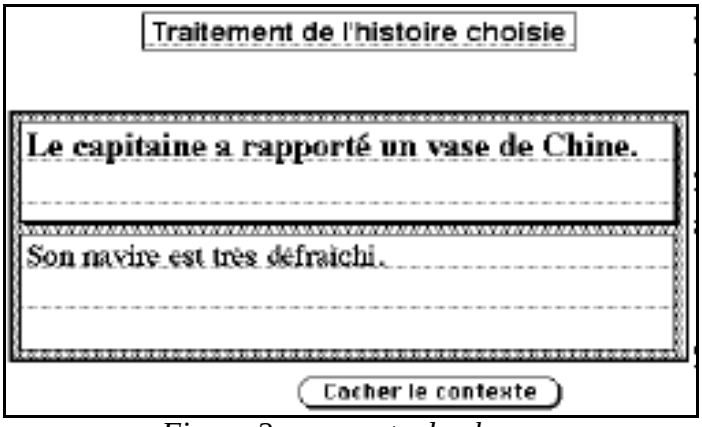


*Figure 2: une carte de phrase*

Pour les besoins de la démonstration, chaque histoire est présentée dans une carte de phrase sur laquelle le contexte de la phrase ambiguë peut être caché ou non.

Pour que les histoires soient traduites, on demande la traduction des champs des cartes de phrases.

Comme souhaité plus haut, on n'est jamais interrompu par une question. Les objets se contentent d'indiquer qu'ils attendent l'intervention de l'utilisateur afin que la traduction se poursuive, et l'utilisateur décide quand intervenir et à quelle question répondre.

### 1.2. Outils d'interaction

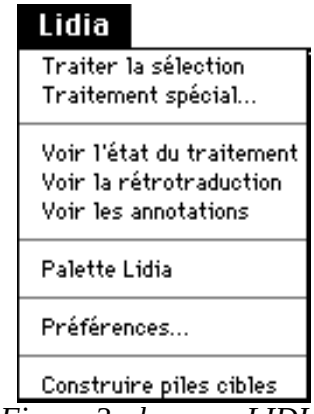


*Figure 3 : le menu LIDIA*

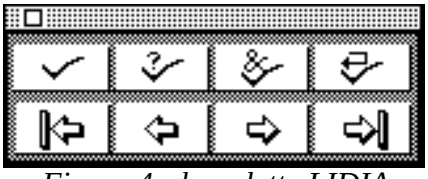

*Figure 4 : la palette LIDIA*

Lorsque les préférences ont été définies, l'utilisateur dispose de menus et d'une palette pour interagir avec LIDIA.

L'interaction se fait au moyen du menu **LIDIA** (fig. 3), du menu **Messages**, d'une palette (fig. 4), de boutons de rétro-action (fig. 6 et 8) et de fenêtres flottantes (fig. 7).

Le menu ci-contre montre les huit actions possibles. L'utilisateur peut aussi accéder aux traitements les plus fréquents avec une palette flottante (fig. 4).

Les outils de la première ligne sont les outils LIDIA (sélectionner un objet à traiter, voir l'état d'avancement des traitements, voir les annotations, voir les rétrotraductions). Les outils de la seconde ligne permettent de se déplacer dans la pile courante.

## II.2 Démonstration

Choisissons (✓), l'outil de sélection LIDIA, et sélectionnons un objet à traduire (fig. 5), ici la phrase : "le capitaine a rapporté un vase de Chine".

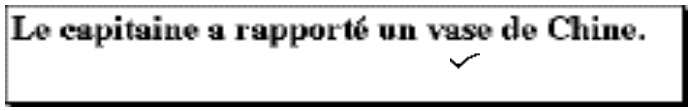


*Figure 5 : selection d'un objet*

Un bouton de suivi apparaît sur l'objet sélectionné.

Cliquons sur ce bouton (fig 6). Une fenêtre flottante apparaît (fig. 7).


*Figure 6 : l'utilisateur demande l'état d'avancement des traitements*

Le traitement en cours est distingué en gras, les traitements déjà effectués sont en style normal, et les traitements à venir sont en italiques.

Ici, le système est en train d'effectuer l'analyse de la phrase choisie.


*Figure 7 : fenêtre flottante pour l'état des traitements*

Si l'analyse produit des ambiguïtés, le système nous prévient qu'il va falloir l'aider à les lever, en faisant apparaître :

- un nouvel item dans le menu **Message** ;
- un nouveau bouton sur l'objet concerné, comme dans la figure 8.

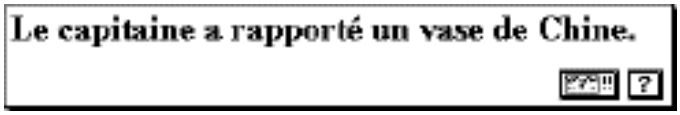


*Figure 8 : l'objet a une ou plusieurs questions à poser*

Nous pouvons choisir de répondre aux questions immédiatement ou plus tard.

Cliquons dès maintenant sur le bouton .

Une première question apparaît (fig. 9), pour déterminer l'attachement du groupe nominal "de Chine". Nous répondons qu'il s'agit d'un vase chinois, et non pas d'un objet rapporté de Chine.

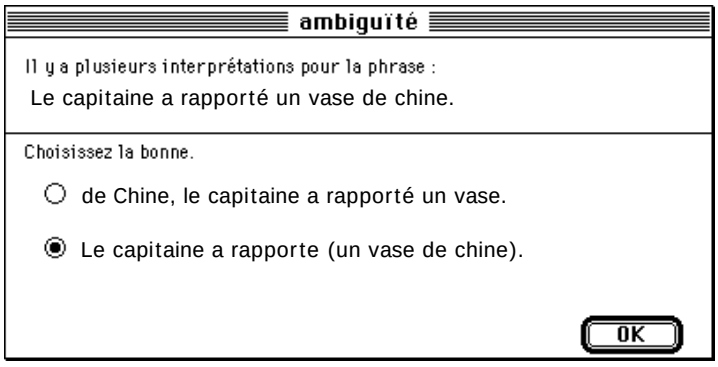


*Figure 9 : problème d'attachement (histoire 2)*

Un second dialogue apparaît (fig. 10), qui nous permet de sélectionner le sens du mot "capitaine".

Les sens des mots qui apparaissent dans ces dialogues proviennent de Parax, une maquette de BDLM [38], également réalisée en HyperCard.

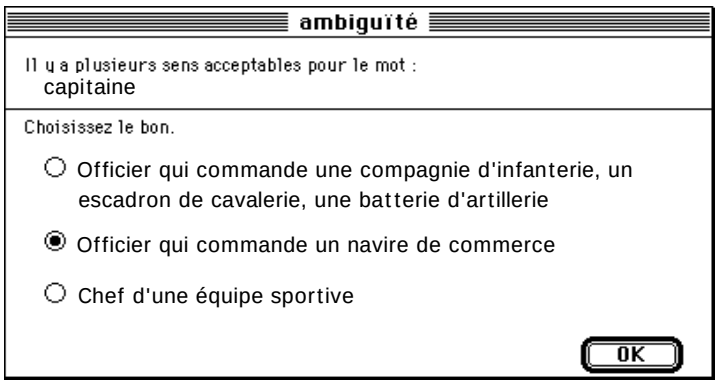


*Figure 10 : sélection du sens de capitaine (histoire 2)*

Le processus de désambiguïsation est terminé pour cette phrase.

Nous pouvons sur demande obtenir une vue de la forme annotée du texte.

Celle qui est produite ici montre, pour l'analyse finalement retenue, les groupes syntagmatiques, accompagnés de leurs fonctions syntaxiques, ainsi que la classe syntaxique de chaque occurrence (fig. 11).

On voit qu'il n'est pas déraisonnable d'imaginer que des utilisateurs expérimentés pourraient insérer eux-mêmes de telles marques, évitant ainsi certains dialogues.

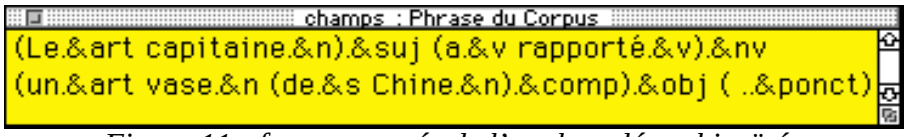


*Figure 11 : forme annotée de l'analyse désambiguïsée*

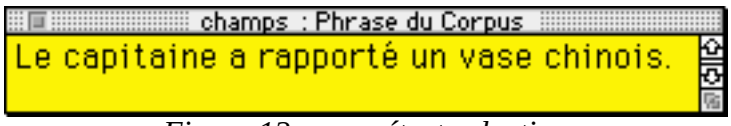


*Figure 12 : une rétrotraduction*


*Figure 13 : traduction des deux histoires en allemand*

Pour vérifier les traductions produites par le système dans chacune des langues cibles, nous pouvons demander une "rétrotraduction".

Pour la seconde interprétation de la phrase de l'exemple et pour l'allemand, nous obtenons le texte de la figure 12.

Finalement, le système produit une carte d'histoires traduites.

### II.3 Processus de clarification

Le processus de désambiguïsation est organisé autour d'un moteur de reconnaissance de patrons [2]. Pour cinq classes d'ambiguïtés parmi les huit que nous avons définies, nous utilisons des schémas de patrons.

Un schéma de patrons est un ensemble de patrons qui partagent un ensemble de variables.

Un schéma de patrons est reconnu sur une phrase (nécessairement ambiguë) lorsque chacun de ses patrons se superpose à au moins une des analyses de la phrase, que toutes les analyses sont couvertes, et que les variables partagées par plusieurs patrons s'instancient chacune sur le même segment de phrase. La figure 14 illustre le processus de filtrage.

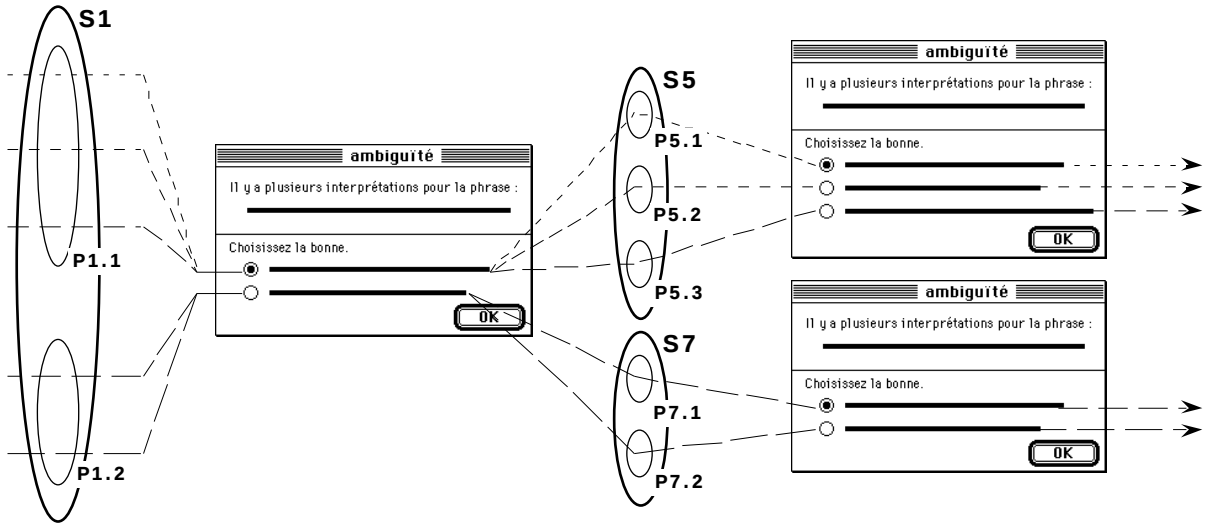


*Figure 14 : processus de filtrage avec des schémas de patrons*

Trois analyses sont filtrées par le patron P1.1 du schéma S1 alors que les deux autres sont filtrées par le patron P1.2. On peut alors construire un dialogue avec deux items, le premier permettant de choisir les trois premières analyses, le second, les deux autres. Pour chacune des deux familles ainsi créées, il faut trouver un schéma de patrons qui permette de choisir une analyse. C'est le cas avec les schéma S5 et S7.

Une méthode de construction de paraphrase est associée à chaque patron. Ces méthodes sont construites au moyen d'un ensemble de treize opérateurs de base.

Par exemple, la figure 15 montre les deux arbres produits lors de l'analyse de la phrase "Le capitaine a rapporté un vase de Chine.".

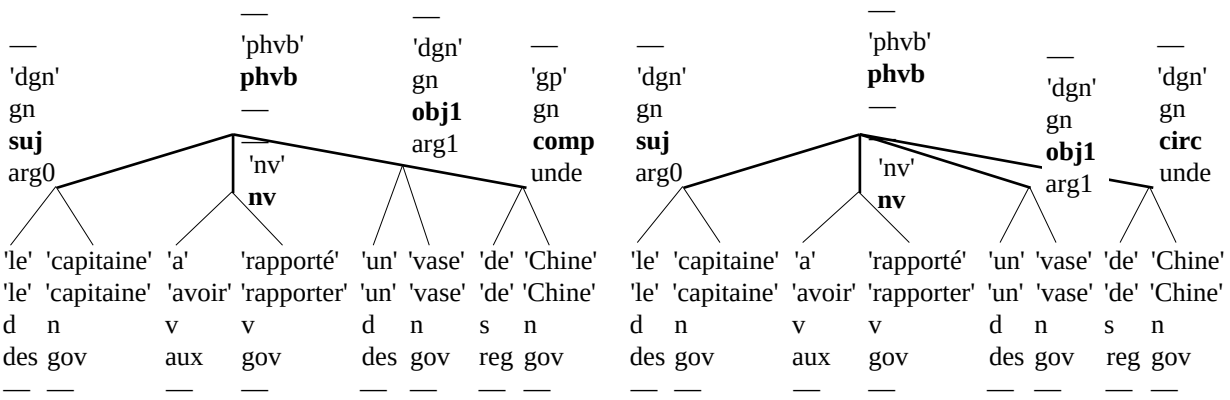


*Figure 15 : structure multisolution, multiniveau et concrète pour la phrase "le capitaine a rapporté un vase de Chine"*

Les deux patrons (Patron 12 & Patron 13) utilisés pour reconnaître l'ambiguïté sont présentés dans la figure 16.

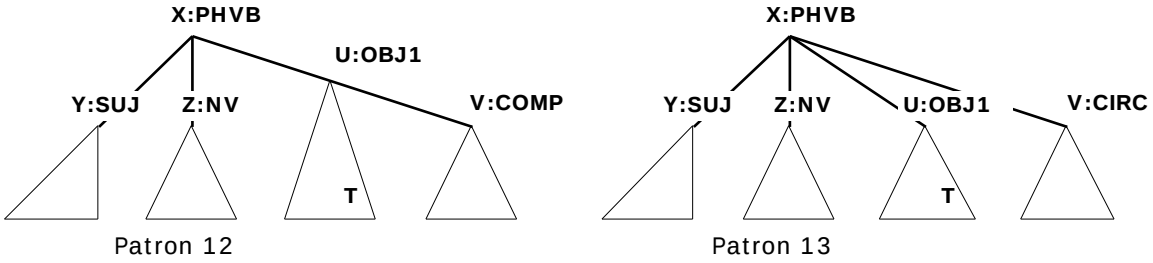


*Figure 16 : deux patrons d'ambiguïté*

La méthode associée au patron 12 est :

- Texte(Y) Texte(Z) Parenthèse(Texte(T), Texte(V))

qui permet de produire le texte suivant :

- Le capitaine a rapporté (un vase de Chine.)

La méthode associée au patron 13 est :

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

qui permet de produire le texte suivant :

- de Chine, le capitaine a rapporté un vase.

# III. Perspectives

## III.1 Bilan de la maquette LIDIA-1.0

Cette maquette a été développée avec des moyens limités, et il était évidemment impossible de réaliser tous les aspects de l'architecture proposée pour la TAFD en général, et encore moins de construire un système à grande couverture lexicale. Cependant, il nous a paru essentiel de prendre en compte tous les aspects, sinon dans l'implémentation, au moins dans la spécification. En effet, il paraissait *a priori* clair, et l'expérience l'a confirmé, que des objectifs ergonomiques pourraient contraindre certains choix informatiques et linguistiques, et réciproquement pour les objectifs informatiques et linguistiques. Les travaux d'autres groupes de recherche nous avaient d'ailleurs bien montré qu'on ne peut pas, par exemple, rajouter ou modifier *a posteriori* un aspect essentiel, comme la convivialité des dialogues, l'architecture distribuée, ou la conception interlingue.

Nous avons donc travaillé "en largeur" sur la plupart des aspects, et "en profondeur" sur quelques-uns seulement, comme le processus de désambiguïsation et l'architecture logicielle. Parmi les aspects non implémentés dans LIDIA-1.0, certains posent encore des problèmes de fond, tandis que d'autres seraient facilement intégrables. Nous sommes d'ailleurs en train de compléter cette maquette, pour obtenir un démonstrateur (LIDIA-1.1) plus complet d'ici fin 1994.

### 1.1. Ce qui n'a pas été abordé

Dans notre scénario, il y a deux phases interactives, la standardisation (ou "normalisation") et la clarification (ou "désambiguïsation"). La première a seulement été évoquée plus haut, et nous ne l'avons pas intégrée à LIDIA-1.0, où le texte d'entrée est supposé standardisé, c'est-à-dire que :

- tous les mots doivent être connus du système, et correctement orthographiés.
- les tournures et syntagmes figés spéciaux doivent avoir été identifiés. Sinon, par exemple, "Activer les bulles d'aides" ne pourrait pas être traduit en anglais par "Show balloons".

– la terminologie utilisée doit respecter les préférences lexicales courantes[23].

– les champs textuels et les boutons doivent être étiquetés par leur type de texte (style d'énoncé s'il s'agit d'un seul énoncé, genre de texte sinon). C'est ce point qui reste à approfondir.

### 1.2. Évolution de la maquette

Nous pensons intéressant de compléter la maquette, pour arriver d'ici quelques mois à une version 1.1, plus complète, et permettant de premières expériences d'acceptabilité.

En ce qui concerne la standardisation, nous comptons utiliser Le correcteur 101™ de Machina Sapiens, qui est en train d'en produire une version intégrable par des développeurs.

En clarification, le mécanisme de reconnaissance des schémas de patrons a encore quelques problèmes, et doit être amélioré pour qu'on puisse traiter toutes les ambiguïtés considérées.

Pour l'instant, le serveur de communication ne peut servir qu'une station LIDIA à la fois. Il devrait pouvoir être partagé par plusieurs. Cette extension ne devrait pas être très difficile à réaliser.

Il reste enfin à améliorer certaines parties du logiciel de base, comme l'accès à la BDLM Parax pour les dialogues de clarification lexicale, et la production des "formes Lisp" des arbres d'analyse produits par Ariane-G5 (formes utilisées par LIDIA pour produire les dialogues de clarification).

### 1.3. Ce qui est illustratif d'un vrai système

L'architecture logicielle et ergonomique de la maquette LIDIA-1.0 a été conçue autour de trois mots-clés : distribution, extensibilité, intégration. Telle quelle, il nous semble qu'elle pourrait être reprise comme base de systèmes de TAFD réels. En effet,

- l'architecture distribuée, par clients et serveurs, marche bien et est de plus en plus répandue.
- l'extensibilité est due à l'usage de la programmation "par objets", qui n'est pas, bien au contraire, limitée à des maquettes ou à des prototypes.
- l'intégration du système de TAFD dans l'environnement d'édition habituel (ici, HyperCard) est une très bonne solution. L'utilisateur n'a pas à se former à un nouveau logiciel et l'accès au traitement de TAFD se fait en utilisant les outils d'interaction familiers.

Du point de vue linguistique, il faut insister sur le fait que l'implémentation, bien que limitée, n'est pas du tout *ad hoc*.

## III.2 Vers un prototype utilisable expérimentalement d'ici l'an 2000

Après cette première expérience, notre nouvel objectif pour 1995-99 est de réaliser un prototype utilisable en interne, par exemple dans le cadre de l'IMAG, qui, avec ses 750 à 800 chercheurs ou assimilés, fournirait un bon terrain d'expérience.

L'idée serait de se limiter à des textes dont la traduction multilingue présente un intérêt réel, et qui n'exigent pas une couverture lexicale énorme. Nous envisageons actuellement de traiter des résumés, des transparents, et peut-être de brèves plaquettes de présentation, qu'il est agréable et utile de pouvoir présenter dans d'autres langues que le français et l'anglais.

Pour passer à un prototype, nous prévoyons de revoir l'architecture logicielle, en particulier en utilisant à la place d'HyperCard Grif, un éditeur puissant et programmable, de développer certains aspects linguistiques et ergonomiques, et enfin de progresser dans les aspects non traités.

### 2.1. Aspects informatiques

HyperCard ne permet pas de structurer finement un texte comprenant plusieurs énoncés, ni de déclencher facilement des actions spécifiques à un type de fragment textuel. C'est pourquoi nous pensons utiliser Grif [33], un éditeur de documents structurés compatible avec SGML, comme outil d'édition linguistique, en l'interfaçant avec les outils utilisés par les auteurs considérés pour produire leurs résumés, transparents et plaquettes (Word™, PowerPoint™, …).

La programmation linguistique de la maquette a été réalisée avec les outils d'Ariane-G5, initialement conçus pour la programmation heuristique, dans le cadre d'une approche par "sous-optimisation". Pour écrire des analyseurs de couverture plus large, et implémenter efficacement le traitement de multiples genres de texte et styles d'énoncé, il sera indispensable et beaucoup plus efficace d'utiliser des outils plus combinatoires, permettant la "programmation ambiguë", outils auxquels nous travaillons dans le cadre du projet Sambre [26].

---

[23] Ces préférences peuvent être imposées par des habitudes locales d'écriture, par le domaine lui-même, ou par une volonté de standardisation — dans une entreprise par exemple.

Enfin, le processus de désambiguïsation que nous avons implémenté dans la maquette est figé et ne comporte qu'un niveau de dialogue. Au niveau d'un prototype, il faut pouvoir expérimenter d'autres stratégies, de nouveaux opérateurs et de nouvelles méthodes de désambiguïsation. Nous prévoyons donc de développer un environnement de description de processus de désambiguïsation, directement utilisable par des développeurs linguistes non-informaticiens. Cet environnement devrait comporter trois composants, permettant de décrire les patrons, les méthodes de production de dialogues, et les stratégies de désambiguïsation.

### 2.2. Aspects linguistiques

Étant donné la limitation des types de textes envisagés, les connaissances grammaticales et lexicales à développer resteront limitées. Dans chaque langue, la couverture lexicale ne devrait être que de 5000 à 7000 termes, ce qui représente tout de même un effort de développement significatif au niveau d'un laboratoire. Cette partie du travail devrait converger avec le projet NADIA [38] de BDLM par acceptions multilingues.

C'est la partie concernant le typage des fragments de texte en genres de texte et styles d'énoncé qui a le moins progressé jusqu'ici. Denos [18] a cependant apporté quelques éléments qui devraient permettre de produire un outil de typage textuel utilisable par des auteurs non spécialistes.

Dans le cadre d'un projet de recherche commun avec ATR-ITL (Japon), nous avons commencé à réunir un corpus qui doit nous permettre d'étudier sur des textes réels le type et la fréquence des ambiguïtés, ainsi que leur importance pour la traduction, et l'acceptabilité de diverses méthodes de désambiguïsation interactive. Tout n'étant pas possible dans le cadre d'un prototype, cela permettra de faire des choix éclairés.

### 2.3. Aspects ergonomiques

Dans LIDIA-1, nous avons pu mettre en œuvre les principes d'asynchronisme et de non-préemptivité. Par contre, les dialogues ne sont pas multimodaux.

Dans le prototype, il nous semble important d'offrir à l'auteur le choix entre plusieurs formes de dialogue. Par exemple, on pourrait présenter la géométrie de l'arbre d'analyse la plus probable, sous une forme indentée analogue au mode "plan" de Word™, et l'auteur pourrait la manipuler jusqu'à obtenir celle qu'il a en tête. Ou encore, au lieu de poser des questions par réarrangement et présentation linéaire des mots de l'énoncé source, on pourrait produire des tests, comme le test de la négation pour des ambiguïtés sur la distinction thème/rhème ("il boit du petit lait" —> "il ne boit pas du/de petit lait"). L'utilisation de la synthèse de parole serait aussi une possibilité à étudier [7].

Le prototype devrait aussi offrir un premier niveau d'apprentissage, ou plutôt d'autoréglage, qui permettrait de proposer comme réponses par défaut les réponses les plus fréquentes dans le contexte courant.

## III.3 Quelques éléments prospectifs sur la construction de systèmes "tout public"

Quand le grand public disposera-t-il de systèmes de "TAFD pour tous" ? Sachant qu'il faudrait des investissements très importants, et en particulier le développement de très grandes bases de données lexicales multilingues, nous ne pouvons que chercher à démontrer et à convaincre, et en attendant pousser l'étude autant que faire se peut au niveau de la recherche publique.

Démontrer, c'est pour nous arriver à construire et expérimenter le prototype dont il a été question plus haut. Pousser l'étude, ce serait en outre construire un simulateur, utilisant un ou plusieurs "magiciens d'Oz" (compères humains), pour tester un certain nombre d'hypothèses sur l'acceptabilité de différents types de désambiguïsation interactive multimodale.

En effet, est-il bien certain qu'un auteur acceptera de passer 1h30 par page pour normaliser et clarifier son texte, même s'il sait que le traduire dans une seule langue qu'il connaît très bien lui prendrait au moins autant de temps, et même si c'est un maximum rarement atteint ? Ce maximum, que nous nous sommes fixé arbitrairement au départ en fonction du temps que nous passons nous-mêmes à traduire nos documents, représenterait au total environ 10mn d'interaction par phrase, soit une trentaine de questions… Bien sûr, nécessité fait loi, mais un bon simulateur permettrait, comme pour les avions et les centrales nucléaires, d'éliminer *a priori* un certain nombre de causes d'échec.

## Remerciements

Cette recherche a été menée sous l'égide de l'Université Joseph Fourier et du CNRS. Ont contribué à LIDIA-1.0 : J.-Ph. Guilbaud, N. Nédobejkine, E. Blanc, M. Axtmeyer et D. Levenbach, pour le linguiciel ; P. Guillaume, M. Quézel-Ambrunaz, G. Sérasset et M. Lafourcade, pour le logiciel. Merci également à F. Peccoud, inspirateur de tous les développements en direction de l'hypermédia et du multimodal au GETA.

## Références bibliographiques


[1] **Blanchon H. (1990)** *LIDIA-1 : Un prototype de TAO personnelle pour rédacteur unilingue.* Proc. X-èmes Journées sur les systèmes experts et leurs applications. Conférence spécialisée “Le traitement automatique des langues naturelles et ses applications”, Avignon, 28 mai-1 juin 1990, EC2, 51-60.

[2] **Blanchon H. (1992)** *A Solution to the Problem of Interactive Disambiguation.* Proc. COLING-92, Nantes, 23-28 July 1992, C. Boitet, ed., vol. 4/4, 1233-1238.

[3] **Blanchon H., Guilbaud J. P. & Nédobejkine N. (1992)** *LIDIA : the disambiguation process — le processus de désambiguïsation.* Exposition COLING-92, Nantes, 23-28 juillet 1992, Pile HyperCard.

[4] **Boitet C. (1988)** *Hybrid Pivots using m-structures for multilingual Transfer-Based MT Systems.* Jap. Inst. of Electr., Inf. & Comm. Eng., June 1988, NLC88-3, 17—22.

[5] **Boitet C. (1988)** *PROs and CONs of the pivot and transfer approaches in multilingual Machine Translation.* Proc. Int. Conf. on “New directions in Machine Translation”, Budapest, 18–19 August 1988, BSO, 13 p.

[6] **Boitet C. (1989)** *Motivation and Architecture of the LIDIA Project.* Proc. MTS-II (MT Summit), Munich, 16-18 août 1989, 12 p.

[7] **Boitet C. (1989)** *Speech Synthesis and Dialogue Based Machine Translation.* Proc. ATR Symp. on Basic Research for Telephone Interpretation, Kyoto, December 1989, 12 p.

[8] **Boitet C. (1990)** *Software and lingware engineering in recent (1980—90) classical MT : Ariane-G5 and BV/aero/F-E.* Proc. ROCLing-III (tutorials), Taipeh, Aug. 1990, 21 p.

[9] **Boitet C. (1990)** *Towards Personal MT : on some aspects of the LIDIA project.* Proc. COLING-90, Helsinki, 20-25 août 1990, H. Karlgren, ed., ACL, vol. 3/3, 30-35.

[10] **Boitet C. (1993)** *La TAO comme technologie scientifique : le cas de la TA fondée sur le dialogue.* In “Études et Recherches en Traductique”, A. Clas & P. Bouillon, ed., Presses de l’Université de Montréal, Montréal, 32 p.

[11] **Boitet C. & Blanchon H. (1993)** *Dialogue-Based MT for Monolingual Authors and the LIDIA project.* Proc. NLPRS’93 (Natural Language Processing Rim Symposium, Fukuoka, 6-7/12/93, H. Nomura, ed., Kyushu Institute of Technology, 208—222.

[12] **Boitet C. & Zaharin Y. (1988)** *Representation trees and string-tree correspondences.* Proc. COLING-88, Budapest, 22–27 Aug. 1988, ACL, 59—64.

[13] **Brown R. D. (1989)** *Augmentation.* Machine Translation, 4, 1299-1347.

[14] **Brown R. D. & Nirenburg S. (1990)** *Human-Computer Interaction for Semantic Disambiguation.* Proc. COLING-90, Helsinki, 20-25 août 1990, H. Karlgren, ed., ACL, vol. 3/3, 42-47.

[15] **Carbonnell J. G. & Tomita M. (1985)** *New Approaches to Machine Translation.* Proc. TMI-85 (Conf. on Theoretical and Methodological Issues in Machine Translation of Natural Languages), Hamilton, N.Y., 14-16 Aug. 1985, S. Nirenburg, ed., 59-74.

[16] **Chandioux J. (1988)** *10 ans de METEO (MD).* In “Traduction Assistée par Ordinateur. Actes du séminaire international sur la TAO et dossiers complémentaires”, A. Abbou, ed., Observatoire des Industries de la Langue (OFIL), Paris, mars 1988, 169—173.

[17] **Chandler B., Holden N., Horsfall H., Pollard E. & McGee Wood M. (1987)** *N-tran Final Report.* Alvey Project, 87/9, CCL/UMIST, Manchester.

[18] **Denos N. (1993)** *Typage interactif de fragments de textes en “Styles d’Enoncé” et “Genres de Texte”.* Rapport de DEA d’informatique, GETA, IMAG, UJF&INPG, juin 1993.

[19] **Heidorn G. E., Jensen K. & Miller L. A. (1982)** *The EPISTLE Text-Critiquing System.* IBM System Journal, **21**/1, 305-326.

[20] **Huang X. M. (1990)** *A Machine Translation System for the Target Language Inexpert.* Proc. COLING-90, Helsinki, 20-25 Aug. 1990, H. Karlgren, ed., ACL, vol. 3/3, 364-367.

[21] **Hutchins W. J. (1986)** *Machine Translation: Past, Present, Future.* E. Horwood, ed., John Wiley & Sons, Chichester, England, 382 p p.

[22] **JEIDA (1989)** *A Japanese view of Machine Translation in light of the considerations and recommendations reported by ALPAC, USA.* Japanese Electronic Industry Development Association, Tokyo, 197 p.

[23] **Kay M. (1973)** *The MIND system.* In “Courant Computer Science Symposium 8: Natural Language Processing”, R. Rustin, ed., Algorithmics Press, Inc., New York, 155-188.

[24] **Kittredge R. (1983)** *Sublanguage — Specific Computer Aids to Translation — a survey of the most promising application areas.* Contract n° 2-5273, Université de Montréal et Bureau des Traductions, mars 1983, 95 p.

[25] **Kittredge R. (1986)** *Analyzing Language in Restricted Domains.* In “Sublanguage Description and Processing”, R. Grishman & R. Kittredge, ed., Lawrence Erlbaum, Hillsdale, New-Jersey.

[26] **Lafourcade M. (1993)** *Projet SAMBRE : génie logiciel pour le génie linguiciel.* GETA, IMAG, déc. 93.

[27] **Maruyama H., Watanabe H. & Ogino S. (1990)** *An Interactive Japanese Parser for Machine Translation.* Proc. COLING-90, Helsinki, 20-25 août 1990, H. Karlgren, ed., ACL, vol. 2/3, 257-262.

[28] **Melby A. K. (1981)** *Translators and Machines - Can they cooperate ?* META, **26**/1, 23-34.

[29] **Melby A. K. (1982)** *Multi-Level Translation Aids in a Distributed System.* Proc. COLING-82, Prague, 5-10 juillet 1982, vol. 1/2, 215-220.

[30] **Melby A. K., Smith M. R. & Perterson J. (1980)** *ITS : An Interactive Translation System.* Proc. COLING-80, Tokyo, 30 septembre-4 octobre 1980, M. Nagao, ed., 424-429.

[31] **Nirenburg S. (1989)** *Knowledge-based Machine Translation.* Machine Translation, 4, 5-24.

[32] **Nyberg E. H. & Mitamura T. (1992)** *The KANT system: Fast, Accurate, High-Quality Translation in Practical Domains.* Proc. COLING-92, Nantes, 23-28 July 92, C. Boitet, ed., ACL, vol. 3/4, 1069—1073.

[33] **Quint V., Vatton I., André J. & Richy H. (1991)** *Grif et l'édition de documents structurés : nouveaux développements.* Cahiers GUTemberg, 9, juillet 91, 49—65.

[34] **Richardson S. D. & Braden-Harder L. C. (1988)** *The experience of developing a large-scale natural language text processing system: CRITIQUE.* Proc. 2nd conference on Applied Natural Language Processing, 1988, 195-202.

[35] **Rimon M., McCord M. C., Schwall U. & Pilar M. (1991)** *Advances in Machine Translation Research in IBM.* Proc. Machine Translation Summit III, Washington, D.C., 1-4 juillet 1991, vol. 1/1, 8 p., 11-18.

[36] **Sadler V. (1989)** *Working with analogical semantics : Disambiguation technics in DLT.* T. Witkam, ed., Distributed Language Translation (BSO/Research), Floris Publications, Dordrecht, Holland, 256 p.

[37] **Seligman M. & Boitet C. (1993)** *A "Whiteboard" Architecture for Automatic Speech Translation.* Proc. International Symposium on Spoken Dialogue, Tokyo, 10–12 November 1993, Waseda University, 4 p.

[38] **Sérasset G. & Blanc É. (1993)** *Une approche par acceptions pour les bases lexicales multilingues.* Proc. IIIèmes journées scientifiques "Traductique-TA-TAO", Montréal, octobre 1993, Presses de l'Université de Montréal, 17 p.

[39] **Sigurdson J. & Greatex R. (1987)** *MT of on-line searches in Japanese Data Bases.* RPI, Lund Univ., 124 p.

[40] **Somers H. L., Tsujii J.-I. & Jones D. (1990)** *Machine Translation without a source text.* Proc. COLING-90, Helsinki, 20-25 Aug. 1990, H. Karlgren, ed., ACL, vol. 3/3, 271-276.

[41] **Tomita M. (1984)** *Disambiguating Grammatically Ambiguous Sentences by Asking.* Proc. COLING-84, Stanford, 2-6 juillet 1984, ACL, 476-480.

[42] **Tomita M. (1986)** *Sentence Disambiguation by asking.* Computers and Translation, **1**/1, 39-51.

[43] **Van Herwijnen E. (1990)** *Practical SGML.* Kluwer, Dordrecht, 307 p.

[44] **Vasconcellos M. & León M. (1988)** *SPANAM and ENGSPAM : Machine Translation at the Pan American Health Organization.* In "Machine Translation systems", J. Slocum, ed., Cambridge Univ. Press, 187—236.

[45] **Vauquois B. (1988)** *BERNARD VAUQUOIS et la TAO, vingt-cinq ans de Traduction Automatique, ANALECTES. BERNARD VAUQUOIS and MT, twenty-five years of MT.* C. Boitet, ed., Ass. Champollion & GETA, Grenoble, 700 p.

[46] **Vauquois B. & Boitet C. (1985)** *Automated translation at Grenoble University.* Comp. Ling., **11**/1, January-March 85, 28—36.

[47] **Vauquois B. & Chappuy S. (1985)** *Static grammars: a formalism for the description of linguistic models.* Proc. TMI-85 (Conf. on theoretical and metholodogical issues in the Machine Translation of natural languages), Colgate Univ., Hamilton, N.Y., Aug. 1985, S. Nirenburg, ed., 298-322.

[48] **Veronis J. & Ide N. (1990)** *Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine-Readable Dictionaries.* Proc. COLING-90, Helsinki, 18—25 Aug. 1990, H. Karlgren, ed., 389—394.

[49] **Weaver A. (1988)** *Two Aspects of Interactive Machine Translation.* In "Technology as Translation Strategy", M. Vasconcellos, ed., State University of New York at Binghamton, Binghamton, 116-123.

[50] **Wehrli E. (1990)** *STS: An Experimental Sentence Translation System.* Proc. COLING-90, Helsinki, 20-25 Aug. 1990, H. Karlgren, ed., ACL, vol. 1/3, 76-78.

[51] **Wehrli E. (1991)** *Pour une approche interactive au problème de la traduction automatique.* Proc. Colloque "L'environnement traductionnel. La station de travail du traducteur de l'an 2001", Mons, 25-27 avril 1991, AUPELF & UREF, 59-68.

[52] **Wehrli E. (1992)** *The IPS System.* Proc. COLING-92, Nantes, 23-28 July 1992, C. Boitet, ed., vol. 3/4, 870-874.

[53] **Whitelock P. J., Wood M. M., Chandler B. J., Holden N. & Horsfall H. J. (1986)** *Strategies for Interactive Machine translation : the experience and implications of the UMIST Japanese project.* Proc. COLING-86, Bonn, 25-29 août 1986, IKS, 25-29.

[54] **Wood M. M. (1989)** *Japanese for speakers of English: The UMIST/Sheffield Machine Translation Project.* In "Recent Developments and Applications of Natural Language Processing", J. Peckham, ed., Kogan Page ltd, London, 56-64.

[55] **Wood M. M. G. & Chandler B. (1988)** *Machine Translation For Monolinguals.* Proc. COLING-88, Budapest, 22-27 Aug. 1988, 760—763.

[56] **Zaharin Y. (1986)** *Strategies and heuristics in the analysis of a natural language in Machine Translation.* Ph.D. thesis, Universiti Sains Malaysia, Penang.

[57] **Zaharin Y. (1990)** *Generation of Synthesis Programs in ROBRA (ARIANE) from String-Tree Correspondence Grammars.* Proc. COLING-90, Helsinki, 18—25 Aug. 1990, H. Karlgren, ed., 425—430.

[58] **Zajac R. (1988)** *Interactive Translation : a new approach.* Proc. COLING-88, Budapest, Aug. 1988.

-o-o-o-o-o-o-o-o-o-o-

# Table des matières



-o-o-o-o-o-o-o-o-o-o-