

THÈSE
présentée par

Hervé BLANCHON

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER – GRENOBLE 1

(ARRÊTÉS MINISTÉRIELS DU 5 JUILLET 1984 ET
DU 30 MARS 1992)

Spécialité

INFORMATIQUE

**LIDIA-1 : UNE PREMIÈRE MAQUETTE
VERS LA TA INTERACTIVE “POUR TOUS”**

21 janvier 1994

Jury :

M. Yves	CHIARAMELLA	Président
M. Guy	PÉRENNOU	Rapporteur
M. Éric	WEHRLI	Rapporteur
M. François	PECCOUD	Examineur
M. Christian	BOITET	Directeur

THÈSE PRÉPARÉE AU SEIN DU LABORATOIRE GETA (IMAG, UJF & CNRS)

*À tous, parents et amis
pour leur soutien et leur affection*

If you cannot — in the long run — tell everyone what
you have been doing, your doing has been worthless.

Erwin Schrödinger

Remerciements

À l'heure où cette étape de mon travail s'achève, et parce qu'ils m'ont permis d'atteindre des rivages plus paisibles, je voudrais remercier un trop grand nombre de personnes pour pouvoir les citer toutes. Que les absents me pardonnent et sachent que je ne les ai pas oubliés pour autant.

Monsieur Yves Chiaramella, professeur à l'Université Joseph Fourier, m'a fait l'honneur de présider le jury.

Messieurs Guy Pérennou, professeur à l'Université Paul Sabatier, et Éric Wehrli, professeur à l'Université de Genève, ont accepté avec bienveillance d'être rapporteurs et m'ont prodigué des remarques pertinentes qui m'ont permis d'améliorer ce document.

Monsieur François Peccoud, professeur à l'Université Pierre Mendès France et directeur du GETA m'a accueilli au sein du laboratoire, s'est constamment intéressé à mon travail et m'a permis de travailler dans les meilleures conditions.

Monsieur Christian Boitet, professeur à l'Université Joseph Fourier, a été un directeur de thèse et un *mentor* remarquable. Il m'a fait partager son enthousiasme et sa rigueur scientifique et m'a donné le goût de diffuser et discuter mes idées et résultats avec la communauté scientifique. Il va sans dire que ce travail lui doit beaucoup.

Ce travail ne serait rien sans l'équipe LIDIA qui a collaboré au projet.

Le linguiciel est le fruit du travail de Jean-Philippe Guilbaud pour l'allemand et l'interface entre le linguiciel et le logiciel LIDIA, Nicolas Nédobekine pour la partie russe, et Étienne Blanc pour l'anglais.

La base de données lexicales Parax maquetée par Étienne Blanc a été complétée par Kiki Levenbach pour l'allemand, Monique Axtmeyer pour l'anglais, et Nicolas Nédobekine pour le russe.

L'accès au linguiciel Ariane-G5 a été spécifié et implémenté par Maurice Quézel-Ambrunaz et Pierre Guillaume, qui ont réalisé le réseau LIDIA. J'ai particulièrement apprécié la patience dont ce dernier a fait preuve, acceptant avec le sourire d'innombrables modifications des spécifications.

Les goûters et les tâches administratives afférentes à la vie d'un thésard voyageur ont été assurés avec sourire et empressement par Nadine, Christine et Martine.

Je tiens à remercier tout le personnel de la médiathèque IMAG, et Kiki Levenbach, remarquable gestionnaire de notre bibliothèque, pour m'avoir aidé à explorer leurs trésors bibliographiques et à accéder d'autres documents.

Plusieurs collègues m'ont envoyé des documents introuvables : Michael Mac Cord, Toon Witkam, Taijiro Tsutsumi, Hiroshi Maruyama, Danit Segev, Dafna Seinwald et Nils Dahlbäck. Hiroshi Maruyama m'a même envoyé de magnifiques diapositives, ici intégrées sous forme électronique.

Je ne voudrais pas oublier Georges Fafiotte, grand amateur de CACAO devant le "Grand Tuteur Intelligent", qui m'a fait paraître moins longues un certain nombre de nuits passées en sa compagnie devant nos écrans.

Je tiens aussi à adresser mes très sincères et profonds remerciements à Mathieu Lafourcade, Jean Gaschler et Gilles Sérasset, qui ont grimpé sur mon navire en pleine tempête et n'ont cessé d'écoper avec ardeur et enthousiasme afin que nous puissions accoster sur des plages enchanteresses et y rencontrer la première incarnation de LIDIA. J'espère pouvoir leur rendre la pareille lorsque leurs voiles seront en vue de la côte et que la fatigue les aura gagnés à la fin de leur longue route.

Je pense enfin à Chantal et à la petite Claire qui non contentes de supporter des absences fréquentes, m'ont toujours encouragé.

Table des Matières

INTRODUCTION	1
---------------------	----------

Partie I De la TA Interactive pour spécialiste à la “TAFD” pour tous

INTRODUCTION	13
CHAPITRE 1 LES EXPÉRIENCES PRÉCÉDENTES	15
1.1. Les pionniers et la TA interactive pour le traducteur.....	15
1.1.1. MIND (1963-73).....	15
1.1.2. ITS (1973-81).....	17
1.1.3. ALPS (1980-).....	23
1.2. L’interaction comme béquille ajoutée a posteriori	25
1.2.1. KBMT-89 (1987-1989).....	26
1.2.2. LMT (1989-).....	36
1.3. L’interaction comme ingrédient de base.....	43
1.3.1. DLT (1982-1989).....	44
1.3.2. N-Tran (1984-1987).....	58
1.3.3. JETS (1985-).....	63
1.3.4. ITS2 (1990-).....	73

CHAPITRE 2	UNE VOIE NOUVELLE : LA TAO INDIVIDUELLE OU "TAFD POUR TOUS"	79
2.1.	Contraintes spécifiques de la TAFD.....	80
2.1.1.	Contraintes matérielles et logicielles.....	80
2.1.2.	Contraintes linguicielles.....	82
2.1.3.	Contraintes ergonomiques.....	82
2.2.	Architecture linguistique et voies intermédiaires nouvelles.....	83
2.2.1.	Transfert multiniveau à acceptions, propriétés, et relations interlingues	83
2.2.2.	Langages guidés	84
2.2.3.	Accessibilité des connaissances	87
2.3.	Situations pour la TAFD	87
2.3.1.	Limitation des paradigmes actuels	88
2.3.2.	Critères de choix de l'approche par dialogue.....	88
2.3.3.	Situations traductionnelles adaptées à la TAFD.....	89
CHAPITRE 3	LIDIA-1, UNE MAQUETTE DE TAFD	91
3.1.	Situation concrète retenue.....	91
3.1.1.	Une langue source, plusieurs langues cibles	91
3.1.2.	Traduction de piles HyperCard de documentation technique	91
3.1.3.	Architecture distribuée et traitements asynchrones.....	92
3.2.	Choix fonctionnels pour le maquetage.....	93
3.2.1.	Utilisation du générateur Ariane-G5 sur IBM.....	94
3.2.2.	Interface et dialogue de désambiguïsation	97
3.2.3.	Organisation des traitements linguistiques.....	97
3.3.	État d'avancement.....	100
3.3.1.	Supports informatiques.....	100
3.3.2.	Linguiciels de TA pure.....	100
3.3.3.	Standardisation et désambiguïsation	100
PARTIE II	LA MAQUETTE LIDIA-1 – VUE EXTERNE	
	INTRODUCTION	103
CHAPITRE 4	UTILISATEUR ET UTILISATION	105
4.1.	Environnement de travail	105
4.1.1.	Configuration logicielle de la station de travail	106
4.1.2.	Accès au système.....	106

4.1.3.	Piles.....	107
4.2.	Personnalisation du système.....	110
4.2.1.	Profil de la tâche	111
4.2.2.	Profil des ressources lexicales.....	112
4.2.3.	Profil des serveurs.....	114
4.2.4.	Profil de l'utilisateur	116
4.3.	Mise en œuvre des traitements.....	119
4.3.1.	Menus.....	119
4.3.2.	Palettes	124
4.3.3.	Rétroaction.....	125
CHAPITRE 5	STANDARDISATION	127
5.1.	Orthographe	127
5.1.1.	L'outil 101 et son intérêt pour LIDIA.....	128
5.1.2.	Scénarios envisagés.....	130
5.1.3.	Correction.....	132
5.2.	Préférences lexicales.....	133
5.3.	Tournures et syntagmes figés spéciaux.....	138
5.4.	Typage.....	141
CHAPITRE 6	CLARIFICATION	147
6.1.	Ambiguïté et traduction automatique : une classification.....	148
6.1.1.	Ambiguïtés spécifiques de l'oral.....	148
6.1.2.	Ambiguïtés de forme.....	149
6.1.3.	Ambiguïtés de structure	150
6.1.4.	Ambiguïtés de relations logiques et sémantiques	151
6.1.5.	Ambiguïtés anaphoriques.....	151
6.1.6.	Autres ambiguïtés	152
6.2.	Ambiguïtés retenues	153
6.2.1.	Corpus	153
6.2.2.	Pile de démonstration - vue externe.....	155
6.2.3.	Pile de démonstration - vue interne.....	157
6.3.	Scénario et dialogues.....	157
6.3.1.	Scénario pour la clarification.....	158
6.3.2.	Classes de dialogue	159
6.3.3.	Types de dialogue	159
	BILAN	165

PARTIE III LA MAQUETTE LIDIA-1 – VUE INTERNE

	INTRODUCTION	169
CHAPITRE 7	ARCHITECTURE	171
7.1.	Station de rédaction.....	171
7.1.1.	HyperCard	171
7.1.2.	Parax.....	173
7.1.3.	Serveur LIDIA.....	176
7.2.	Serveur de communication.....	177
7.2.1.	RemoteMacMainFrame.....	177
7.2.2.	Scripts.....	177
7.2.3.	Serveur.....	177
7.3.	Serveur de traduction.....	178
7.3.1.	Réseau LIDIA/IBM.....	178
7.3.2.	Linguiciel en Ariane-G5.....	184
7.3.3.	Génie linguiciel entre Parax et les dictionnaires Ariane-G5.....	188
CHAPITRE 8	IMPLÉMENTATION	191
8.1.	Serveurs	191
8.1.1.	Superviseur LIDIA.....	192
8.1.2.	Serveur de désambiguïsation.....	193
8.1.3.	Serveur de communication.....	194
8.2.	Communications.....	195
8.2.1.	Apple events	195
8.2.2.	Scripts.....	198
8.2.3.	Messages	199
8.3.	Informations manipulées et codage.....	200
8.3.1.	Besoin de transcriptions	200
8.3.2.	Transcriptions dans la maquette	202
8.3.3.	Fichier miroir.....	203
CHAPITRE 9	DÉSAMBIGUÏSATION	207
9.1.	Méta-classes et classes de problèmes	207
9.1.1.	Polysémie	208
9.1.2.	Ambiguïté de classe syntaxique	208
9.1.3.	Ambiguïté de géométrie	210
9.1.4.	Ambiguïté de décoration	215

9.2.	Idées directrices.....	218
9.2.1.	Principes	218
9.2.2.	Stratégie	218
9.2.3.	Techniques, opérateurs et méthodes	221
9.3.	Mise en œuvre	224
9.3.1.	Filtres et filtrage.....	224
9.3.2.	Méthodes.....	233
9.3.3.	Résultats.....	235
	REMARQUES SUR LES TECHNIQUES D'IMPLÉMENTATION	245
	CONCLUSION	247
	BIBLIOGRAPHIE	255
	INDEX	271
	ANNEXES	277
ANNEXE 1	Phases AM, AX et AY fvx	279
ANNEXE 2	Phase AS1 fvx	283
ANNEXE 3	Phase AS2 fvx	287
ANNEXE 4	Phase AS3 fvx	291
ANNEXE 5	Phase AS5 fvx	295
ANNEXE 6	Phase AS6 fvx	299
ANNEXE 7	Phase GM fvx-lsp	301
ANNEXE 8	Phase AS7 fvx	303
ANNEXE 9	Phase TL fvx-sax	305
ANNEXE 10	Phase TX fvx-sax	307
ANNEXE 11	Phase TS fvx-sax	309
ANNEXE 12	Phase GS1 sax	311
ANNEXE 13	Phase GS2 sax	313
ANNEXE 14	Phase GY sax	315
ANNEXE 15	Phase GM fvx-sax	317
ANNEXE 16	Modèles Linguistiques Ariane pour LIDIA-1 en Chiffres	319

Liste des figures

CHAPITRE 1 LES EXPÉRIENCES PRÉCÉDENTES

Figure 1.1	Schéma général de KBMT-89	27
Figure 1.2	Schéma de l'augmenteur de KBMT-89.....	28
Figure 1.3	Vue éclatée de l'écran de l'augmenteur	29
Figure 1.4	Analyse de la phrase d'entrée avec KBMT-89.....	31
Figure 1.5	Traitement des analyses produites : augmentation automatique	31
Figure 1.6	Affichage initial de quatre menus.....	31
Figure 1.7	Recherche dans le lexique des sens du verbe "set"	32
Figure 1.8	Nouveaux menus après la désambiguïsation du verbe "set"	32
Figure 1.9	Nouveaux menus après la désambiguïsation de "on".....	33
Figure 1.10	Génération d'une paraphrase pour vérification	34
Figure 1.11	Vérification de la paraphrase produite.....	34
Figure 1.12	Affichage de la traduction de la phrase d'entrée	35
Figure 1.13	Une vue de TWS.....	35
Figure 1.14	Structure du système DLT	46
Figure 1.15	Proposition des sens de score supérieur au seuil	48
Figure 1.16	Proposition des sens de score inférieur au seuil	48
Figure 1.17	Organisation des traitements dans N-Tran	59
Figure 1.18	Écran de désambiguïsation avec N-Tran	62
Figure 1.19	Écran de sélection de la phrase à traduire avec JETS.....	66
Figure 1.20	Début de la désambiguïsation interactive avec JETS.....	67
Figure 1.21	Sélection d'un nouveau groupe	67
Figure 1.22	Sélection d'une première contrainte	68
Figure 1.23	Résultat de l'application de la première contrainte	68
Figure 1.24	Sélection d'un nouveau groupe pour choisir une nouvelle contrainte....	69
Figure 1.25	Sélection d'une seconde contrainte	69

Figure 1.26	Résultat de l'application de la seconde contrainte.....	70
Figure 1.27	Sélection d'un nouveau groupe pour faire un traitement manuel.....	70
Figure 1.28	Sélection manuelle d'un modifié (en vert).....	71
Figure 1.29	Propagation des nouvelles contraintes imposées.....	71
Figure 1.30	Sélection d'un nouveau groupe.....	72
Figure 1.31	Choix manuel du modifié, désambiguïsation terminée.....	72
Figure 1.32	Architecture du système ITS2.....	74
CHAPITRE 2	UNE VOIE NOUVELLE : LA TAO INDIVIDUELLE OU "TAFD POUR TOUS"	
Figure 2.1	Une première architecture de distribution.....	80
Figure 2.2	Une seconde architecture de distribution.....	81
Figure 2.3	Une troisième architecture de distribution.....	81
Figure 2.4	Configuration privée pour la TAFD.....	81
CHAPITRE 3	LIDIA-1, UNE MAQUETTE DE TAFD	
Figure 3.1	Station de traduction hébergeant le serveur de communication.	93
Figure 3.2	Serveur de communication autonome.....	93
Figure 3.3	Organisation logique des processus linguistiques en Ariane-G5.....	95
Figure 3.4	Un arbre d'analyse multiniveau.....	97
Figure 3.5	Organisation générale du processus de traduction en LIDIA-1.....	99
CHAPITRE 4	UTILISATEUR ET UTILISATION	
Figure 4.1	Bouton d'accès aux traitements LIDIA.....	106
Figure 4.2	Menu LIDIA minimum.....	107
Figure 4.3	Menu LIDIA après définition de préférences.....	107
Figure 4.4	Schéma de construction d'une pile HyperCard.....	108
Figure 4.5	Les objets HyperCard.....	109
Figure 4.6	Organisation des dossiers LIDIA.....	110
Figure 4.7	Menu surgissant pour choisir un panneau de profil.	111
Figure 4.8	Préférences LIDIA "Profil de la tâche".....	112
Figure 4.9	Préférences LIDIA "Profil des ressources lexicales".....	113
Figure 4.10	Préférences LIDIA : sélection du correcteur orthographique.....	114
Figure 4.11	Sélection du serveur de traduction.....	114
Figure 4.12	Préférences LIDIA "Profil des serveurs".....	115
Figure 4.13	Préférences LIDIA : sélection du serveur de communication.....	115
Figure 4.14	Préférences LIDIA "profil de l'utilisateur".....	116
Figure 4.15	Traitement spécial en mode normal.....	120
Figure 4.16	Traitement spécial avec reprise de la désambiguïsation.....	120
Figure 4.17	Traitement spécial... reprise complète.....	121
Figure 4.18	Textoïde montrant les annotations.....	122

Figure 4.19	Textoïde montrant la rétrotraduction.....	122
Figure 4.20	État d'avancement des traitements	122
Figure 4.21	Menu LIDIA complet.....	123
Figure 4.22	Menu Messages : accès par carte.....	123
Figure 4.23	Menu Messages : accès par traitement (1).....	123
Figure 4.24	Menu Messages : accès par traitement (2).....	124
Figure 4.25	Palette Lidia correspondant au menu de la figure 4.21	124
Figure 4.26	État du traitement et demande de typage.....	126
Figure 4.27	État du traitement et demande de clarification	126
Figure 4.28	État du traitement, annotations et rétrotraduction.....	126

CHAPITRE 5 STANDARDISATION

Figure 5.1	Résultat d'analyse produit par 101	128
Figure 5.2	Saisie d'un nouvel adjectif avec 101	129
Figure 5.3	Saisie d'un nouveau nom commun avec 101	130
Figure 5.4	Correction orthographique, scénario imaginé	131
Figure 5.5	Correction orthographique, scénario réalisé.....	131
Figure 5.6	101 détecte une erreur.....	132
Figure 5.7	101 explique l'erreur.....	132
Figure 5.8	101 trouve une autre erreur.....	132
Figure 5.9	Standardisation terminologique, scénario imaginé.....	133
Figure 5.10	Détection d'un terme non standard.....	135
Figure 5.11	Validation de remplacement par un terme standard	135
Figure 5.12	Maintien de l'utilisation d'un terme non standard.....	136
Figure 5.13	Détection d'un terme non standard.....	136
Figure 5.13	Remplacement d'un terme avec conflit en genre et en nombre.....	137
Figure 5.14	Poursuite de la standardisation terminologique.....	137
Figure 5.15	Scénario envisagé pour le traitement des tournures	138
Figure 5.16	Détection de la tournure potentielle "gardien de l'ordre"	139
Figure 5.17	Validation de la tournure "gardien de l'ordre"	140
Figure 5.18	Détection de la tournure potentielle "ordre du jour"	140
Figure 5.19	Typage textuel, scénario imaginé	141
Figure 5.20	Sélection du style d'énoncé -Bouton d'action.....	142
Figure 5.21	Exemples associés au style d'énoncé -Bouton d'action	143
Figure 5.22	Définition associée au style d'énoncé -Bouton d'action	143
Figure 5.23	Sélection d'un type textuel non exemplifié	144
Figure 5.24	Choix multiple pour le type d'un énoncé	144

CHAPITRE 6 CLARIFICATION

Figure 6.1	Carte d'histoires relatives à une phrase du corpus LIDIA-1	156
------------	--	-----

Figure 6.2	Carte de traitement d'une histoire - sans contexte	156
Figure 6.3	Carte de traitement d'une histoire - avec contexte.....	157
Figure 6.4	Scénario pour la clarification.....	158
Figure 6.5	Dialogue avec complétion et accès à des explications.....	160
Figure 6.6	Explications pour l'ambiguïté de classe syntaxique	160
Figure 6.7	Dialogue avec paraphrasage et accès à des explications.....	161
Figure 6.8	Explications pour l'ambiguïté de subordination adjectivale.....	161
Figure 6.9	Dialogue avec remplacement sémantique et accès à des explications.	162
Figure 6.10	Explications pour l'ambiguïté de construction argumentaire du verbe	162
Figure 6.11	Dialogue pour expert non linguiste – 1.....	163
Figure 6.12	Dialogue pour expert non linguiste – 2.....	163
Figure 6.13	Dialogue pour expert non linguiste – 3.....	164

CHAPITRE 7 ARCHITECTURE

Figure 7.1	Hierarchie par défaut de transmission des messages dans HyperCard	172
Figure 7.2	Entrée "fermer" dans la base Parax.....	174
Figure 7.3	Problème contrastif entre le français et l'allemand.....	175
Figure 7.4	Équivalent russe de #fermer_replier	175
Figure 7.5	Composants logiciels de la station de rédaction	176
Figure 7.6	Composants logiciels sur le serveur de communication	177
Figure 7.7	Composants logiciels sur le serveur de traduction.....	178
Figure 7.8	Le serveur de communication et une machine interface.....	179
Figure 7.9	Une machine serveur.....	179
Figure 7.10	Les machines virtuelles du réseau LIDIA/IBM.....	180
Figure 7.11	Traitements linguistiques effectués sur le serveur de traduction	182
Figure 7.12	Entrée française "atteindre" de la base Parax	188
Figure 7.13	Entrée du dictionnaire d'analyse du français en Ariane-G5	189
Figure 7.14	Entrée "atteindre" du dictionnaire de transfert français-allemand.....	189
Figure 7.15	Entrée allemande "treffen" de la base Parax.....	189
Figure 7.16	Entrée "erreichen" du dictionnaire de génération de l'allemand	190

CHAPITRE 8 IMPLÉMENTATION

Figure 8.1	Vue générale de l'architecture de la maquette LIDIA-1.....	191
Figure 8.2	Services LIDIA sur la station de rédaction	192
Figure 8.3	Implémentation du serveur de communication.....	194
Figure 8.4	Script de upLoad	198
Figure 8.5	Communication et HyperCard.....	199
Figure 8.6	Communication et serveur LIDIA	199
Figure 8.7	Communication et serveur de communication.....	200
Figure 8.8	Extrait du transcripteur.....	203

CHAPITRE 9 **DÉSAMBIGUÏSATION**

Figure 9.1	Ambiguïté de coordination verbale	209
Figure 9.2	Ambiguïté de classe simple	210
Figure 9.3	Ambiguïté de structure argumentaire du verbe	211
Figure 9.4	Ambiguïté de coordination nominale	212
Figure 9.5	Ambiguïté de coordination adjectivale.....	213
Figure 9.6	Ambiguïté de subordination	214
Figure 9.7	Ambiguïté d'étiquetage logico-sémantique.....	215
Figure 9.8	Ambiguïté d'étiquetage syntaxique	216
Figure 9.9	Ambiguïté d'ordre des arguments d'un verbe transitif direct.....	217
Figure 9.10	Stratégie de production de l'arbre de questions.....	219
Figure 9.11	Arbre de désambiguïsation	220
Figure 9.12	Arbre de désambiguïsation concret	220
Figure 9.13	Patrons pour la coordination verbale	228
Figure 9.14	Patrons pour la structure argumentaire du verbe – type 1	229
Figure 9.15	Patrons pour la structure argumentaire du verbe – type 2	229
Figure 9.16	Patrons pour la coordination nominale.....	230
Figure 9.17	Patrons pour la coordination adjectivale.....	231
Figure 9.18	Patrons pour la subordination – type 1	232
Figure 9.19	Patrons pour la subordination – type 2	233
Figure 9.20	Dialogue pour l'ambiguïté de polysémie – exemple 1	235
Figure 9.21	Dialogue pour l'ambiguïté de polysémie – exemple 2	235
Figure 9.22	Dialogue pour l'ambiguïté de coordination verbale	236
Figure 9.23	Dialogue pour l'ambiguïté de classe syntaxique – exemple 1.....	236
Figure 9.24	Dialogue pour l'ambiguïté de classe syntaxique – exemple 2.....	237
Figure 9.25	Dialogue pour l'ambiguïté de structure argumentaire – type 1	237
Figure 9.26	Dialogue pour l'ambiguïté de structure argumentaire– type 2	238
Figure 9.27	Dialogue pour l'ambiguïté de coordination nominale.....	238
Figure 9.28	Dialogue pour l'ambiguïté de coordination adjectivale.....	239
Figure 9.29	Dialogue pour l'ambiguïté de subordination type 1	239
Figure 9.30	Dialogue pour l'ambiguïté de subordination type 1 généralisé.....	240
Figure 9.31	Dialogue pour l'ambiguïté de subordination – seconde étape 1.....	240
Figure 9.32	Dialogue pour l'ambiguïté de subordination – seconde étape 2.....	241
Figure 9.33	Dialogue pour l'ambiguïté de subordination type 2 – exemple 1.....	241
Figure 9.34	Dialogue pour l'ambiguïté de subordination type 2 – exemple 2.....	242
Figure 9.35	Dialogue pour l'ambiguïté d'étiquetage logico-sémantique	242
Figure 9.36	Dialogue pour l'ambiguïté d'étiquetage syntaxique – exemple 1	243
Figure 9.37	Dialogue pour l'ambiguïté d'étiquetage syntaxique – exemple 2	243
Figure 9.38	Dialogue pour l'ambiguïté d'ordre des arguments d'un verbe.....	244

Introduction

Situation, motivation

Les recherches et développements en Traduction Automatisée Fondée sur le Dialogue (TAFD) sont motivés par l'incapacité des paradigmes actuels à répondre aux nouveaux besoins créés par l'importance croissante des langues nationales dans le contexte de l'internationalisation, et par les récents progrès méthodologiques et technologiques.

Paradigmes actuels en TAO¹

Vers 1949, les USA, puis l'URSS, ont lancé des programmes motivés par le besoin de renseignements. C'est la *TAO du veilleur*.. Il s'agit de traduction totalement automatique, dont on attend des traductions "grossières", produites 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...)². Ce besoin est toujours actuel. Cependant, il s'agit maintenant plus de veille scientifique, technique, économique et financière que de renseignement militaire³.

Une quinzaine d'années plus tard, on a commencé à travailler sur la *TAO du réviseur*, dans laquelle on produit des traductions "brutes", destinées à être révisées. Dans cette optique, la

¹ Certains passages de cette thèse sont extraits de documents réalisés par différents membres du GETA dans le cadre du projet LIDIA. Je n'ai pas jugé utile de les paraphraser. Les passages cités sont en style citation. Les trois paragraphes de cette partie sont repris de [Boitet 1993] lui-même version française étendue de [Boitet & Blanchon 1993b].

² 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.

³ À titre d'exemple, on peut citer l'accès en anglais à des bases de données japonaises depuis la Suède ou encore depuis l'Europe (service Japinfo utilisant des traductions "grossières" d'ATLAS-II à peine "arrangées").

machine doit remplacer le traducteur, promu réviseur⁴. 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 a commencé à apporter des solutions, 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 thesaurus de "bitextes" (textes + traductions) etc., accessibles depuis un traitement de texte, le tout formant un "poste de travail pour le traducteur", réalisé sur micro-ordinateur ou station de travail. Il s'agit d'outils comme Mercury/Termex™ sur PC, WinTool™ sur Macintosh, ou de systèmes complets (Weidner-Bravice [Balkan 1992], TSS de Alps [Zirkle 1990]). Le réviseur peut utiliser le même environnement, ou préférer travailler directement sous l'outil final de PAO (publication assistée par ordinateur).

Limitation des paradigmes actuels pour de nouveaux besoins

Il est maintenant souhaitable et possible de s'attaquer à la TAO personnelle, ou TAO pour le rédacteur ; ou mieux, **TA Fondée sur le Dialogue**. 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. En effet, 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 des langues traditionnellement importantes de ce point de vue, et d'une promotion volontariste de bien d'autres, pour les amener au même niveau (malais-indonésien ou arabe, par exemple). A notre sens, cette évolution ne fera que se renforcer, les langues étant, comme le note le Professeur 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 cela est envisageable, le résultat est souvent difficile à comprendre, voire illisible.

Les trois types de TAO "classique" ne conviennent évidemment pas 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 du grand public, supposés ne connaître aucune des langues cibles (ou au plus une, et ce imparfaitement).

⁴ 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 est fait chez Xerox avec Systran pour la traduction de notices d'anglais en 4 langues.

Or, la TAO pour le rédacteur ne permet aucune révision. Sauf si l'on conçoit un système pour un groupe de spécialistes, elle ne permet pas non plus le recours à une base de connaissances pour "comprendre explicitement" le contenu des textes. Enfin, elle ne permet pas de se restreindre à un langage "contrôlé" (langage artificiel, non ambigu, à "consonance naturelle"), ni même à un "sous-langage" déterminé. C'est le grand public qu'il faut viser.

Est-ce la quadrature du cercle ? Oui, si l'on veut tout à la fois : du texte libre, pas de prédiction, pas de postédition, et une grande qualité. Non, si l'on accepte un **langage guidé**, et une **interaction avec l'auteur**, permettant d'arriver :

- à une "standardisation" du texte à traduire, sous ses aspects lexicaux, grammaticaux et stylistiques ;
- à une "clarification" de ce texte, permettant de réduire les ambiguïtés lexicales, grammaticales et sémantiques .

Les systèmes classiques sont essentiellement fondés sur des connaissances linguistiques (structure de la langue, régularité du corpus, sémantique "intrinsèque"). Il existe un seul système prototype, KBMT-89, de CMU [Nirenburg & al. 1989], "fondé sur la connaissance", qui utilise une "ontologie" de son domaine (PC-XT et PC 5550 japonisé). Un système dérivé, KANT [Nyberg & Mitamura 1992], à vocation industrielle, est en cours de développement. Son architecture impose de surcroît l'emploi d'un langage contrôlé, ce qui la rend d'autant moins utilisable pour une future "TAO pour tous".

Les systèmes de TAO personnelle seront, eux, essentiellement fondés sur le dialogue, même si, dans certains cas, on peut imaginer de les lier à un système expert du domaine traité. Si nous ne pouvons coder la connaissance nécessaire à une compréhension explicite parfaite, nous pouvons utiliser le dialogue pour "aller la chercher dans la tête de l'auteur", et aboutir à une "compréhension indirecte", c'est-à-dire à une représentation linguistique "profonde" du texte de qualité et de finesse au moins égales à celles que pourrait produire un processus de "compréhension explicite" (au moins égales, car l'auteur sera toujours plus compétent que tout système expert sur ce qu'il veut dire).

Progrès méthodologiques et technologiques

L'idée de la TAFD date des années soixante dix [Kay 1973] et a été incorporée à plusieurs maquettes ou prototypes dans les années soixante-dix et quatre-vingt [Chandler, *et al.* 1987 ; Melby, *et al.* 1980 ; Weaver 1988]. Si ces travaux n'ont pas donné lieu à des systèmes utilisables en pratique, c'est que les dialogues devaient être conduits par des spécialistes⁵, 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* [Blanchon 1990a ; Boitet 1989a ; Huang 1990 ; Maruyama, *et al.* 1990 ; Sadler 1989 ; Somers, *et al.* 1990]. Nous préférons ce dernier terme. D'un côté, en effet, "auteur" est moins restrictif que "rédacteur" : un auteur est quelqu'un qui crée 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

⁵ ITS [Melby, *et al.* 1980] demandait même *plusieurs* intervenants, un linguiste spécialiste du système pour l'analyse, et un bilingue pour chaque langue cible.

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”⁶.

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 éventuellement 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.

⁶ “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.

Intérêt

La TAFD pour l'auteur, par le public qu'elle vise et les situations traductionnelles qu'elle entend couvrir, pose un certain nombre de nouveaux problèmes, eux-mêmes dûs à un nouvel ensemble de contraintes matérielles et logicielles, linguicielles, et ergonomiques. Elle suggère aussi de nouvelles approches.

L'auteur que nous envisageons n'aura pas à sa disposition une station de travail de haut de gamme, mais un micro du genre Macintosh ou PC de milieu de gamme. Il faut donc partager le linguiciel et distribuer les tâches. Il faut aussi que l'accès au processus de traduction soit parfaitement intégré à l'environnement de rédaction.

Du côté du linguiciel, la TAFD nécessite une couverture lexicale et grammaticale très large. Cette contrainte suggère de construire et utiliser une base lexicale multilingue par "acceptions interlingues". Elle suggère aussi la définition de types de texte qui permettraient de guider l'analyse, et de préférences lexicales qui permettraient de diminuer la polysémie des mots.

En TAFD, l'ergonomie est un aspect absolument crucial, et les choix ergonomiques influencent directement l'architecture du système. Les problèmes qui méritent d'être abordés concernent d'abord le contenu et les modalités du dialogue de désambiguïsation, l'adéquation du niveau de dialogue avec l'expertise de l'utilisateur, et enfin l'accessibilité des connaissances linguistiques du système, afin que l'utilisateur apprenne à mieux rédiger.

Pour apporter une réponse cohérente à ces problèmes, il faut intégrer différents savoir-faire de façon plus harmonieuse. Les aspects ergonomique, système et linguistique doivent être intégrés à part égale afin de pouvoir proposer une solution qui s'approche au plus près de nos objectifs.

Essayer d'apporter des solutions à certains des problèmes posés, dans le cadre d'une approche pluridisciplinaire, tel est l'objectif du présent travail.

Organisation de la thèse

Dans la première partie de ce document, intitulée “De la TA interactive pour spécialiste à la TAFD pour tous”, nous examinons d’abord les expériences précédentes en les classant en trois grandes catégories :

- les expériences pionnières,
- les systèmes dans lesquels on a introduit le dialogue après coup, dans une architecture initialement totalement automatique, pour remédier aux défauts et lacunes constatés, et enfin
- les systèmes qui ont été pensés et conçus comme une combinaison de processus automatiques et de processus plus ou moins interactifs. Nous pouvons alors introduire le concept de TAFD, et préciser quelles en sont les contraintes spécifiques, quelle architecture linguistique et quelles voies intermédiaires nouvelles sont à explorer, et proposer des situations traductionnelles adaptées — à notre avis. Nous décrivons enfin la situation concrète choisie par le GETA pour réaliser la maquette LIDIA-1.

Lorsque nous avons pu trouver des informations sur le déroulement d’une session de désambiguïsation interactive, nous avons choisi de les inclure dans le corps du document et de ne pas les rejeter en annexes. L’interaction doit être considérée comme un élément fondamental des systèmes auxquels nous nous sommes intéressés. Montrer des sessions complètes de désambiguïsation permet d’imaginer les conditions réelles d’utilisation et de les comparer avec ce que nous proposons. Ainsi, dans cette partie, pour les projets KBMT-89 et JETS, nous donnons toutes les informations que nous avons pu obtenir grâce à l’amabilité des auteurs.

Dans la seconde partie, intitulée “La maquette LIDIA-1 – vue externe”, nous montrons la vision qu’un utilisateur aurait d’un système réel. Le premier chapitre décrit l’environnement de travail de l’utilisateur, comment il peut personnaliser cet environnement et comment il peut mettre en œuvre les traitements LIDIA. Le second et le troisième chapitre présentent respectivement les choix effectués dans la réalisation des

classes de traitement définies, à savoir la standardisation⁷ et la clarification⁸. En fin de partie, le lecteur trouvera des illustrations permettant d'imaginer les niveaux de dialogues qu'un prototype devrait offrir pour que l'interaction soit adaptée à l'expérience de l'utilisateur.

Dans la troisième partie, intitulée “La maquette LIDIA-1 – vue interne”, nous décrivons les mécanismes mis en jeu pour qu'un tel système fonctionne. Notre programme comporte trois niveaux de distribution des tâches, sur la station de rédaction, sur un serveur de communication et sur un serveur de traduction. Nous présentons en détail les composants qui, sur la station de rédaction et les serveurs, coopèrent pour mener à bien le travail de traduction. Nous décrivons aussi l'implémentation réalisée. Nous présentons en particulier la méthode que nous avons conçue pour produire le dialogue de désambiguïsation, méthode dont les principales qualités sont la simplicité et la généricité. Afin d'illustrer la méthode que nous proposons, nous fournissons des exemples commentés de sessions de désambiguïsation, pour chaque classe d'ambiguïté retenue dans la maquette.

⁷ La standardisation doit permettre d'obtenir un texte typé et correct du point de vue lexical (mots préférés), orthographique et syntaxique.

⁸ La clarification doit permettre de désambiguïser le texte au niveau du lexique et de la syntaxe de surface et profonde.

Première partie

De la TA Interactive pour spécialiste
à la TAFD pour tous

Introduction

Bien sûr, nous ne sommes pas les premiers à avoir pensé à introduire l'interaction homme-machine dans un processus de traduction, même fortement automatisé. Notre concept de TAFD marque une évolution logique, fondée sur les leçons d'un certain nombre d'expériences antérieures.

Le premier chapitre de cette partie, intitulé “les expériences antérieures”, présente un état de l'art de la TA ou TAO interactive, en classant les travaux pertinents en trois classes : – les pionniers et la TA interactive pour spécialistes, – l'interaction au secours du système, – l'interaction comme pilier du système.

Dans le chapitre suivant intitulé “Une nouvelle approche : la TAO individuelle ou la “TAFD pour tous”, nous présentons notre approche comme une suite logique de ces expériences du “troisième type” (l'interaction comme pilier). Il s'agit du concept de Traduction Automatisée Fondée sur le Dialogue. Ce concept induit ses contraintes spécifiques — contraintes matérielles et logicielles, linguicielles, et ergonomiques —, nécessite l'exploration de nouvelles architectures linguistiques, et, finalement, s'adresse à un ensemble de situations précises.

Enfin, dans le troisième chapitre, intitulé “LIDIA-1, une maquette de TAFD”, nous décrivons la situation choisie au GETA pour maquetter un système de TAFD. Nous avons choisi de traduire des piles HyperCard du français vers l'anglais, l'allemand et le russe. Les outils de traduction sont accessibles sur un serveur de traduction depuis la station de rédaction. Ils sont écrits en Ariane-G5, le générateur de systèmes de TAO développé par le GETA. Les étapes de standardisation et de clarification sont prises en charge par les logiciels résidant sur la station de rédaction. Dans cette première maquette, le dialogue de désambiguïsation est unimodal.

Chapitre 1

Les expériences précédentes

1.1. Les pionniers et la TA interactive pour le traducteur

Il semble que l'idée d'utiliser l'interaction homme-machine soit due à Kay et Kaplan, qui l'ont expérimentée pour la première fois dans les années 60 dans le cadre du projet MIND. D'autres travaux en TA interactive ont suivi, notamment les systèmes ITS et ALPS, tous deux construits à Provo, l'un pour la recherche (et éventuellement la traduction des textes sacrés des Mormons) et l'autre pour la commercialisation auprès des services de traduction et des traducteurs indépendants.

1.1.1. MIND (1963-73)

a. Historique

Dans la fin des années 60 et le début des années 70, l'un des grands projets en linguistique informatique fut le projet MIND (Management of Information through Natural Discourse) à la Rand Corporation. Ses principaux maîtres d'œuvre furent Martin Kay et Ronald Kaplan. MIND [Hutchins 1986 p296 ; Kay 1973] devait être un système de traitement du langage comprenant un ensemble extensible de programmes de base pour le traitement linguistique. Ces programmes devaient pouvoir être combinés pour réaliser différents types de systèmes, depuis des testeurs de grammaires jusqu'à des systèmes de traduction automatique en passant par des systèmes de question-réponse.

b. Structure du système

Les principaux composants de MIND sont : un analyseur morphologique, un processeur syntaxique, un "désambiguïseur" ("disambiguator" en anglais), un processeur sémantique et un composant de sortie. Un superviseur organise l'ordonnancement de ces composants pour construire des processus plus complexes.

c. La désambiguïsation

Nous nous intéressons ici au “désambigüiseur”. Le rôle de ce composant est d’élaborer et d’implémenter une stratégie pour lever les ambiguïtés qui apparaissent dans la structure produite par l’analyseur syntaxique. Élaborer une stratégie, c’est choisir parmi un ensemble de possibilités, parfois très grand, celle qui semble demander le moins de travail. À chaque pas, le programme essaie de poser une question dont la réponse produise une partition de l’ensemble des analyses possibles qui comporte le plus grand nombre de parties ayant des cardinaux les plus égaux possibles.

Voici un exemple de session imaginée par M. Kay [Kay 1973]

Soit à traduire la phrase suivante en français : “They filled the tank with gas.”

i. Traitement de la polysémie :

Le système pourrait demander d’abord :

DOES THE WORD “TANK” REFER TO	
1.	A MILITARY VEHICLE, OR
2.	A VESSEL FOR FLUIDS?

Avec la réponse 1, le système produirait “char d’assaut”, “réservoir” sinon.

Le système demanderait ensuite :

DOES “GAS” REFER TO	
1.	GASOLINE, OR
2.	VAPOR?

Avec la réponse 1, le système produirait “essence”, “gaz” sinon.

ii. Traitement de l’attachement des constituants :

Une troisième question porterait sur l’attachement de “with gas” :

DOES THIS MEAN	
1.	“FILL WITH GAS”
2.	“TANK WITH GAS”

Dans le premier cas, “with gas” est l’argument 2 du verbe “fill” (quelqu’un (arg0) remplit quelque chose (arg1) avec quelque chose (arg2)). Dans le second cas, “with gas” est un complément de “tank” : une propriété de “tank”, (un réservoir contenant de l’essence).

iii. Traitement de la référence pronominale :

DOES “THEY” REFER TO	
1.	THE SOLDIERS,
2.	THE TANKS,
3.	THE SHELLS,
4.	THE ENEMY,
	...etc.

Les référents possibles seraient trouvés dans le texte précédent ou le contexte suivant rapproché qui suit. Kay pense qu’une question du type

WHAT DOES “THEY” REFER TO?

serait préférable.

d. Conclusion

Avec le projet MIND, on a la première apparition de l’idée d’un processus de désambiguïsation interactive. Les questions envisagées ne font pas appel à des connaissances linguistiques. Les ambiguïtés sont réparties en catégories. M. Kay indique que le système serait plus performant si les ambiguïtés transportées d’une langue vers l’autre ne faisaient pas l’objet de questions. En effet, si l’on veut traduire vers le français “He saw the girl with the telescope”, l’ambiguïté de structure est conservée. Dans un tel cas, il faut faire confiance au lecteur humain pour retrouver le sens exact, et on ne devrait donc pas proposer de dialogue de désambiguïsation.

Comme on ne trouve dans la littérature qu’une seule référence, il semble que le projet MIND soit resté une expérience de laboratoire. Cela est probablement dû à l’ambition de ses concepteurs d’en faire une sorte de boîte à outils pour le traitement linguistique et à l’état de la linguistique informatique à cette époque.

Les idées de Kay, à Xerox Parc depuis les années 70, ont évolué vers une vision plus pragmatique de la situation. Dès 1980 [Kay 1980 ; Kay 1982], il a proposé l’intégration progressive de réels savoir-faire dans une station de travail du traducteur afin que l’on puisse utiliser les acquis du domaine immédiatement et de façon productive.

1.1.2. ITS (1973-81)

a. Historique

Le Translation Science Institute fut créé en 1970 à Provo (Utah) par l’Université Brigham Young et l’Église des Mormons [Hutchins 1986 p299] et placé sous la direction de Eldon G. Lytle. Le projet visait au développement d’un système interactif pour traduire des textes anglais vers le français, l’espagnol, le portugais, l’allemand et le chinois [Melby 1978 ; Melby, *et al.* 1980]. En 1979, une évaluation menée par les commanditaires a montré qu’ITS avait peu de possibilités commerciales [Melby 1981]. À cette date, une partie de l’équipe créa la société privée ALPS.

Après la création de ALPS, Alan K. Melby continua de travailler au développement d’une nouvelle version d’ITS [Melby 1981 ; Melby 1982]. Avec la première version, les

traducteurs pensaient qu'ils travaillaient pour la machine car ITS les forçait «à répondre à beaucoup de questions inintéressantes et à réviser beaucoup de phrases dont ils pensaient qu'elles auraient dû être correctement traduites». Dans la nouvelle version, les traducteurs devaient contrôler le processus et avoir la machine à leur service. Nous n'avons pas trouvé de trace de cette nouvelle version dans la littérature.

En fait, les travaux semblent avoir été abandonnés vers 1980-1981, Melby se tournant alors vers des aides aux traducteurs professionnels travaillant sur PC [Melby 1982 ; Melby 1987 ; Melby 1991].

b. Fiche d'identité

Nom du projet	ITS (Interactive Translation system)
Langue(s) source(s)	anglais
Langue(s) cible(s)	espagnol, portugais, allemand, français
Type de système	outil de TAO interactif pour traducteur
Paradigme	jusqu'en 1980, système de traduction 1 vers n de type seconde génération : analyse, transfert, génération ; architecture proposée pour la nouvelle version : système de traduction 1 vers 1
Architecture informatique	terminaux branchés sur un mainframe IBM, puis ordinateurs personnels en réseau
Corpus	textes issus des publications de la LDS Church (Mormons). Ces articles ne sont pas tous des dissertations théologiques. On y trouve des articles de beaucoup d'auteurs dans des styles très différents sur des thèmes très variés comme le jardinage, la nouvelle littéraire, les cultures étrangères, les relations parents-enfants, etc.

c. Structure du système

L'interaction apparaît en analyse et en transfert. L'interaction en transfert ne peut pas être partagée par les langues cibles (elle n'est pas multilingue).

Lors de la phase d'analyse, le texte est traité phrase par phrase, et une représentation du sens de chaque phrase est produite. Cette représentation comporte toutes les informations explicitement présentes dans la phrase, et éventuellement des informations contextuelles — calculées à partir des informations explicites qui apparaissent soit dans la phrase, soit ailleurs dans le texte —, ou fournies par l'utilisateur (en fait, la plus grande partie). Pendant cette phase, l'ordinateur pose des questions à l'opérateur sur l'écran du terminal.

Lors de la phase de transfert, on doit aussi poser des questions pour chaque langue cible afin de résoudre le problème du choix des mots à utiliser. Ce choix ne peut, en effet,

être fait automatiquement à partir de l'analyse du texte anglais. Après le processus de génération, le texte peut être postédité sur la console.

L'interaction sert à résoudre quatre types d'ambiguïté : ambiguïté de catégorie, ambiguïté de relation entre les mots, ambiguïté "sémantique" (cf. exemples plus loin) et le problème de choix des équivalents dans la langue cible.

d. Désambiguïsation avec ITS version 1

1. Après l'analyse morphologique, on pose des questions pour résoudre des ambiguïtés de catégorie ("structural disambiguation"). Dans un système automatique, ce type d'ambiguïté est très souvent résolu au cours des étapes qui suivent l'analyse morphologique. Les deux exemples suivants illustrent ce cas de figure.

exemples :

ENGLISH ANALYSIS - STRUCTURAL DISAMBIGUATION

SENTENCE ID : 1 She <1> called <2> up <3> the <4> stairs <5> to <6> her <7> sister
<8> after <9> calling <10> up <11> her <12> friend <13> on <14> the <15> phone
<16> . <17>

CATEGORY CLUSTER FOR : up <3>

1V Verb - They will have to up the price of gas.

2A Particle - You should look up the word in the book.

3P Preposition - They live just up the street.

SELECT CATEGORY

3 (confirm guess)

CONFL VL : 2

Dans l'exemple précédent, on présente à l'opérateur les trois catégories possibles pour l'occurrence "UP". Le système a déjà examiné le contexte — la phrase — pour déterminer le meilleur choix pour la catégorie du mot. Il propose une catégorie, avec un niveau de confiance (CONFL). En effet, lors de la recherche le système a trouvé que "UP"<3> est précédé d'un verbe et suivi d'un groupe nominal. Il a par conséquent décidé que "UP"<3> était une préposition, ce qui est exact. Mais on remarquera que l'occurrence "UP"<11> apparaît dans le même environnement en fonctionnant comme une particule.

ENGLISH ANALYSIS - STRUCTURAL DISAMBIGUATION

SENTENCE ID : 2 His <1> voice <2> is <3> relaxing <4> . <5>

CATEGORY CLUSTER FOR : is <3>

1 Vc Copula - He is ill. It is kicked by me.

2 Vp Progressive marker - He is swimming.

3 Ve Existential - There is a God in heaven.

SELECT CATEGORY

2 (enter 1)

CONFL VL : 2

Parfois un mot, qui n'appartient qu'à une seule catégorie, peut être associé à plusieurs structures. La structure qui doit être construite sera différente selon la fonction. En examinant l'occurrence "RELAXING", le système choisit la solution 2 (forme progressive), à cause de la marque "ING", alors qu'ici "RELAXING" fonctionne comme un simple adjectif. L'opérateur corrige en introduisant la bonne réponse.

2. Une première étape de l'analyse structurale construit la structure du descripteur (une structure arborescente appelée "arbre de jonction" ou "junction-tree") qui détermine les relations entre les mots. Une fois que les informations du dictionnaire ont été examinées pour chaque occurrence, l'étape suivante consiste à construire une représentation de la signification de la phrase ("structure building"). Ici, l'interaction sert à régler le problème des relations des mots entre eux.

exemples :

ENGLISH ANALYSIS - STRUCTURAL BUILDING

SENTENCE ID : 3 We <1> should <2> avoid <3> falling <4> victims <5> to <6> error <7> . <8>

WHAT IS falling <6> ?

PROMPT : (NOUN | J = ADJ | V = ADV) PV | V

GUESS : N PV

REPLY : (confirm guess)

Les mots avec un suffixe "ING" peuvent fonctionner de manières très différentes. Ils peuvent être nominaux, adjectivaux, ou adverbiaux. Ils peuvent être constitués d'un seul mot ou de tout un prédicat. Ici, "falling victims to error" est un prédicat. Dans ce cas, "ING" marque la nominalisation du prédicat.

ENGLISH ANALYSIS - STRUCTURAL BUILDING

SENTENCE ID : 4 The <1> witch <2> made <3> the <4> prince <5> a <6> frog <7> . <8>

WHAT IS frog <7> ?

PROMPT : DO/NVP/APPOS

GUESS : DO

REPLY : (confirm guess)

Lorsqu'au moins deux substantifs apparaissent en séquence, il est difficile de déterminer quelles sont les relations qui s'établissent entre eux. Ainsi, il y a deux interprétations pour la phrase précédente. Dans l'une, la sorcière transforme le prince en grenouille. Dans l'autre la sorcière fait apparaître une grenouille pour le prince. Dans le premier cas, prince est l'objet direct, et dans le second, c'est grenouille qui joue ce rôle. D'autres configurations posent aussi des problèmes parmi lesquelles on compte les appositions (my friend John), les composés substantif/substantif (a two car garage), les quantificateurs (he drove his car 1000 miles) et les noms fonctionnant comme adverbes (she gave the boy a book this morning).

3. Une deuxième étape de l'analyse annote le descripteur en précisant les propriétés sémantiques ("referential disambiguation").

exemples :

ENGLISH ANALYSIS - REFERENTIAL DISAMBIGUATION

SENTENCE ID : 5 I <1> found <2> the <3> pen <4> in <5> the <6> yard <7> . <8>

WHAT IS THE MEANING OF pen <4> ?

GUESS : 1

REPLY : (confirm guess)

1. writing instrument
 2. enclosure (e.g. children's play pen)
 3. penitentiary
-

Lorsque la structure de la phrase est découverte, des informations contextuelles lui sont rajoutées. De nombreux mots sont ambigus quant à leur sens. C'est l'utilisateur qui est alors le mieux placé pour préciser quel est le bon.

ENGLISH ANALYSIS - REFERENTIAL DISAMBIGUATION

SENTENCE ID : 6 The <1> saboteur <2> blew <3> up <4> the <5> balloon <6> . <7>

WHAT IS THE MEANING OF blew <3> up <4> ?

GUESS : 1

REPLY : (confirm guess)

1. explode
 2. inflate
 3. enlarge
 4. blow upwards
-

Cette désambiguïisation sémantique peut aussi être nécessaire sur des groupes d'occurrences qui doivent être considérés comme une unité. Dans la phrase ci-dessus, "blow up" est considéré comme une unité. Quelques expressions de ce type ne sont pas ambiguës (exemple : "to be fed up with"). Cependant, beaucoup d'expressions sont ambiguës et doivent être désambiguïées.

ENGLISH ANALYSIS - REFERENTIAL DISAMBIGUATION

SENTENCE ID : 7 You <1> should <2> seek <3> divine <4> help <5> through <6> prayers <7> . <8>

WHAT IS THE NUMBER OF you <1> ?

PROMPT : SINGULAR/PLURAL

GUESS : PLURAL

REPLY : (confirm guess)

Les occurrences peuvent aussi être ambiguës au niveau du genre. L'avertissement ci-dessus peut être adressé à une personne en particulier, ou à une audience. Seul le contexte peut le dire. Et le fait que "you" soit singulier ou pluriel a une grande importance pour la traduction.

 ENGLISH ANALYSIS - REFERENTIAL DISAMBIGUATION

SENTENCE ID : 8 The <1> president <2> is <3> arriving <4> at <5> the <6> airport <7> . <8>

WHAT IS THE RELATIVE TENSE OF is <3> ?

PROMPT : CONTEMPORANEOUS / POSTERIOR

GUESS : CONTEMPORANEOUS

REPLY : POSTERIOR

On rencontre aussi des problèmes avec le temps des verbes. L'événement décrit ci-dessus peut avoir lieu aussi bien dans le présent que dans le futur. C'est le contexte seul qui renseigne sur le temps réel.

4. Lorsque le processus d'analyse a pu construire une représentation du sens unique pour une phrase, on procède au choix lexical. Lors du transfert lexical, des questions peuvent être posées pour le choix des équivalents qui ne peuvent être trouvés automatiquement par l'analyse de l'anglais. Cette phase dépend de la langue cible, et doit donc être répétée pour chaque langue visée.

5. Le texte produit doit ensuite être révisé et reformaté.

e. Conclusion

Les mesures du temps d'interaction furent décevantes : pour une page de 250 occurrences, on compte 7 minutes d'interaction pendant la phase d'analyse, 10 pour la phase de transfert et 15 minutes de postédition. Pour un total de 32 minutes, on atteint une qualité à peu près équivalente au premier brouillon d'un traducteur humain. Selon les auteurs, pour arriver à une utilisation intéressante et attrayante, l'interaction devrait prendre 10 minutes par page et la sortie devrait nécessiter quelques modifications mineures.

Les questions posées pour résoudre une ambiguïté utilisent des termes relatifs au modèle linguistique sous-jacent. Si certaines questions sont facilement compréhensibles, d'autres font intervenir une connaissance très précise du modèle linguistique. L'utilisateur doit donc être un linguiste spécialiste de ce système. De plus, pour répondre aux questions posées pendant la phase de transfert, l'utilisateur doit aussi avoir une connaissance de la langue cible.

De nouvelles versions du système devaient être développées et différer de la version 1 en certains points, ce devait être des systèmes bilingues. L'idée de 1 vers n expérimentée précédemment était intéressante parce que la plus grande partie de l'interaction servait pour toutes les langues visées. Mais, de l'avis des auteurs eux-mêmes, l'analyse était rendue trop complexe pour correspondre à n'importe quelle langue cible. Ainsi Melby [Melby 1981 p27 §4] dit «This is somewhat like trying to define a foreign policy which every citizen will approve of.» (ce que l'on peut traduire par : c'est un peu comme si on souhaitait définir une politique étrangère qui soit approuvée par chaque citoyen). Ce commentaire nous permet de douter du caractère de seconde génération mis en avant pour le type du système. Le travail sur la version suivante devait se concentrer sur la réduction de l'interaction et devait rendre les questions plus intelligibles, plus intuitives que dans la version précédente, sans un trop grand entraînement. De ces travaux annoncés, nous n'avons trouvé aucune trace dans la littérature.

1.1.3. ALPS (1980-)

a. Historique

La société ALPS (Automated Language Processing Systems Ltd.) fut fondée en 1980 par cinq des développeurs de ITS afin de donner une chance commerciale à leur approche interactive de la traduction. Le système TSS a été mis sur le marché en 1983 [Bateman 1985 ; Tenny 1985] et la compagnie a ouvert en 1984 un bureau à Neuchâtel. ALPS s'appelle maintenant ALPNET [Zirkle 1990].

b. Fiche d'identité

Nom du produit	TSS (Translation Support System)
Couples de langues disponibles en 1992	anglais Æ français, allemand, espagnol et italien français Æ anglais allemand Æ anglais
Type de système	Outil interactif d'aide au traducteur
Investissement	40 homme/année
Architecture informatique	micro-ordinateur IBM PC/AT, IBM 4300
Corpus	Pas de domaine particulier

c. Structure du système

Le Système TSS (Translation Support System) [Sibley 1988] est un progiciel conçu pour seconder le traducteur tout au long du processus de traduction. TSS automatise les recherches terminologiques, la création et l'enrichissement des dictionnaires, ainsi que la mise en forme, l'impression et la transmission du document traduit.

Pour la traduction, TSS regroupe plusieurs outils.

Éditeur de traduction : l'éditeur de traduction se caractérise par la présentation unique à l'écran de multiples possibilités de mise en forme et par l'accès direct aux dictionnaires en ligne. L'éditeur de traduction est exploitable à partir de — et vers — toutes les langues utilisant l'alphabet romain.

AutoTerm : la fonction AutoTerm [Good 1988] effectue automatiquement une recherche terminologique dans les dictionnaires en ligne et affiche à l'intention du traducteur la traduction des termes figurant dans le texte de départ. AutoTerm permet de travailler vers toutes les langues utilisant l'alphabet romain, à partir des langues de départ suivantes : français, allemand, espagnol, italien, anglais.

TransActive : la fonction TransActive produit une première traduction d'un texte en procédant phrase par phrase. Le traducteur peut ensuite modifier ou améliorer la traduction comme il l'entend. TransActive permet de traduire à partir de l'anglais vers le français, l'allemand, l'espagnol et l'italien ; à partir du français vers l'anglais ; à partir de l'allemand vers l'anglais.

Caractéristiques de TransActive :

- identification par le programme des ambiguïtés syntaxiques, grammaticales et lexicales, et leur résolution par les interventions conjuguées du progiciel et du traducteur,
- présentation au traducteur de la première traduction de chaque phrase pour révision et modification si nécessaire,
- accès direct aux dictionnaires permettant au traducteur de les réviser et de les actualiser pour chaque document.

Traitement des répétitions : le traitement des répétitions qui intervient exclusivement dans le domaine de la traduction, retrouve automatiquement les traductions du texte de départ qui se répètent, et les place dans le texte d'arrivée. L'utilisateur peut décider à tout moment d'utiliser cette fonction interactivement ou automatiquement, ou de s'en passer. Le traitement des répétitions peut aussi être suspendu à tout moment.

d. Désambiguïssation avec TransActive

Le système propose trois types d'interaction [Weaver 1988] : lexicale, syntaxique, et stylistique.

L'interaction la plus commune est l'interaction lexicale. Souvent, une occurrence du texte source peut être traduite de plusieurs façons dans la langue cible. Si le bon sens est dicté par des contraintes syntaxiques, un processus d'analyse complexe choisit la traduction appropriée. Lorsque le système ne peut choisir, il demande l'aide de l'utilisateur en présentant la liste des sens possibles dans la langue cible visée.

exemple vers l'espagnol :

(1) The information stored in the memory is processed immediately.

choose a translation for : information

(a) datos

(b) información

Les concepteurs font remarquer avec justesse que si le choix est laissé à la machine et que si celle-ci ne choisit pas le bon, au moment de la postédition les modifications du texte généré peuvent être plus importantes que la simple modification de l'occurrence mal traduite.

exemple :

(1a) **los** datos almanecenados en memeoría se processan inmediateamente.

(1b) **la** información almanecenada en memoria se procesa inmediateamente.

Le second type d'interaction concerne les ambiguïtés syntaxiques.

exemple :

(2) This demonstrates the high efficiency and sophistication of the Series 50 systems.

is the text speaking of : high sophistication

(a) yes

(b) no

Le troisième type concerne le style que l'on veut produire en langue cible (ici l'allemand) et ne concerne donc pas les ambiguïtés.

exemple :

(3) The manuals supplied by IBM contain the needed information.

(a) extended participial modifier

(b) clause (der/die/das ... werden)

(c) clause (der/die/das ... sein)

(d) parallel to English

Dans cet exemple, (a) ou (b) sont des choix appropriés :

(3a) Die von IBM gelieferten Handbücher enthalten die benötigte Auskunft.

(3b) Die Handbücher, die von IBM geliefert werden, enthalten die benötigte Auskunft.

e. Conclusion

En répondant correctement aux questions lexicales, syntaxiques et stylistiques du système, le traducteur aide le système à produire un premier jet de traduction meilleur qu'il n'aurait été sans interaction.

Pour de tels outils, les dictionnaires sont plus petits (il y a besoin de moins d'information selon les auteurs). Cette petite taille peut être un facteur de choix. Le traducteur peut faire évoluer les performances du système en modifiant et en enrichissant les dictionnaires. Il peut aussi changer les paramètres du composant de traduction en les adaptant au type de document à traiter et à la forme de la traduction à produire.

Enfin, la traduction produite nécessite souvent d'être révisée, ce que l'on fait sous TSS.

1.2. L'interaction comme béquille ajoutée a posteriori

Après ces expériences pionnières, l'interaction à des fins de désambiguïsation a été intégrée de deux manières très différentes à des systèmes de traduction. Dans le premier cas, l'intégration est prévue dès la conception du projet. Dans le second, l'interaction n'est pas intégrée a priori, mais a posteriori, pour pallier les lacunes du système. Les projets KBMT et LMT que nous étudions maintenant sont l'illustration de ce second type de systèmes.

1.2.1. KBMT-89 (1987-1989)

a. Historique

Ce projet de maquette de système de traduction a vu le jour en 1987 suite à une décision prise par IBM de financer un projet de recherche d'une durée de deux ans au centre de recherche en traduction automatique de l'université Carnegie Mellon. L'objectif était la réalisation d'un prototype basé sur l'intégration des connaissances du domaine. Les spécifications furent développées par Carbonell et Tomita [Carbonell, *et al.* 1981 ; Carbonell & Tomita 1985 ; Carbonell & Tomita 1987] . Une démonstration de la première version eut lieu en février 1988. Le prototype final, appelé KBMT-89, [Nirenburg 1989 ; Nirenburg & al. 1989] fit l'objet d'une démonstration en février 1989.

KBMT-89 est une maquette de recherche et non un prototype industriel. Les deux langues du système sont l'anglais et le japonais, la traduction se faisant dans les deux sens. Le domaine couvert par le système concerne des manuels d'installation et de maintenance de micro-ordinateurs PC186 et PC3550 japonisés. Deux manuels (IBM 1983, 1986) ont constitué le corpus de base. Le système a été testé sur 300 phrases non-préédites.

b. Fiche d'identité

Nom du projet	KBMT-89
Langue(s) source(s)	anglais et japonais
Langue(s) cible(s)	anglais et japonais
Type de système	système de TAO interactive fondé sur la connaissance
Paradigme	interlingue
Architecture informatique	système distribué, grossièrement parallèle
Corpus	2 manuels d'installation et de maintenance de micro-ordinateurs, en japonais et en anglais

La connaissance dont dispose le système se compose de :

- un modèle du domaine (ontologie) de 1500 concepts ;
- deux lexiques d'analyse contenant 800 unités lexicales pour le japonais et 900 pour l'anglais ;
- deux lexiques de génération contenant 800 termes pour le japonais et 900 pour l'anglais ;
- une grammaire d'analyse pour l'anglais et une pour le japonais ;
- une grammaire de génération pour l'anglais et une pour le japonais.

Les formalismes développés pour le système sont :

- un système de représentation de la connaissance, FRAMEKIT ;
- un langage pour représenter les modèles du domaine ;

- un formalisme grammatical spécialisé basé sur la Grammaire Lexicale Fonctionnelle (LFG) ;
- un langage pour représenter les sens du texte (l'interlingue).

c. Structure du système

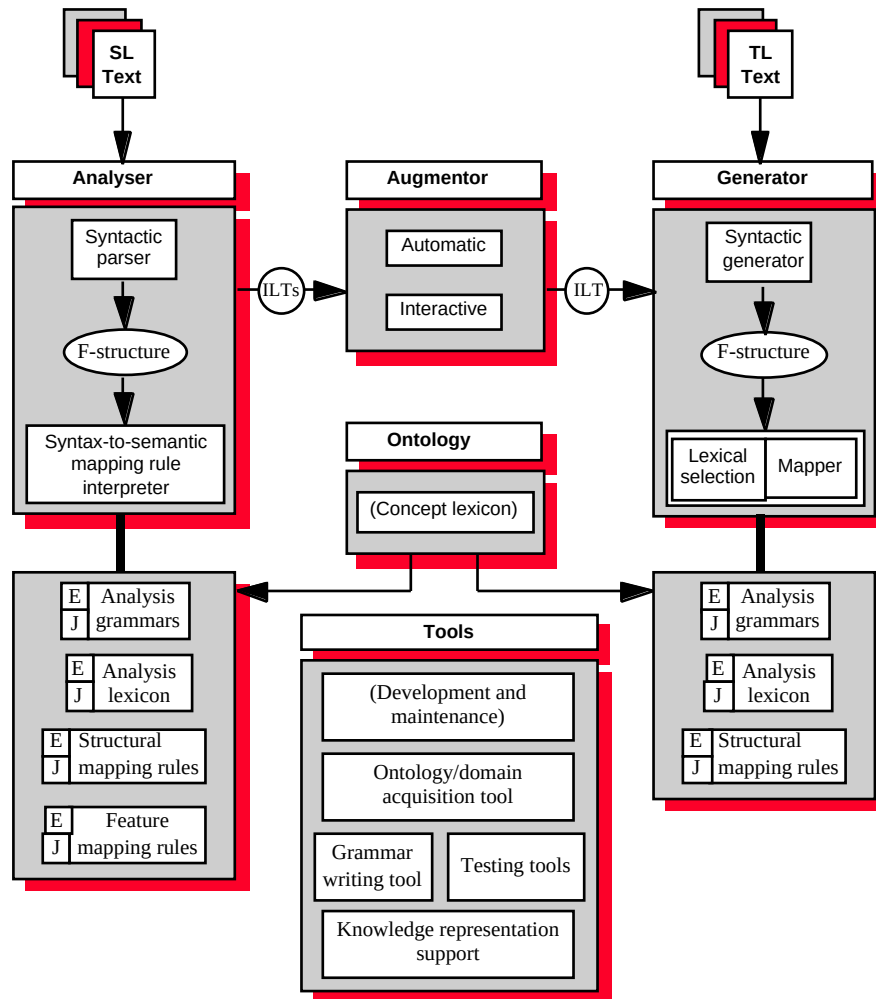


Figure 1.1 : Schéma général de KBMT-89

Le système inclut les composants suivants :

- un analyseur syntaxique avec un interpréteur de contraintes sémantiques ;
- un module d'application de contraintes sémantiques ;
- un désambigüiseur interactif : l'Augmentor ;
- un générateur sémantique produisant la structure syntaxique dans la langue source et effectuant la sélection lexicale ;
- un générateur syntaxique produisant le texte dans la langue cible.

Comme l'utilisation d'une ontologie ne suffisait pas à calculer toutes les informations nécessaires à la traduction, ni à résoudre toutes les ambiguïtés même de ce corpus très réduit, il fallut greffer un module de désambiguïstation interactive sur le système. Ce module de désambiguïstation interactive a profité des travaux de Carbonell (1983) sur la modélisation de l'utilisateur et de la réflexion de Church et Patil [Church & Patil 1982],

puis de Tomita [Tomita 1984 ; Tomita 1985 ; Tomita 1986] sur la désambiguïsation interactive et la TAO personnelle.

d. Désambiguïsation avec l'augmenteur

L'augmenteur [Brown 1989 ; Brown & Nirenburg 1990] sert de pont entre l'analyse et la génération. Comme son nom l'indique, il ajoute des informations dans la structure produite par l'analyseur. L'augmentation consiste à ajouter des valeurs manquantes, qui n'ont pas pu être calculées par l'analyseur, et à supprimer des ambiguïtés que l'analyseur n'a pas pu éliminer.

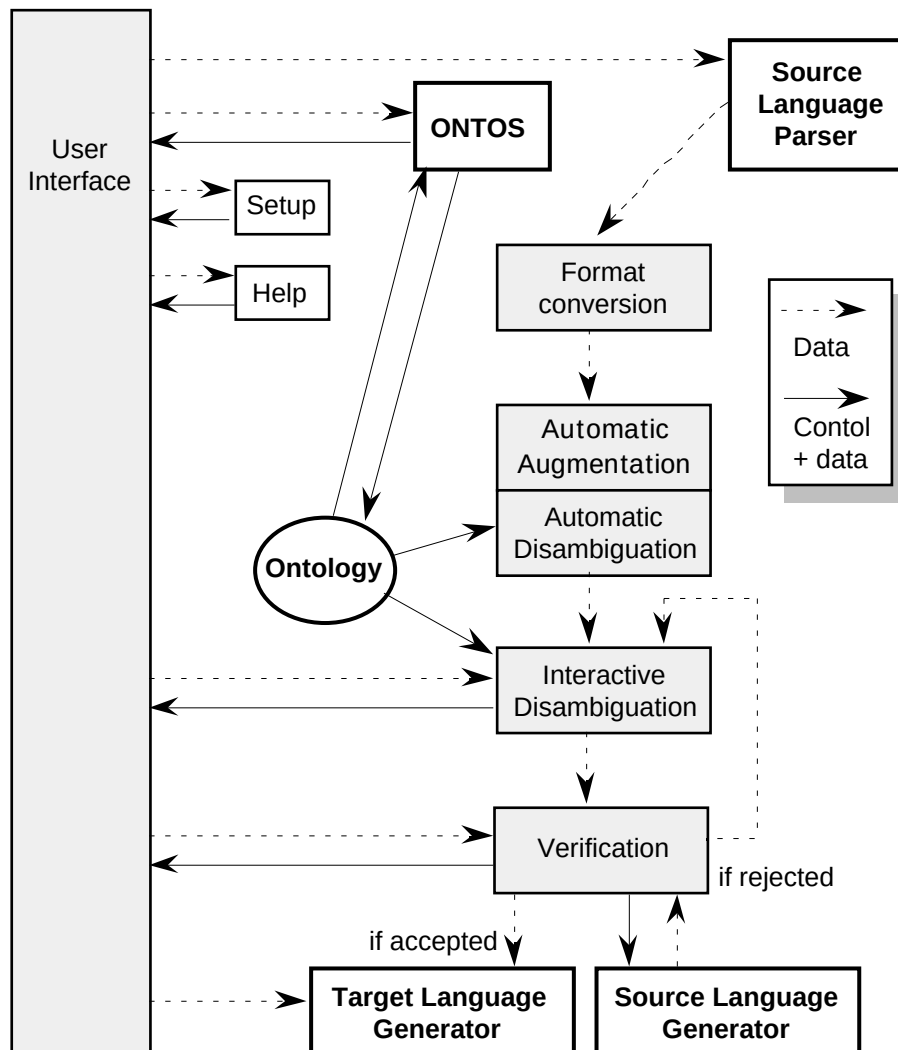


Figure 1.2 : Schéma de l'augmenteur de KBMT-89

L'augmenteur (figure 1.2) a trois composants principaux : un convertisseur de format, un augmenteur automatique et un augmenteur interactif. C'est ce dernier composant qui nous intéresse plus particulièrement ici. On y fait appel lorsqu'il manque encore des informations dans la structure ou que des ambiguïtés n'ont pu être résolues automatiquement. Le système demande alors à l'utilisateur de fournir les informations nécessaires et de choisir la bonne parmi les diverses interprétations possibles, cela jusqu'à obtenir une structure interlingue (InterLingua Text) non ambiguë.

i. Interface utilisateur [Goodman & Nirenburg 1991]

L'écran de l'augmenteur (figure 1.3) est divisé en deux parties, l'une réservée aux entrées/sorties et l'autre à l'affichage des questions. La première partie, située dans le bas de l'écran, est constituée de trois champs. Dans le champ supérieur, l'utilisateur saisit une phrase à traduire. Dans le champ intermédiaire, le système affiche dès que possible une paraphrase en langue source de la phrase origine. Si la paraphrase veut dire la même chose que la phrase origine, c'est que le système, en utilisant ou non l'augmenteur, a produit la bonne analyse. Le champ inférieur sert à l'affichage de la traduction de la phrase origine.

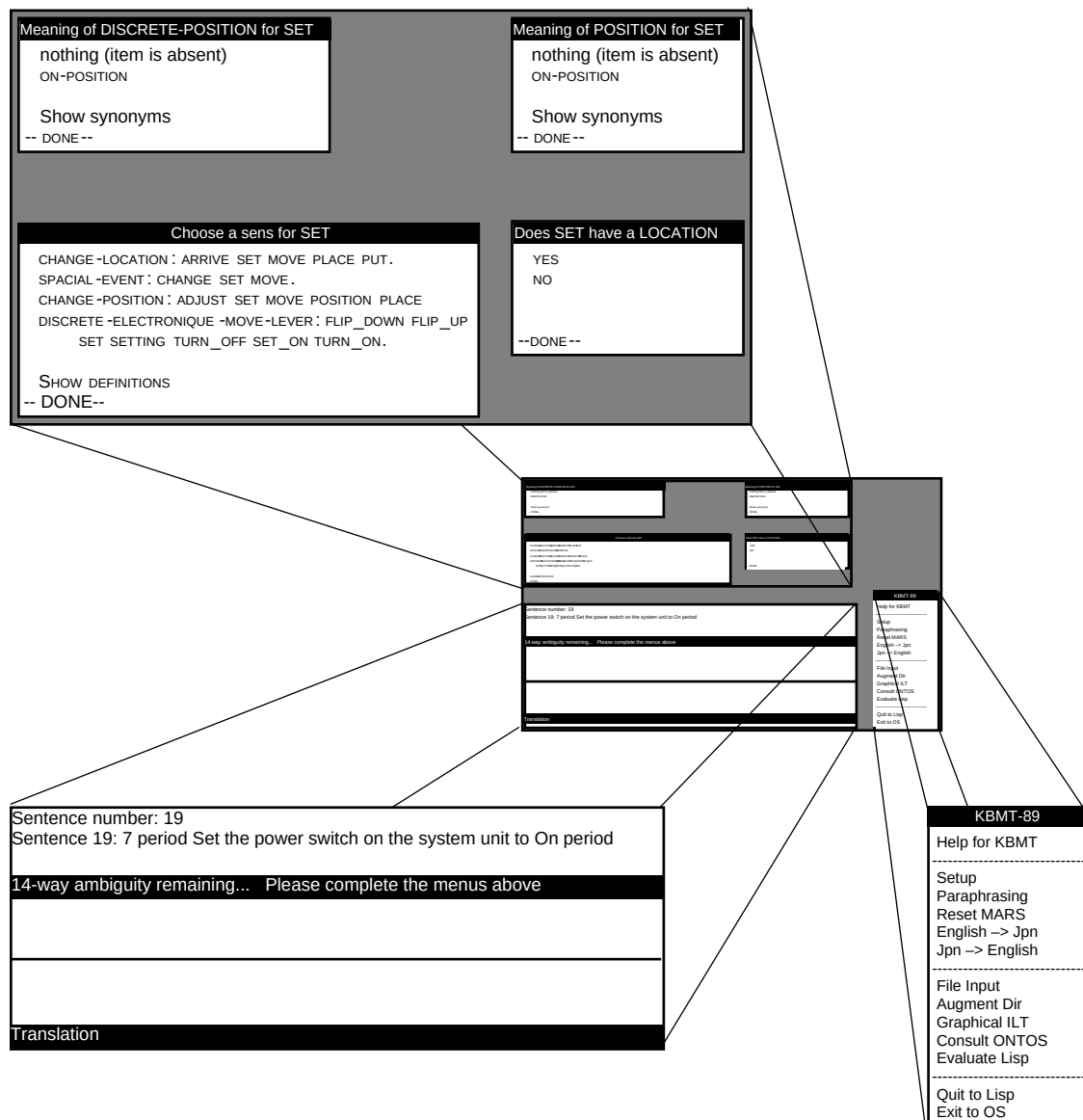


Figure 1.3 : Vue éclatée de l'écran de l'augmenteur

À droite de ces trois champs se trouve le menu qui permet de contrôler le système. Il permet de :

- demander de l'aide,
- régler les paramètres du système,

- choisir si oui ou non une paraphrase de confirmation doit être produite,
- réinitialiser le contexte du résolveur d'anaphores,
- lire le texte source dans un fichier,
- traiter toutes les analyses stockées dans un répertoire,
- voir une représentation graphique de la structure interlingue (ILT),
- consulter l'ontologie,
- évaluer une expression lisp arbitraire,
- sortir de KBMT-89.

Contrairement à la partie réservée aux entrées/sorties, le contenu de la partie réservée à l'affichage des questions évolue en reflétant l'état du dialogue entre l'augmenteur et l'utilisateur. Jusqu'à quatre fenêtres peuvent apparaître simultanément dans cette zone. Les types de problèmes que l'on peut résoudre sont : la sélection du sens d'une occurrence, la sélection du rôle d'un syntagme, la sélection d'un référent et la sélection du bon attachement.

ii. Utilisation de l'augmenteur

Au cours d'une traduction, s'il se trouve que l'augmentation automatique ne permet pas de calculer toutes les informations nécessaires, ni de résoudre toutes les ambiguïtés, des dialogues sont produits afin que l'utilisateur fournisse les informations manquantes. Lorsque plusieurs questions sont en suspens, l'utilisateur peut choisir la question à laquelle il souhaite répondre d'abord. Les autres questions sont alors actualisées en fonction de la réponse fournie. Trois types de questions ont été implémentés.

Menu de choix de sens : lorsque l'analyseur et l'augmenteur automatique n'ont pu déterminer lequel des divers sens d'une occurrence doit être choisi, ce menu présente une liste des sens. L'utilisateur doit choisir le sens approprié.

Menu de sélection du rôle : lorsque l'analyseur et l'augmenteur automatique n'ont pu déterminer le rôle que remplit un syntagme, tel que DESTINATION ou BUT, ce menu présente les rôles possibles. L'utilisateur doit choisir le rôle approprié.

Menu de sélection du référent : lorsque l'analyseur et l'augmenteur automatique n'ont pu déterminer un unique référent pour un pronom ou un nom, ce menu présente la liste des référents, construite au cours des analyses précédentes, parmi lesquels l'augmenteur n'a pu choisir. L'utilisateur doit donc choisir le référent approprié.

- Voici un exemple de session avec l'augmenteur :

D'abord on entre la phrase à traduire, ici : "7. Set the Power switch on the system unit to ON." L'augmenteur transmet la phrase à l'analyseur et attend le résultat. L'augmenteur affiche le message "Parsing..." dans la ligne de statut (figure 1.4 ci-dessous).

Lorsque l'augmenteur a lu les analyses candidates, il les traduit de la forme, en liste de listes, produite par l'analyseur en un ensemble isomorphe de cadres (frames) FRAMEKIT. Il procède ensuite à l'augmentation en rendant explicites des informations implicites et en effectuant des réorganisations structurales. Cela est indiqué dans la ligne de statut en même temps que le nombre d'analyses en cours de traitement (figure 1.5).

Sentence number: 19 Sentence 19: 7 period Set the power switch on the system unit to On period
Parsing...
Translation

Figure 1.4 : Analyse de la phrase d'entrée avec KBMT-89

Sentence number: 19 Sentence 19: 7 period Set the power switch on the system unit to On period
Augmenting 14 candidates
Translation

Figure 1.5 : Traitement des analyses produites : augmentation automatique

Meaning of DISCRETE-POSITION for SET nothing (item is absent) ON-POSITION Show synonyms -- DONE--	Meaning of POSITION for SET nothing (item is absent) ON-POSITION Show synonyms -- DONE--
Choose a sens for SET CHANGE -LOCATION : to change the location of. SPACIAL -EVENT : Event whose focus is spacial. CHANGE -POSITION : to change the spacial orientation of. DISCRETE -ELECTRONIQUE -MOVE -LEVER : e.g. flip a power. Show synonyms -- DONE--	Does SET have a LOCATION YES NO --DONE--
Sentence number: 19 Sentence 19: 7 period Set the power switch on the system unit to On period	
14-way ambiguity remaining... Please complete menus above	
Translation	

Figure 1.6 : Affichage initial de quatre menus

Meaning of DISCRETE-POSITION for SET nothing (item is absent) ON-POSITION Show synonyms -- DONE--	Meaning of POSITION for SET nothing (item is absent) ON-POSITION Show synonyms -- DONE--
Choose a sens for SET CHANGE -LOCATION : ARRIVE SET MOVE PLACE PUT. SPACIAL -EVENT : CHANGE SET MOVE. CHANGE -POSITION : ADJUST SET MOVE POSITION PLACE DISCRETE -ELECTRONIQUE -MOVE -LEVER : FLIP_DOWN FLIP_UP SET SETTING TURN_OFF SET_ON TURN_ON. Show definitions -- DONE--	Does SET have a LOCATION YES NO --DONE--
Sentence number: 19 Sentence 19: 7 period Set the power switch on the system unit to On period	
14-way ambiguity remaining... Please complete menus above	
Translation	

Figure 1.7 : Recherche dans le lexique des sens du verbe “set”

Meaning of DISCRETE-POSITION for SET nothing (item is absent) ON-POSITION Show synonyms -- DONE--	Meaning of POSITION for SET nothing (item is absent) ON-POSITION Show synonyms -- DONE--
Does SET have a CIRCUIT YES NO --DONE--	Does SET have a LOCATION YES NO --DONE--
Sentence number: 19 Sentence 19: 7 period Set the power switch on the system unit to On period	
6-ways ambiguity remaining... Please complete menus above	
Translation	

Figure 1.8 : Nouveaux menus après la désambiguïsation du verbe “set”

Après l'augmentation et la désambiguïsation automatique, les ambiguïtés résiduelles donnent lieu à une session de désambiguïsation interactive.

Dans ce cas, quatre menus apparaissent, indiquant qu'il y a au moins quatre points pour lesquels les candidats diffèrent (figure 1.6). Nous allons travailler avec le menu inférieur gauche d'abord parce qu'il contient le plus grand nombre d'items et que l'on pense donc qu'il va permettre de résoudre l'ambiguïté plus rapidement.

Avant de choisir "discrete-electronic-move-lever" comme sens approprié pour "set", on peut demander à voir les synonymes qui apparaissent dans le lexique pour chacun des sens possibles (figure 1.7).

Lorsqu'on a effectué la sélection et cliqué sur "DONE", l'augmenteur élimine les analyses qui ne correspondent pas au choix effectué. Le nombre des analyses pertinentes passe de 14 à 6, comme indiqué dans la ligne de statut. Un nouveau dialogue remplace celui auquel on vient de répondre (figure 1.8). On peut aussi choisir plusieurs items dans un menu si l'on n'est pas sûr de choisir le bon.

On choisit maintenant "on-position" dans le menu supérieur gauche ; cela réduit le nombre de candidats à deux et fait disparaître trois des menus, car deux des autres menus étaient dépendants du menu supérieur gauche (figure 1.9).

Does POWER and SWITCH have a PART-OF	
YES NO	
--DONE--	
Does SET have a LOCATION	
YES NO	
--DONE--	
Sentence number: 19 Sentence 19: 7 period Set the power switch on the system unit to On period	
2-way ambiguity remaining... Please complete menus above	
Translation	

Figure 1.9 : Nouveaux menus après la désambiguïsation de "on"

Lorsque l'on choisit "YES" dans le menu supérieur gauche (figure 1.9 ci-dessus), il ne reste plus qu'une analyse. Elle est alors transmise au générateur de l'anglais pour paraphrasage (figure 1.10).

Sentence number: 19 Sentence 19: 7 period Set the power switch on the system unit to On period
Paraphrasing results...
Translation

Figure 1.10 : Génération d'une paraphrase pour vérification après désambiguïsation

Ici, l'analyseur n'est pas capable de générer à partir de la structure ILT ; et l'augmenteur montre la structure ILT brute dans la fenêtre centrale (figure 1.11). On répond par l'affirmative lorsque le système demande si la paraphrase est correcte.

Sentence number: 19 Sentence 19: 7 period Set the power switch on the system unit to On period Is the paraphrase correct?
Please verify the paraphrase below
Unable to generate from this ILT: [clause500 (\$SENTENCE\$ Sentence 19: 7 period Set the Power switch on the system unit to On periode) (MAIN-CLAUSE YES) (NUMBER-BULLET
Translation

Figure 1.11 : Vérification de la paraphrase produite à partir de l'analyse désambiguïsée

Le processus se termine et la traduction apparaît (figure 1.12).

Sentence number: 19
Sentence 19: 7 period Set the power switch on the system unit to On period
Is the paraphrase correct?y
Sentence number:

Please verify the paraphrase below

Unable to generate from this ILT:
[clause500
(\$SENTENCE\$ Sentence 19: 7 period Set the Power switch on the system unit to On periode)
(MAIN-CLAUSE YES)
(NUMBER-BULLET

Translation: @7 . sisutemu yunitto@NO den-gen- suitti@WO ire@TE@KU@DA@SA@I

Translation

Figure 1.12 : Affichage de la traduction de la phrase d'entrée

Si la paraphrase avait été jugée incorrecte, le système aurait demandé de répondre à une nouvelle série de questions. Cette fois, toutes les analyses écartées par le processus de désambiguïsation automatique sont conservées au départ.

Cela permet de faire des choix différents et corriger d'éventuelles erreurs produites pendant le processus de désambiguïsation automatique.

e. Conclusion

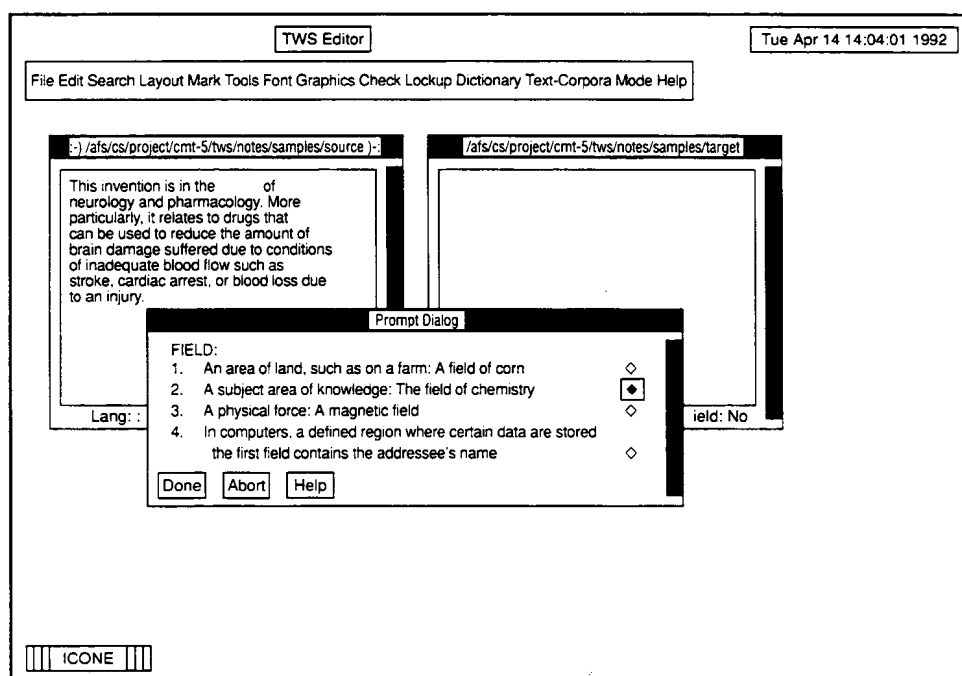


Figure 1.13 : Une vue de TWS

Pour répondre aux questions posées lors de la phase de désambiguïsation interactive, l'utilisateur n'a pas besoin d'avoir des connaissances de linguiste. Le dialogue, en langue source, est géré au moyen de menus qui lui permettent de choisir la bonne solution parmi

celles qui sont proposées. Il est cependant difficile de répondre aux questions sans entraînement.

Par rapport aux autres systèmes que nous connaissons, KBMT-89 est, avec JETS et ITS2, le seul qui ait une interface moderne. Nous avons aussi apprécié les dialogues en parallèle qui s'actualisent en fonction de la réponse courante.

Les travaux sur la traduction fondée sur la connaissance se poursuivent au Centre for Machine Translation de CMU avec le système KANT [Mitamura 1991].

D'autre part, l'expérience de KBMT a été intégrée dans le projet TWS (Translator's Workstation) [Nirenburg 1992] de station de travail du traducteur. La figure 1.13 illustre l'intégration des acquis de KBMT dans TWS .

1.2.2. LMT (1989-)

a. Historique

LMT (Logic-based Machine Translation) [McCord 1989a ; McCord 1989b] est un système de traduction automatique expérimental développé par M. McCord au centre de recherche Thomas J. Watson d'IBM pour tester l'utilisation de la programmation logique en traitement automatique des langues naturelles. Le système a connu deux stades de développement majeurs, l'un en 1989 et l'autre en 1991.

L'analyse de l'anglais utilise un formalisme de grammaire logique, la grammaire logique "à cases", qui permet aux grammaires d'être plus compactes, et qui intègre un traitement modulaire de la syntaxe, du lexique, et de la sémantique. La grammaire est écrite indépendamment de l'objectif de traduction.

LMT utilise un transfert syntaxique, bien que l'arbre d'analyse syntaxique de l'anglais contienne des résultats de choix sémantiques et des relations grammaticales profondes. La vérification des types sémantiques est réalisée durant les phases d'analyse et de transfert grâce au mécanisme d'inférence de Prolog.

L'algorithme de transfert utilise les variables logiques et l'unification. Après la phase de transfert, le composant de génération syntaxique de l'allemand produit un arbre de surface par application d'un système de transformation d'arbres. Ces transformations mettent en œuvre une version augmentée du mécanisme de reconnaissance de forme de Prolog. LMT utilise un seul lexique qui contient des informations sur les langues source et cible ainsi que des informations morphologiques et idiosyncratiques de l'allemand.

Entre 1989 et 1991, IBM a développé un environnement, LMT, pour engager des coopérations avec d'autres centres de recherche IBM en Allemagne, Espagne, Israël et au Danemark. Ces coopérations ont produit plusieurs résultats [Rimon, *et al.* 1991], dont la production d'un composant de désambiguïsation interactive [Ben-Ari, *et al.* 1988].

Les recherches sur LMT sont réactivées chez IBM en France⁹.

⁹ Le 28 Octobre 1993, on pouvait lire, dans la liste LN, une contribution de Jean-Marc Lange : "Nous poursuivons le développement du prototype anglais-français du système de traduction LMT. A cet effet, nous recherchons pour un contrat de 7 mois un(e) ingénieur linguiste possédant les qualités suivantes : - bonne connaissance de Prolog; - excellente connaissance des langues anglaise et française (lexique, grammaire, et aspects contrastifs) dans le domaine technique; - si

b. Fiche d'identité

Nom du projet	LMT (Logic-based Machine Translation)
Couples de langues	anglais $\leftrightarrow$ allemand allemand $\leftrightarrow$ anglais, anglais $\leftrightarrow$ espagnol danois $\leftrightarrow$ anglais, anglais $\leftrightarrow$ danois
Type de système	système autonome puis système interactif
Paradigme	seconde génération (analyse-transfert-génération) utilisation de la programmation logique (Prolog). Analyse basée sur les grammaires à cases [McCord 1980]. Une large portion de l'outil est indépendante des langues source et cible, formant ainsi un environnement de développement de différents systèmes
Corpus	LMT-89 a été testé sur un corpus de 500 phrases extraites du manuel de XEDIT. 95% des traductions produites pouvaient être lues et comprises (dans le sens de la langue source) par un lecteur allemand

c. Désambiguïsation avec LMT

La technique proposée pour la désambiguïsation interactive est fondée sur trois suppositions :

- L'ambiguïté d'une phrase est souvent la conséquence d'un équilibre délicat entre les mots auxquels peuvent être associées différentes parties du discours, et des informations implicites cachées qui peuvent être retrouvées de différentes manières. Dans les deux cas, un changement mineur qui bouscule cet équilibre ou explicite les informations cachées produit une phrase non ambiguë qui préserve l'un des sens possibles de la phrase originale.
- Il est facile pour un lecteur humain de décider si oui ou non une paraphrase préserve le sens de la phrase originale. De plus, si la phrase originale n'est pas ambiguë pour un humain, la paraphrase du mauvais sens conduit dans beaucoup de cas à un non-sens qui est facilement identifié.
- Il n'est pas nécessaire que le système comprenne complètement la source de l'ambiguïté pour l'identifier et décider si oui ou non elle va être préservée par la traduction. Comprendre est nécessaire pour choisir la bonne interprétation, une tâche qui est laissée à l'utilisateur.

Ainsi, une phrase est d'abord analysée avec un analyseur général de la langue source. En fonction des différents arbres d'analyse produits par l'analyseur, de modèles spéciaux stockés dans le module de désambiguïsation, et d'informations syntaxiques issues des

dictionnaires bilingues et de la langue cible, les ambiguïtés potentielles sont identifiées. La *préservabilité* de l'ambiguïté est vérifiée. Si l'on décide que l'ambiguïté n'est pas préservable, la phrase est paraphrasée, et les paraphrases sont présentées à l'utilisateur. En fonction de la réponse de l'utilisateur, l'analyse appropriée est transmise aux phases suivantes du processus de traduction.

Pour l'instant, on traite quatre types d'ambiguïtés, les autres étant plus délicates à identifier et à paraphraser.

1. l'ambiguïté de partie du discours,

• *Définition* :

Plusieurs parties du discours peuvent être affectées à une occurrence.

• *exemples* :

(1) The management **requests control** information.

(2) **Visiting grandfather** will take up the whole day.

2. l'ambiguïté de structure,

• *Définition* :

Les occurrences de la phrase ont une partie du discours unique, mais les attachements peuvent être effectués de plusieurs manières.

• *exemples* :

(3) Businessmen who are afraid to take risks **frequently** lose out their competitors.

(4) **Good** boys and girls go to heaven.

3. l'ambiguïté de sens (polysémie),

• *Définition* :

Recouvre au sens strict la polysémie et l'homonymie.

• *exemples* :

(5) Dan walked to the **bank**.

(6) The thief stole two million dollars from the **bank**.

4. certaines ambiguïtés implicites.

• *Définition* :

L'ambiguïté implicite n'apparaît pas dans la géométrie des analyses, elle résulte de l'anaphore, des trous, et d'autres informations cachées qui peuvent être retrouvées de différentes manières.

• *exemples* :

(7) The chickens are ready to **eat**.

(8) The stone hit the shop window and **it** broke.

La procédure de production des paraphrases se divise en quatre étapes :

1. identification de l'ambiguïté,

2. test de préservabilité de l'ambiguïté,

3. production des paraphrases, et
4. évaluation de la réponse de l'utilisateur et sélection du sens approprié.

Pour chaque type d'ambiguïté, les étapes de l'algorithme sont quelque peu différentes. Dans les paragraphes suivants, les différentes étapes de l'algorithme et leur application à chaque type d'ambiguïté sont discutées.

Comme ce travail est effectué dans le contexte du projet multilingue, il est intéressant de savoir quelles sont les parties de l'algorithme qui sont indépendantes de la langue cible, ou en d'autres termes celles qui sont partageables par les projets qui ont en commun la même langue source et celles qui sont particulières à un couple de langues.

Il est réconfortant de voir que certaines parties sont complètement générales. Le résultat de l'étude de généricité est fourni dans le tableau suivant.

	Identification	Test	Paraphrasage	Évaluation
partie du discours	Générale	Paire	Source	Générale
Structure	Générale	Paire	Source	Générale
Sens des mots	Paire	–	Paire	Générale
Implicite	Paire	Paire	Source	Paire

i. Identification des ambiguïtés

La première étape du processus de désambiguïsation est l'identification des ambiguïtés. Les algorithmes d'identification des ambiguïtés de catégorie et de structure sont communs à toutes les ambiguïtés de chacune des deux classes (i.e. les paires de langues impliquées, e.g. nom-verbe, adjectif-verbe, etc., n'est pas codé directement dans l'algorithme). Par contre, pour l'identification du sens des mots et les ambiguïtés implicites, chaque cas doit être connu par avance et traité séparément. Bien sûr, comme c'est très souvent le cas en traitement des langues naturelles, écrire un système à large couverture nécessite beaucoup de travail.

Identification des ambiguïtés de partie du discours

Un parcours en parallèle des feuilles des arbres obtenus permet de construire une liste de tous les mots auxquels ont été affectées plusieurs parties du discours.

Identification des ambiguïtés de structure

Un parcours en parallèle de deux arbres d'analyse de bas en haut permet de localiser la première différence entre les arbres.

Identification des ambiguïtés sur le sens des mots

Dans le cadre du projet, on utilise un dictionnaire bilingue. Le choix du mot en langue cible est effectué en fonction de son environnement syntaxique dans l'arbre

d'analyse syntaxique. Lorsque l'on saisit une entrée pour un mot, si deux sens ne peuvent être différenciés par des critères syntaxiques, il y a ambiguïté.

- (9) He **held** a radical view.
 (10) He **held** a radical conference.

Dans les deux cas, "held" possède un objet direct non-humain et non-concret, mais la traduction en hébreu du verbe "hold" est différente dans les deux cas. Si le verbe "hold" est identifié dans cet environnement syntaxique, une ambiguïté est identifiée.

Ce traitement mot à mot nécessite un travail considérable. Dans le projet, le dictionnaire bilingue joue un rôle majeur, et chaque entrée requiert un travail précis de recensement des environnements syntaxiques qui imposent la sélection d'une certaine traduction. Ainsi, inventorier les configurations dans lesquelles aucune sélection automatique ne peut être effectuée est une partie assez aisée du travail de création du dictionnaire.

Identification des ambiguïtés implicites

Des patrons potentiellement ambigus sont définis. En voici quelques exemples :

verb(base=be) adj infinitival-clause

The chicken **are ready to eat**.

np² np* noun

John presented his paper in the **desert langage society symposium**.

adj np² np* noun

I take part in the **artificial intelligence project**.

On recherche la présence de ces patrons dans l'arbre d'analyse, ce qui n'est pas une tâche simple.

ii. Test de préservabilité

Lorsqu'une ambiguïté est identifiée, on vérifie sa préservabilité dans la langue cible. Il faut noter que cette préservabilité dépend fortement de la paire de langues considérée.

Préservation de l'ambiguïté de partie du discours

Quelques parties du discours, telles que les adjectifs et verbes en "ing", qui sont interchangeables en anglais le sont aussi en hébreu.

- (11) The book is **interesting**.

Seules les parties du discours conflictuelles sont traitées.

Préservation de l'ambiguïté de structure

Quelques ambiguïtés de structure, telles que de nombreux cas d'attachement prépositionnel, peuvent être préservées.

L'ambiguïté du fameux exemple (12) "I saw a man in the park with a telescope." est préservée si l'on traduit vers le français. Par contre, la même ambiguïté ne peut être

préservée lorsque l'on traduit vers l'hébreu. Le "with" de la phrase précédente est traduit par le préfixe "Be" si le télescope est un instrument du verbe voir et par le mot "IM" sinon.

Un autre exemple concerne le double rôle de "**that**" qui peut introduire une proposition relative ou un complément.

(13) We signalled the guide **that** we could not hear.

L'hébreu possède un préfixe "SHE" qui peut être utilisé dans les deux sens.

Préservation de l'ambiguïté de sens

À cause des techniques utilisées pour identifier une ambiguïté de sens, ces ambiguïtés ne peuvent pas être préservées. En effet une telle ambiguïté est identifiée en consultant un dictionnaire de transfert propre à chaque paire de langues.

Préservation de l'ambiguïté implicite

Beaucoup de patrons ne sont jamais préservés. Ceux qui peuvent l'être, parfois, possèdent leur propre algorithme de test. Par exemple, une référence implicite par un pronom peut être préservée si, en hébreu, le remplaçant du nom a le même genre et le même nombre que le nom anglais.

iii. Paraphrasage

Lorsque le système est confronté à une ambiguïté qui ne peut être préservée, il produit une paraphrase. Quand c'est possible, deux paraphrases peuvent être produites pour deux lectures différentes afin de mettre cette différence en relief. Dans un tel cas, l'utilisateur doit choisir la paraphrase qui convient. Si une seule paraphrase peut être produite, l'utilisateur doit dire si l'intention originale est préservée ou non.

Paraphrasage de l'ambiguïté de partie du discours

Selon le type de conflit de parties du discours que l'on observe pour un mot ou une suite de mots, une ou deux paraphrases sont générées.

Dans l'exemple 1, on identifie le conflit suivant :
 $N_1 V_2 N_3 V_4$ contre $N_1 N_2 V_3 N_4$

Les paraphrases construites sont :
the N₁ V₂ the V₄ contre *the N₂ V₃ the N₄*

Cela donne finalement ;

(14) The management requests the information

(15) The requests control the information

Paraphrasage de l'ambiguïté de structure

Selon les différences de structure qui se présentent entre les deux arbres, une ou deux paraphrases sont générées. Dans l'exemple (3), où l'adverbe peut être attaché soit à la proposition principale, soit à la clause subordonnée, l'ambiguïté peut être évitée avec une ponctuation correcte. Ainsi, les deux possibilités sont présentées à l'utilisateur :

(16) Businessmen who are afraid to take risks, frequently lose out their competitors.

(17) Businessmen who are afraid to take risks frequently, lose out their competitors.

Pour l'exemple 4, l'adjectif est distribué sur les deux groupes nominaux :

(18) Good boys and good girls go to heaven.

Paraphrasage de l'ambiguïté de sens

Une entrée du dictionnaire a un synonyme pour chaque sens qui ne peut être désambiguïsé par un environnement syntaxique. Le mot ambigu est remplacé par les synonymes possibles. Pour l'exemple (10), les paraphrases obtenues sont :

(19) He convened a radical conference.

(20) He maintained a radical conference.

Bien que la désambiguïsation sur le sens des mots soit effectuée à partir du dictionnaire bilingue (écrit par des experts de la langue source et de la langue cible), les synonymes présentés à l'utilisateur sont dans la langue maternelle.

Paraphrasage de l'ambiguïté implicite

À chaque patron d'ambiguïté implicite est associé un algorithme de paraphrasage. Pour le patron verb(base=be) adj infinitival-clause, dans l'exemple 7, la proposition infinitive est mise au passif.

(21) The chicken are ready to be eaten.

On peut noter que pour le même patron avec un verbe intransitif, comme dans l'exemple (22), un non-sens est généré, comme on le montre dans l'exemple (23).

(22) John is ready to sleep.

(23) John is ready to be slept.

iv. Évaluation de la réponse de l'utilisateur

Si le système présente au traducteur humain une paraphrase, celui-ci doit décider si le sens de la phrase originale est conservé. Si le système présente deux paraphrases, l'utilisateur doit choisir celle qui préserve le sens du texte. La réponse est ensuite transmise au système.

Évaluation pour une ambiguïté de catégorie

Les arbres d'analyse ne contenant pas le bon étiquetage en parties du discours sont laissés de côté.

Évaluation pour une ambiguïté de structure

Les arbres ne possédant pas la bonne structure sont écartés.

Évaluation pour une ambiguïté de sens

La traduction correcte est attachée au mot en question.

Évaluation pour une ambiguïté implicite

L'information explicite est attachée aux bons nœuds (e.g. nombre et genre des pronoms, etc.).

d. Conclusion

Les auteurs de ce désambiguïseur évaluent ainsi leur approche [Ben-Ari, *et al.* 1988] :

«L'une des techniques utilisées dans les projets de traduction automatique pour « maîtriser le problème de l'ambiguïté est de construire un module de désambiguïsation « interactive. Beaucoup d'ambiguïtés peuvent être préservées durant la traduction. Les « ambiguïtés qui ne peuvent être transportées doivent être identifiées, puis présentées au « traducteur humain de telle façon qu'il puisse effectuer une désambiguïsation interactive. « Nous avons utilisé une technique de paraphrasage. Les quatre phases principales du « processus de désambiguïsation interactif, identification, test de préservabilité, « paraphrasage et évaluation ont été discutées dans le contexte de quatre types « d'ambiguïtés : partie du discours, structure, sens des mots et implicite.

«Nous sommes conscients que la méthode présentée doit être inapplicable pour « certains cas d'ambiguïté, et que même lorsqu'elle s'applique, nous avons besoin « d'algorithmes plus nombreux. Néanmoins, nous pensons que les principes utilisés « établissent une bonne base pour le développement d'un composant qui filtre beaucoup « de cas de mauvaises interprétations. Un tel composant doit permettre d'améliorer la « qualité des traductions produite par un système de traduction automatique.»

Nous avons eu connaissance très tardivement de ces travaux, les bases de nos idées sur le processus de désambiguïsation étaient déjà posées. nos idées sont assez proches de celles qui ont été développées dans LMT. Nous reprenons bien volontiers à notre compte les trois principes qui motivent la démarche (début du § c.).

Les outils de paraphrasage que nous utilisons sont plus sommaires. Nous ne pouvons pas, par exemple, produire la paraphrase de l'exemple (2). Par contre, nous ne construisons pas de paraphrases absurdes au sens de l'exemple (23). Dans la présentation de leur méthode, les auteurs ne semblent pas imaginer que l'analyseur puisse produire plus de deux arbres. Une différence majeure réside en ce que nous procédons par différenciation, c'est à dire que toutes les interprétations de la phrase sont présentées à l'auteur.

1.3. L'interaction comme ingrédient de base

À côté des projets KBMT et LMT, dans lesquels l'interaction est rajoutée après coup, pour remédier aux faiblesses du système, d'autres projets ont intégré l'interaction dès la phase de conception. L'architecture informatique, ergonomique et linguistique de ces systèmes tient donc compte de leur dimension interactive d'une façon beaucoup plus profonde et adéquate.

1.3.1. DLT (1982-1989¹⁰)

a. Historique

Distributed Language Translation (DLT) est le nom d'un projet de Traduction Automatique conduit par la société de services informatiques Buro voor Systeemontwikkeling (BSO) aux Pays-Bas. Les études préliminaires commencèrent en 1979 sous la direction de Toon Witkam. Une étude de faisabilité fut soutenue par la Commission des Communautés Européennes entre 1982 et 1983 [Witkam 1983].

En 1985, BSO obtint un contrat de 6 ans avec le ministère néerlandais de l'Économie. L'objectif premier était la mise au point pour 1987 d'un prototype anglais-français et l'obtention d'un système commercial pour 1993. L'objectif à long terme du projet DLT était de construire un système pour faire des traductions entre les langues européennes (anglais, français, allemand, italien). Le prototype, écrit en Prolog, est conçu pour traduire un sous-ensemble de l'anglais vers le français. La démonstration eut lieu en décembre 1987 [Witkam 1988]. Depuis cette date, l'équipe de DLT s'est mise à travailler sur une version basée sur des principes totalement différents de ceux du prototype, sans passer au développement d'un système commercial. Ce manque de cohérence fait que, à l'arrêt du projet en 92-93, il n'y avait ni version commerciale de la première approche, ni prototype de la seconde.

La meilleure façon de décrire le système DLT est de le considérer comme un ensemble cohérent de systèmes intégrés destinés à faciliter la génération et l'affichage des textes utilisés en connexion avec des systèmes de traitement de textes et des terminaux vidéotex. DLT est conçu comme un système interactif multilingue fonctionnant sur un réseau d'ordinateurs. Chaque ordinateur se comporte comme un traducteur depuis et vers une langue seulement. Les textes sont transmis à travers le réseau dans un langage intermédiaire : une partie de l'espéranto muni de marques cachées (parenthèses spéciales). DLT se voulait un outil pour la communication interlingue de "textes d'information" (résumés, rapports, manuels) ou de messages commerciaux. DLT est par essence un système distribué, puisque les processus d'analyse et de génération ne sont pas exécutés sur le même terminal.

b. Fiche d'identité

Le projet DLT vise à traduire au moindre coût diverses langues européennes (en commençant par le français, l'anglais et l'allemand) ; les auteurs estiment qu'il pourra être facilement étendu à d'autres langues (japonais, hébreu, arabe).

Nom du projet	DLT (Distributed Language Translation)
Langue(s) source(s)	français, anglais, allemand, italien
Langue(s) cible(s)	français, anglais, allemand, italien
Type de système	outil de communication multilingue interactif

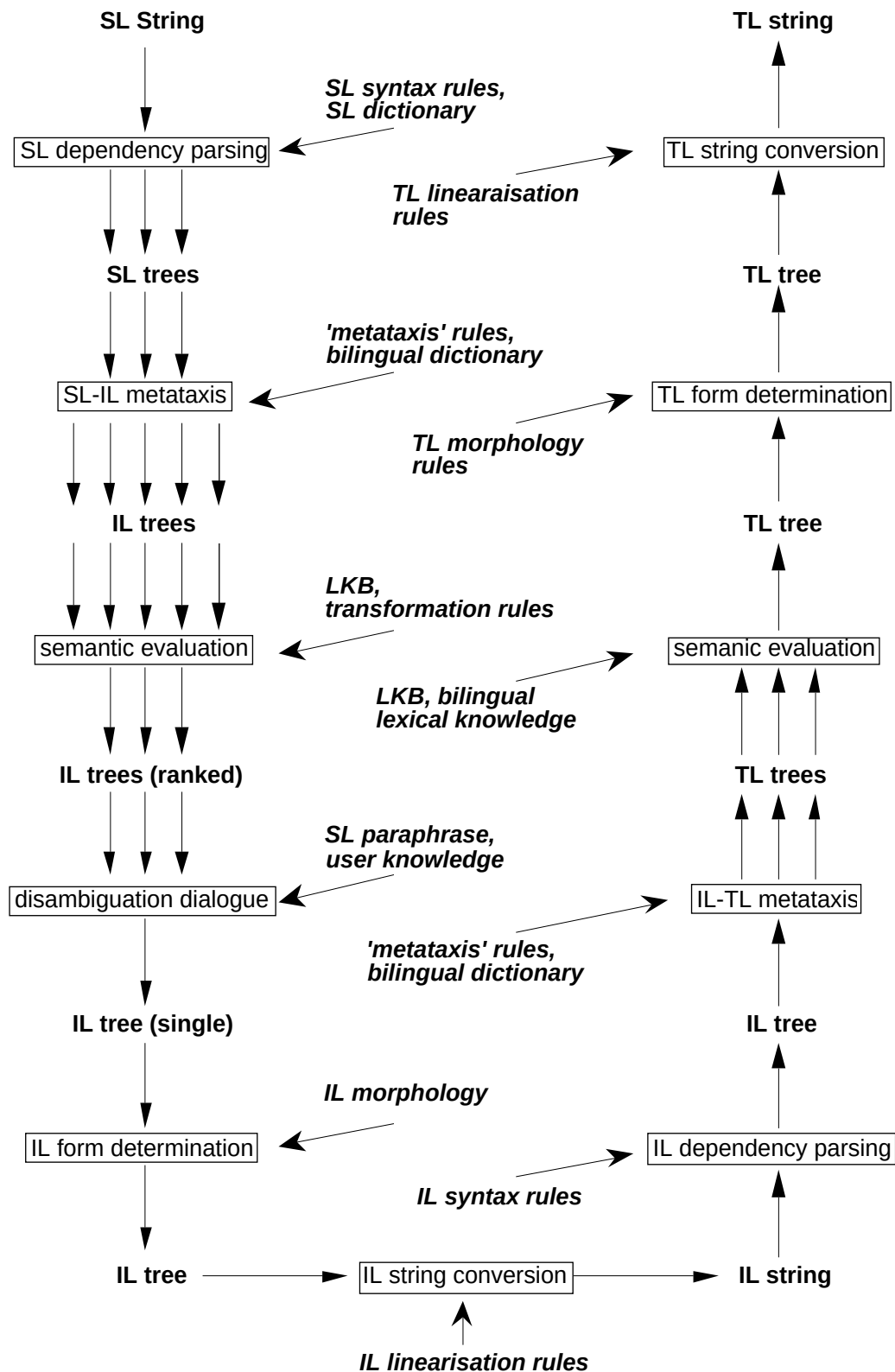
¹⁰ Depuis 1989, les progrès ont été minimes, le projet ayant été "gelé" (communication personnelle de Toon Witkam).

Paradigme	interlingue. L'interlingue est un sous-ensemble de l'espéranto avec des marques cachées.
Architecture	distribuée ; ordinateurs ou terminaux connectés sur un réseau. La station émettrice a en charge le codage du texte source en Interlingue, la station réceptrice le décodage de l'Interlingue vers une langue cible. Un utilisateur monolingue saisit un texte (en anglais par exemple) depuis un terminal sur lequel le texte est immédiatement traduit dans le langage intermédiaire. L'analyse et la traduction en espéranto se font en temps réel ; le système essaie de traduire le texte dès qu'il est saisi, que les phrases soient complètes ou non. Un dialogue entre l'auteur et le système permet de résoudre les ambiguïtés que le système rencontre
Corpus	– textes d'information : les textes qui seront traités peuvent se ranger dans la catégorie des "textes d'information" allant des manuels d'instructions techniques jusqu'aux résumés de textes scientifiques, et des rapports commerciaux jusqu'aux règlements sur le retraitement et le traitement des déchets nucléaires – langage normatif : les textes doivent respecter certaines normes : ils doivent utiliser un langage correct et conventionnel ; l'argot, les familiarités, les expressions de la langue parlée doivent être bannis

c. Structure du système

Le processus de traduction se décompose en 16 étapes [Hutchins & Somers 1992 pp 279-296 ; Schubert 1988] .

- 1 analyse de la langue source avec un Augmented Transition Network (ATN),
- 2 transformation monolingue d'arbre (sur la langue source) : réduction à une forme canonique des variantes monolingues. Can't, cannot vers can not,
- 3 transformation bilingue d'arbre (metataxis): remplacement des mots anglais par leur équivalent en espéranto, remplacement des étiquettes de dépendance syntaxique anglaise par celles de l'espéranto,
- 4 choix des mots interlingues en utilisant des critères syntaxiques et pragmatiques, ainsi que la LKB (Linguistic Knowledge Bank),
- 5 dialogue de désambiguïsation,
- 6 transformation monolingue d'arbre (sur l'interlingue) : détermination des traits morphologiques pour une représentation interlingue correcte,
- 7 libéralisation de la forme interlingue : choix de l'ordre des mots, production d'un texte en interlingue,
- 8 vérification de correction syntaxique du texte interlingue,
- 9 codage et transmission du texte interlingue,
- 10 décodage du texte interlingue sur la station réceptrice,



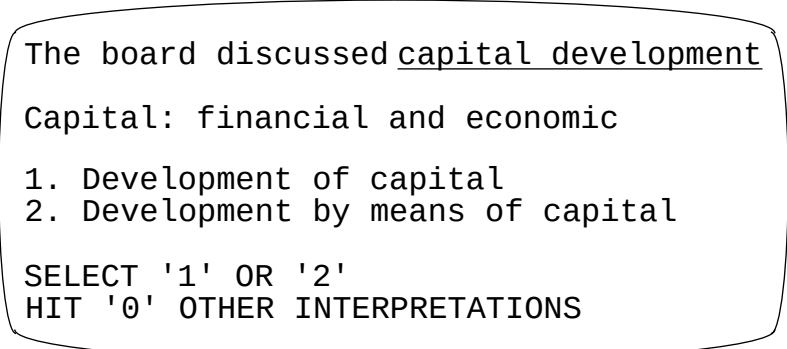
légende : processus **Résultat d'un processus** *Donnée utilisée*

Figure 1.14 : Structure du système DLT

- **11** analyse de l’interlingue : production d’une structure de dépendance,
- **12** transformation monolingue d’arbres (Interlingue) : même chose que la phase 2, mais pour l’interlingue,
- **13** transformation bilingue d’arbres : production d’un ensemble d’arbres de dépendance en français correspondant à la structure interlingue,
- **14** choix des mots français en utilisant des critères syntaxiques et pragmatiques, ainsi que la LKB (Linguistic Knowledge Bank). Il reste un seul arbre après ce processus,
- **15** transformation monolingue d’arbres (cible) : ajustement des dépendances et insertion des gouverneurs et des règles d’accord,
- **16** linéarisation de l’arbre cible : production du texte en français avec les contractions et élisions syntaxiques nécessaires : de le Æ du ; je ai Æ j’ai.

d. Désambiguïsation avec le premier prototype de DLT

- L’utilisation de l’assistance du rédacteur impose certaines contraintes.
- La préoccupation majeure de l’utilisateur n’est pas la traduction, mais la production d’un texte satisfaisant dans sa langue. Il doit donc seulement avoir à répondre à des questions à propos de l’interprétation du texte source. Comme il accepte des corrections orthographiques, grammaticales et stylistiques, l’utilisateur doit accepter des “corrections d’ambiguïtés”.
- L’utilisateur n’est pas un linguiste ou un traducteur qualifié. Les problèmes doivent être indiqués dans sa langue et de la manière la moins technique possible, les arbres et la terminologie linguistique doivent être proscrits.
- L’interruption du travail de frappe doit être limitée à la fin des phrases ou à la fin du texte.
- Le délai d’attente entre la fin de la rédaction d’une phrase et le moment où on demande à l’utilisateur de la désambiguïser doit être le plus court possible.
- Tout texte frappé directement contient inévitablement des fautes de frappe et autres erreurs. Il faut donc que le système soit robuste pour traiter les phrases mal formées. Cela signifie que le système doit être capable de détecter les fautes d’orthographe et de syntaxe avant d’entamer des traitements complexes.
- Lorsque le système a fini de traiter une phrase — en tenant compte du cycle de correction — le processus de désambiguïsation se déroule ainsi :
 - Un certain nombre de possibilités sont montrées sur l’écran, et l’utilisateur doit en choisir une (cf. fig. 1.15). Les premières possibilités qui apparaissent sont celles qui ont obtenu le plus fort score.
 - Sur le texte origine, les éléments auxquels la question réfère sont soulignés pour que l’utilisateur puisse se remémorer le contexte.
 - L’utilisateur doit alors choisir la possibilité qui correspond à ce qu’il a voulu dire. Il fait son choix en tapant le numéro du bon item ou en le désignant avec la souris.
 - L’item sélectionné est alors souligné afin que l’utilisateur puisse confirmer ou infirmer son choix.
 - Si le choix est confirmé, la question suivante est posée. Sinon, le cycle se répète.



The board discussed capital development

Capital: financial and economic

1. Development of capital
2. Development by means of capital

SELECT '1' OR '2'

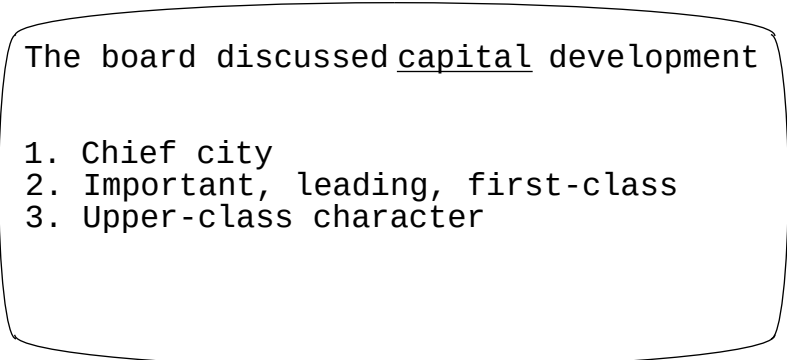
HIT '0' OTHER INTERPRETATIONS

Figure 1.15 : Proposition des sens de score supérieur au seuil

Deux options ont une importance spéciale dans chaque question : le “don’t care” et le “no good alternative shown”.

– Une réponse par “don’t care” signifie que l’utilisateur ne peut ou ne veut pas décider quel item est le meilleur. Cela peut arriver, par exemple lorsque la question concerne de proches synonymes qui, dans le contexte donné, correspondent à l’intention de l’utilisateur. Si l’utilisateur opte pour l’option “don’t care”, le système choisit l’item qui a le meilleur score ou, dans le cas de synonymes très proches, le terme générique qui les recouvre.

– Une réponse par “no good alternative shown” signifie que l’utilisateur ne considère pas que les propositions du système correspondent à ce qu’il a voulu dire. Dans ce cas, la première liste est remplacée par une seconde liste de score inférieur.



The board discussed capital development

1. Chief city
2. Important, leading, first-class
3. Upper-class character

Figure 1.16 : Proposition des sens de score inférieur au seuil

• Deux types de questions sont possibles : la paraphrase et l’isolation ou le soulignement des éléments qui vont ensemble.

- Dans la plupart des cas, le meilleur moyen pour marquer la différence entre les interprétations envisageables est de donner une paraphrase en extension de chacune de celles-ci. La paraphrase est particulièrement adaptée dans le cas des ambiguïtés lexicales — elle consiste à donner la définition d’un dictionnaire — et pour la plupart des ambiguïtés dues aux prépositions.

exemples :

Logo was conceived in the 1960's by Seymour Papert of M.I.T., primarily as a language for introducing children to computers.

Question :

by "introducing" is meant :

1. children are made familiar with computers
 2. computers are made familiar to children
-

A pen can be mounted on the undercarriage, so that when the turtle is made to wander over a sheet of paper, it leaves a record of its path.

Question :

by "made to wander" is meant :

1. constructed so as to wander
 2. forced, compelled, caused to wander
-

- Certaines ambiguïtés structurales peuvent être résolues en présentant les éléments qui dépendent les uns des autres, par exemple dans le cas des conjonctions dont la portée n'est pas claire. Cela peut être fait en présentant les différentes structures ou en indiquant dans le texte lui-même les dépendances. Dans la plupart des cas, les deux approches sont combinées, montrant la structure comme une alternative et en soulignant le texte original.

exemple :

Today such "floor turtles" are less common than "screen turtles", which move and draw on the surface of a cathode ray tube.

Question :

"on the surface" belongs to:

1. draw (alternative)
... which move and draw on the surface ... (souligné dans le texte)
 2. move and draw (alternative)
... which move and draw on the surface ... (souligné dans le texte)
-

e. Premières expérimentations

Une série d'expériences avec des dialogues de désambiguïsation a été menée en collaboration avec le laboratoire de psychologie de l'université d'Utrecht [Papagaaij 1986]. Ces expériences avaient pour but de tester la réponse de l'utilisateur à divers types de questions et de paraphrasage. Elles devaient aider à trouver comment obtenir des résultats optimaux.

Les premiers résultats indiquent l'importance du type de questions à associer à chaque type d'ambiguïté, la nécessité de s'assurer que l'utilisateur a une idée claire de

l'objectif du dialogue, l'importance d'obtenir une rédaction des items de questions les plus "contrastés" possibles pour être certain que l'utilisateur comprend le raisonnement qui conduit à la production des items.

Les expériences les plus récentes ont montré que les meilleurs résultats sont obtenus avec un paraphrasage maximal. Ce type de questions présente l'ambiguïté dans la forme que l'utilisateur comprend le mieux : en langue naturelle. Cela veut dire que le module de désambiguïsation doit être un module d'interface en langue naturelle avancé, avec de puissantes capacités de paraphrasage et un large répertoire de questions. Il est aussi très important de vraiment intégrer le module de désambiguïsation dans le système : préparer le système pour le dialogue et les demandes qui vont être faites, mais aussi envisager la possibilité de rétroaction depuis le dialogue sous la forme d'une certaine capacité d'apprentissage du système.

Papagaaij fait justement remarquer : «Considérant la complexité manifeste d'un processus de désambiguïsation fonctionnant de manière satisfaisante, le sujet a reçu peu d'attention dans la littérature. Bien que quelques autres projets mentionnent le dialogue comme une forme possible de l'assistance, seules quelques petites observations ont été réalisées à ce propos, des expériences à grande échelle seront nécessaires avant que l'on ne trouve une bonne solution au problème».

f. Expérience d'un scénario pour un dialogue de désambiguïsation

Le scénario suivant a été utilisé pour les expériences de simulation conduites au laboratoire de psychologie de l'Université d'Utrecht [Papagaaij 1986]. Ce que nous ne pouvons montrer, c'est le soulignage dynamique qui apparaît sur le texte au fur et à mesure que l'utilisateur répond aux questions.

Pour l'expérience, les sujets étaient placés devant un terminal sur lequel un texte était affiché phrase par phrase. Chaque phrase était suivie d'un certain nombre de questions. Le temps de réponse ainsi que les choix effectués étaient enregistrés pour être dépouillés ensuite. Les utilisateurs ne recevaient aucune information en retour sur la correction de leurs réponses — car aucun retour d'information n'était prévu lors d'une session normale de travail.

Dans le premier modèle, les options "don't care" et "no correct alternative shown" ne figuraient pas. Les réactions des divers sujets montrèrent qu'elles ne devaient pas être omises. En effet, il leur est apparu frustrant d'avoir à choisir un item lorsque manifestement celui qu'ils attendaient n'était pas proposé.

Expérience : (nous retraçons ici l'expérience complète)

TITRE

COMPUTER REACTION; Turning turtle gives one a view of geometry from inside out.

[Brian Hayse in Scientific American Feb. 1984]

SENTENCE 1:

The new way of thinking about geometry has come to be known as "turtle geometry".

SENTENCE 2:

It is closely connected with the programming language LOGO, which in turn has its roots in the Massachusetts Institute of Technology.

QUESTION I:

“its” belongs to:

1. The new way of thinking about geometry
 2. “turtle geometry”
 3. the programming language Logo
-

QUESTION II:

by “programming” is meant:

1. Logo is a language that does programming
 2. Logo is a language of the type programming
-

SENTENCE 3:

Logo was conceived in the 1960’s by Seymour Papert of MIT, primarily as a language for introducing children to computers.

QUESTION I:

by “introducing” is meant:

1. children are made familiar with computers
 2. computers are made familiar with children
-

SENTENCE 4:

Many others have since contributed to its development and to its applications both in education and other fields.

QUESTION I:

“its applications” belongs to:

1. development
 2. Logo
-

QUESTION II:

“in education and other fields” belongs to:

1. applications
 2. development and to its applications
-

SENTENCE 5:

Among them are Harold Abelson and Andrea A. diSessa of MIT, who have set forth the ideas underlying turtle geometry in a remarkably expository work: Turtle Geometry: The Computer as a Medium for Exploring Mathematics.

QUESTION I:

“them” belongs to:

1. fields
 2. applications
 3. Many others
-

QUESTION II:

“of MIT” belongs to:

1. Andrea A. diSessa
 2. Harold Abelson and Andrea A. diSessa
-

QUESTION III:

“who” belongs to:

1. MIT
 2. Harold Abelson and Andrea A. diSessa
-

QUESTION IV:

“in a remarkably expository work” belongs to:

1. turtle geometry
 2. underlying
 3. set forth
-

QUESTION V:

by “work” in “a remarkably expository work” is meant:

1. activity, doing or making something
 2. that which is produced by someone
 3. product of mental activity
-

SENTENCE 6:

The original turtle was a mechanical device: a small wheeled vehicle whose movements could be controlled by instructions typed on a computer keyboard.

QUESTION I:

by “original” is meant:

1. first
 2. not copied or imitated
 3. able to produce new ideas
-

QUESTION II:

“small” belongs to:

1. wheeled vehicle
 2. wheeled
-

SENTENCE 7:

The first such creature was built by the British neurophysiologist W. Grey Walter in the late 1940's.

SENTENCE 8:

It had a dome-shaped cover somewhat like a turtle shell.

QUESTION I:

by “like a turtle shell” is meant

1. a turtle shell has a dome-shaped cover
 2. a dome-shaped cover resembles a turtle's shell
-

SENTENCE 9:

A mechanical turtle can move forward or backward and can change direction by pivoting in place.

QUESTION I:

“by pivoting” belongs to:

1. change direction
 2. forward or backward and can change direction
-

QUESTION II:

“in place” belongs to:

1. pivoting
 2. change direction
-

SENTENCE 10:

A pen can be mounted on the undercarriage, so that when a turtle is made to wander over a sheet of paper, it leaves a record of its path.

QUESTION I:

by “made to wander” is meant:

1. constructed so as to wander
 2. forced, compelled, caused to wander
-

QUESTION II:

by “over” is meant:

1. across
 2. from one side to the other side
 3. above
-

QUESTION III:

“it” belongs to:

1. a sheet of paper
 2. the turtle
 3. A pen
-

QUESTION IV:

by “a record” is meant:

1. written account
 2. facts known about the past
 3. something that provides evidence or information
-

QUESTION V:

“its” belongs to:

1. a sheet of paper
 2. the turtle
 3. A pen
 4. a record
-

SENTENCE 11:

Today such “floor turtles” are less common than “screen turtles”, which move and draw on the surface of a cathode-ray tube.

QUESTION I:

“on the surface” belongs to:

1. draw
 2. move and draw
-

SENTENCE 12:

The turtle itself is represented on the screen by a simple triangular form, which moves in response to commands or programs entered at the keyboard.

QUESTION I:

1. simple triangular form
 2. simple, triangular form
-

QUESTION II:

“entered at the keyboard” belongs to:

1. programs
 2. commands and programs
-

g. Désambiguïisation dans une nouvelle conception de DLT

En 1989, Sadler [Sadler 1989] proposa un nouveau prototype et une nouvelle façon d’envisager le dialogue. Au lieu de présenter toutes les interprétations dans un même dialogue, le système présente l’interprétation qui a obtenu le meilleur score. Si celle-ci est rejetée par l’utilisateur, le système propose alors la suivante, jusqu’à ce que l’utilisateur accepte la proposition.

Avec une telle organisation, on évite de proposer tous les sens d’un mot qui a déjà été rencontré dans le texte, et on présente simplement — pour validation — le sens précédemment sélectionné.

Dans le prototype précédent, la désambiguïisation structurale précédait la désambiguïisation lexicale. L’ordre logique reste le même, sauf que l’ambiguïté de référence — non considérée dans le prototype précédent — est insérée entre les deux phases.

Voici ce qu’envisage Sadler pour le nouveau prototype.

- Ambiguïté structurale

Dans la phrase “The above-mentioned fields are only some of the many areas of society on which advanced technology will have a considerable effect.”, on détecte deux ambiguïtés : l’attachement de “only” et celui de la proposition relative. Les choix du

système peuvent être explicités d’abord en couplant l’adverbe et son gouverneur (avec de la couleur par exemple), ensuite en développant le pronom relatif.

Interpretation

The above-mentioned fields are only some of the many areas of society on which areas advanced technology will have a considerable effect.

- Dans le cas d’attachement de pré-modifieur, la répétition des liens de dépendance est efficace :

pour “cold water level”, le système peut suggérer

Interpretation

cold water, water level

pour “Defense Advanced Research Projects Agency”, l’interprétation la plus probable peut être explicitée comme suit :

Interpretation

Defense Agency, Advanced Projects, Research Projects

- Pour la portée des modifieurs, on peut changer l’ordre des syntagmes.

pour “high-speed modulators and optical couplers”, on obtient :

Interpretation

optical couplers, high-speed modulators

- Ambiguïté référencielle :

pour “The translator-reviser may if he wishes call the translation on his display screen. It will be displayed in a special format.”, on obtient :

Interpretation

It (the translation) will be displayed in a special format.

Des techniques similaires peuvent être utilisées pour le traitement des ellipses.

- Ambiguïtés fonctionnelles :

Pour “radiation protection legislation”, on doit pouvoir produire :

Interpretations

legislation **on** radiation protection, protection **against** radiation

Si de telles relations ne sont pas explicitées, avec une base de bitextes, on peut produire :

Interpretation as in
measuring techniques in radiation protection

Si l'utilisateur n'est pas certain que l'exemple corresponde au bon sens, il doit être possible de montrer d'autres exemples :

measuring techniques in	radiation protection
on the professional qualifications of	radiation protection personnel
an order introducing the Sievert as a	radiation protection measuring unit was enacted
an intercomparison program for	radiation protection dosimeters, the aim of which
technical capability in the field of	radiation protection measures relating to all peaceful

- Ambiguïté lexicale

Ce dernier type est le plus embarrassant. La technique utilisée dans le premier prototype — utilisation de paraphrases explicites stockées dans le dictionnaire bilingue — s'est avérée peu satisfaisante et extrêmement coûteuse en main-d'œuvre. La solution que l'on propose maintenant est proche de la solution précédente : l'interprétation préférée peut être illustrée en utilisant comme exemple le contexte de l'unité à traduire dans la base de connaissance bilingue (base de bitextes ou "BKB").

Pour la phrase : "The above-mentioned fields are only some of the many areas of society on which advanced technology will have a considerable effect.", les problèmes d'ambiguïté lexicale concernent "fields", "areas", "society", "advanced technology", et "have a considerable effect."

Un dialogue est proposé, dans lequel un exemple est associé à chaque problème.

Particular efforts have been made in the following **fields** - Smoking requirements for graphics within the many **areas** of CIM but excluding CAD effects are most strongly felt by the least favoured sections of the **society** centres of excellence for **advanced technology**, bringing together car-makers two factors naturally **have a considerable effect** on the overall production set-up.

Pour chaque locution ambiguë, on peut bien entendu obtenir un dialogue développé comme par exemple pour "areas."

The draft Plan identified certain	areas	for co-operation in research and
information was essential in such	areas	as the relationship of aging of
from country to country, they were	areas	of common concern to both
of technological changes in the	areas	concerned, the product sector
ventures with other companies in some	areas	of activity.
memorisation recorded in all	areas	of computer technology.
the progress made in the various	areas	of integration in which the common

Une certaine expérimentation est nécessaire pour vérifier effectivement l'efficacité de cette méthode.

h. Conclusion

Il faut reconnaître à ce système son antériorité comme système pour l'auteur. Cela explique certaines erreurs d'interface et d'expérimentation.

L'utilisateur n'est pas maître du système. Dans le mode "désambiguïsation après chaque phrase", l'utilisateur peut effectuer du travail inutile s'il décide, plus tard, de modifier une phrase précédemment désambiguïsée. On peut dire aussi que ce mode de travail perturbe nécessairement l'utilisateur, constamment interrompu dans sa rédaction et dans sa réflexion.

D'autre part, l'expérimentation conduite pour étudier la méthode de désambiguïsation ne permet pas de la valider. En effet, les tests ne s'adressent pas aux rédacteurs du texte. Il faut que le sujet interprète des phrases qu'il n'a pas rédigées avant de répondre aux questions. La situation d'expérimentation est donc fort éloignée de la situation réelle, où le dialogue s'adresse au rédacteur, supposé être motivé pour désambiguïser un document dont il veut transmettre le sens.

Il y a cependant des idées à reprendre. Par exemple, la gestion des répétitions lexicales est intéressante. La répétition du même dialogue peut être jugée agaçante. L'utilisateur aura donc le moyen de spécifier une interprétation globale "par défaut". Dans un texte financier, si l'utilisateur fixe un sens par défaut au mot "bank" lors de la première occurrence de ce terme, lors des apparitions suivantes le système fera un simple rappel "bank = financial institution". L'utilisateur n'aura qu'à valider ce message si le sens n'a pas changé. S'il ne le valide pas, le dialogue normal s'affiche.

Les propositions de Sadler [Sadler 1989] méritent d'être testées, cette fois en situation réelle. On peut imaginer que seule l'interface fonctionne et qu'un magicien d'Oz [Dahlbäck & Jönsson 1986 ; Dahlbäck, *et al.* 1993] simule le système de traduction et pose les questions. De telles expériences doivent être prochainement menées à ATR-ITL où K.H. Loken-Kim et ses collaborateurs ont réalisé une première version d'un simulateur, AMUSE (ATR Multi-Media Simulation Environment).

1.3.2. N-Tran (1984-1987)

a. Historique

N-Tran est un projet, développé au Centre for Computational Linguistics de l'université de Manchester, et soutenu par International Computers Limited et le Science and Engineering Research Council dans le cadre du programme Alvey de recherche sur les systèmes intelligents fondés sur la connaissance (Intelligent Knowledge-Based Systems). Le projet, intitulé "lire et écrire en japonais sans connaître cette langue" fut financé pour une première période de deux ans d'octobre 1984 à octobre 1986 [Johnson & Whitelock 1985 ; Tsujii 1986], puis reconduit pour un an jusqu'en octobre 1987. Avec N-Tran [Johnson & Whitelock 1985 ; Tsujii 1986 ; Wood 1989 ; Wood & Chandler 1988], on souhaitait permettre à un anglais monolingue, rédacteur de documents techniques, de produire simultanément des versions en anglais et en japonais. Un prototype a finalement été présenté en décembre 1987 [Chandler, *et al.* 1987].

b. Fiche d'identité

Nom du projet	N-Tran
Langue(s) source(s)	anglais
Langue(s) cible(s)	japonais
Type de système	outil de TAO interactif (en phase d'analyse et de transfert)
Paradigme	seconde génération : analyse – transfert – génération
Architecture informatique	au cours du développement, des versions ont fonctionné sur DEC MicroVax II, ICL PERQ et finalement sur station de travail Sun 3/50
Corpus	documentation technique

c. Structure du système

Le système comporte les modules suivants : un module d'analyse de l'anglais, un module de transfert, un module de génération du japonais et des dictionnaires monolingues (anglais et japonais) et bilingue.

L'analyse de l'anglais est basée sur les grammaires lexicales fonctionnelles de Kaplan et Bresnan (1982) et sur les grammaires syntagmatiques généralisées de Gazdar et al. (1985). La génération du japonais est basée sur la grammaire catégorielle de Streedman (1985).

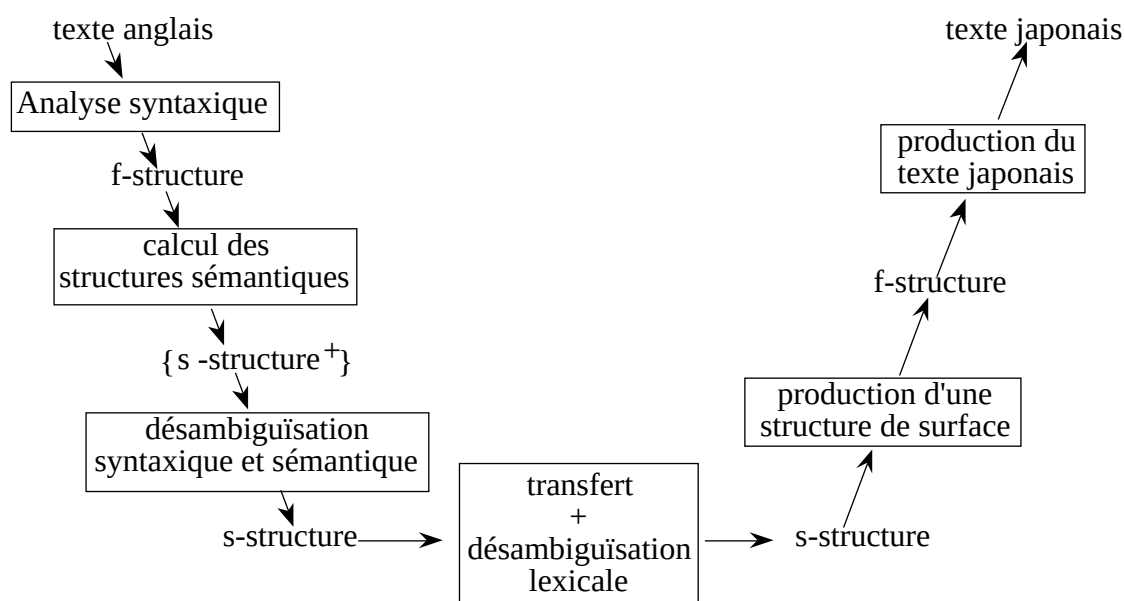


Figure 1.17 : Organisation des traitements dans N-Tran

L'analyseur construit d'abord une structure de surface de dépendances syntaxiques, ou f-structure, qui est ensuite transformée en une structure profonde ou sémantique : s-structure (semantic f-structure). Si plusieurs s-structures sont produites, on fait intervenir l'utilisateur pour qu'il choisisse la structure qui correspond au sens de la phrase. Le mécanisme mis en œuvre permet de représenter les ambiguïtés purement structurelles, comme l'attachement des groupes prépositionnels, et aussi les ambiguïtés de rection, comme dans "écrire sur le pont du navire" où "pont" peut être la localisation ou l'objet de "écrire".

Les diverses interprétations possibles sont présentées à l'utilisateur qui choisit celle qui convient. Une seule s-structure demeure, qui est transmise à la phase de transfert. Dans un premier temps, le système cherche les équivalents japonais. L'utilisateur peut être amené à intervenir si le système ne peut effectuer la sélection lexicale. Aux informations sémantiques de la s-structure anglaise sont aussi associées les informations sémantiques japonaises correspondantes. Durant la deuxième phase du transfert, le système calcule les valeurs sémantiques qui n'ont pas été calculées pendant la première phase du transfert. On obtient finalement une s-structure japonaise qui est transformée en f-structure en utilisant les dictionnaires du japonais. La chaîne de surface en japonais est obtenue en appliquant des règles de précedence linéaire (ordre des mots).

d. Désambiguïisation avec N-Tran

Les auteurs du système préfèrent dire qu'il permet la prédiction interactive au cours du processus de traduction plutôt que de parler de traduction interactive, du fait qu'aucune information contrastive ou bilingue n'est présentée à l'utilisateur. Le prototype N-Tran est conçu pour poser trois types de questions : enrichissement de dictionnaires en ligne, désambiguïisation syntaxique, et sélection lexicale japonaise pendant la phase de transfert.

Si l'on obtient plusieurs structures sémantiques ou s-structures, les choix sont présentés comme un ensemble d'énoncés qui distinguent et caractérisent les interprétations sémantiques produites. Les auteurs reconnaissent que la présentation devrait être améliorée dans un système plus développé. Ils font remarquer que, avec le même mécanisme sous-jacent, on pourrait obtenir des présentations très différentes allant du paraphrasage à des présentations graphiques.

Enfin, des ambiguïtés peuvent apparaître dans la sélection d'équivalents japonais pour un mot ou une expression anglaise. Le système présente les choix possibles à l'utilisateur, qui doit avoir une certaine connaissance du japonais pour faire le bon choix. Cela contredit le fait que l'utilisateur est supposé monolingue. En fait, les dictionnaires japonais de N-Tran devaient inclure des définitions en anglais qui auraient été présentées à l'utilisateur, mais cela n'a pas été complètement réalisé.

exemples :

Voici quelques exemples de dialogues de désambiguïisation, que l'on obtient avec N-Tran. Pour répondre à une question, l'utilisateur tape un numéro pour valider une option ou "f un numéro" pour le nier.

type the number of any true statement or fnumber of any false statement	
1. by is location of replace	true for parses [1-2]
2. metafile is agent of replace	true for parses [1-1]
Files are replaced by metafiles.	
parsing: 11sec parses: 2 deep: 2	

type the number of any true statement or fnumber of any false statement	
1. to is arg2 of write	true for parses [1-2]
2. to is location of write	true for parses [1-1]
please choose:	
The metafiles are written to sequential files.	
parsing: 11sec parses: 2 deep: 2	

type the number of any true statement or fnumber of any false statement	
1. between is location of transport	true for parses [5-1]
2. between is location of use	true for parses [4-1]
3. between is location of information	true for parses [1-1]
please choose:	
They can be used to transport information between systems.	
parsing: 64sec parses: 5 deep: 2	

type the number of any true statement or fnumber of any false statement	
1. workstation is object of active	true for parses [2-1]
2. and is object of active	true for parses [1-1]
3. active is modifier of terminal	true for parses [1-1]
please choose:	
Output is sent to active workstations and terminals.	
parsing: 61sec parses:4 deep: 1	

On observe que le dialogue n'est pas un paraphrasage. On cherche à établir les relations en les nommant.

Voici l'écran de désambiguïsation qui est présenté à l'utilisateur :

```

*** CCL Grammar Development System *** Version 0.65 level 31a ***
-----
type the number of any true statement
or fnumber of any false statement

1      on is location of position      true for parses [2-1]
2      on is location of correspond    true for parses [1-1]
please choose:

-----
The cursor correspond to the puck position on the tablet.

maptrace(1,1, [map(correspond,arg0,pres),
               map(cursor,arg0,correspond),
               map(position,arg1,correspond),
               map(on,loc,correspond) |A]).

maptrace(2,1, [map(correspond,arg0,pres),
               map(cursor,arg0,correspond),
               map(position,arg1,correspond),
               map(on,loc,position) |A]).

The cursor corresponds to the puck position on the tablet.

ka-soru ga taburetto de no pakku iti ni soutou souru

cursor NOM tablet ATTR ADN puk position DAT correspond press

parsing: 36sec parses: 4 deep:1 transfert: 49sec xltns: 2
translation 1

ka-soru ga taburetto no ue no pakku iti ni soutou suru

cursor NOM tablet ADN above_place ADN puck position DAT Correspond pres

translation 2

```

Figure 1.18 : Écran de désambiguïsation avec N-Tran

e. Conclusion

Le dialogue n'a pas été travaillé suffisamment pour que l'on puisse émettre un jugement définitif, mais il est clair qu'il ne peut pas s'adresser à quelqu'un de non entraîné. Les questions qui sont posées reflètent le modèle linguistique sous-jacent que l'utilisateur doit donc connaître. En transfert, le dialogue pour le choix des équivalents lexicaux n'est pas monolingue.

À la suite de ces travaux, un bon nombre de chercheurs ont eu l'idée de concevoir des systèmes où des textes assez stéréotypés dans des domaines fixes peuvent être générés à l'aide d'une interaction avec l'utilisateur.

C'est le cas du nouveau système de l'UMIST [Jones & Tsujii 1990 ; Somers & Jones 1992 ; Somers, *et al.* 1990] qui génère de la correspondance commerciale en français, en allemand et en espagnol. C'est aussi le cas du système de Saito et Tomita [Saito & Tomita

1986], qui produit des textes en japonais, et du système de Zaki et Noor [Zaki & Noor 1991], pour composer des lettres en anglais et en malais. Enfin, il y a des produits commerciaux de ce type, comme Ambassador sur le Macintosh.

Dans le système de l'UMIST, des textes originaux (nouveaux) peuvent être assemblés par composition de textes déjà en mémoire. Le grand avantage de cette variation de la TA, c'est que les textes sont de très bonnes traductions, étant donné qu'elles sont copiées à partir d'un corpus de vraies traductions, plutôt que produites par consultation de dictionnaires et application de règles générales.

1.3.3. JETS (1985-)

a. Historique

Les travaux sur le système JETS [Tsutsumi, *et al.* 1993] ont commencé en 1985 dans les laboratoires d'IBM à Tokyo. Le système JETS est testé en interne et chez des clients.

JETS est un noyau de système de traduction personnalisable sans domaine privilégié. Ses principales caractéristiques du point de vue logiciel sont :

- la facilité de personnalisation,
- une interface pour un utilisateur monolingue japonais,
- une grande facilité d'intégration avec d'autres applications.

Le transfert fondé sur l'exemple de JETS joue un rôle important dans la personnalisation des connaissances linguistiques du système codées dans les dictionnaires, les modèles et les règles de transfert pour des domaines spécifiques. Dans beaucoup de cas, la personnalisation se fait simplement en ajoutant des exemples dans la base.

Selon les concepteurs de JETS, il est nécessaire d'intégrer le système de traduction dans les tâches telles que l'édition de texte, PAO et FAO. JETS fonctionne sur une station IBM RS/6000 et l'éditeur bilingue multifenêtres peut fonctionner sur une RS/6000 sous AIX ou sur un IBM PS/55 sous OS/2. Le système est conçu selon un modèle client-serveur et est accessible sur un réseau local (LAN). JETS peut donc être intégré facilement à d'autres applications. Par exemple, il a déjà été intégré dans Emacs, Publishing/55 (un produit IBM de DTP) et Micro CADAM¹¹.

Les réalisateurs se sont concentrés sur la traduction de textes japonais simples, tels que les annotations dans les dessins techniques et les "notices de changement de conception" (design change notice). Par exemple, IBM a démontré une application prometteuse de traduction japonais-anglais d'annotation dans les dessins techniques en intégrant JETS et Micro CADAM. L'utilisateur appelle JETS depuis un écran Micro CADAM et obtient la traduction des annotations du schéma en cours. Les traductions anglaises sont insérées sous les annotations japonaises ou se superposent sur celles-ci, suivant le choix de l'utilisateur.

¹¹ Système de CAO développé par Dassault et commercialisé par IBM.

b. Fiche d'identité

Nom du projet	JETS
Langue(s) source(s)	japonais
Langue(s) cible(s)	anglais
Type de système	système de TAO interactive
Paradigme	système de traduction fondé sur une analyse de dépendance par propagation de contraintes et un transfert par l'exemple
Architecture informatique	distribuée, modèle client serveur
Corpus	JETS est un noyau de système de traduction personnalisable sans domaine privilégié

c. Structure du système

L'analyse morphologique est faite au moyen d'une grammaire régulière augmentée par un mécanisme de structure de traits et un coût associé à chaque règle. La structure de traits permet la production d'informations grammaticales et le mécanisme de coût permet d'ajuster facilement la grammaire.

L'analyse syntaxique est basée sur une grammaire de dépendances avec contraintes qui permet l'amélioration incrémentale de la grammaire et un traitement uniforme de la grammaire et des interactions de l'utilisateur, vues comme des contraintes. On évite ainsi au système de produire des arbres d'analyse intermédiaires.

Après la détermination des cas de chaque relation établie par l'analyse dépendancielle, une forme canonique est produite. Le transfert du japonais vers l'anglais est un transfert fondé sur l'exemple. Le module de génération de l'anglais est basé sur les grammaires relationnelles.

d. Désambiguïsation avec JETS

JETS offre à l'utilisateur un environnement interactif d'analyse du japonais [Maruyama, *et al.* 1990]. Le système pose des questions pour lever des ambiguïtés lors des phases d'analyses morphologique, syntaxique et sémantique lorsqu'il ne peut résoudre seul les ambiguïtés.

Au cours de l'analyse morphologique, si JETS est confronté à une segmentation (en mots) ambiguë, il montre les diverses possibilités à l'utilisateur qui peut modifier la segmentation produite par le système.

S'il y a un problème d'attachement, JETS décrit le problème et demande à l'utilisateur de le résoudre.

L'une des particularités importantes de JETS est que le système vérifie le sens des mots et des syntagmes ambigus. Par exemple, une phrase japonaise contenant la

conjonction de subordination “tameni” est analysée et une structure intermédiaire est produite. Le système essaie de déterminer si la conjonction introduit un but ou une raison. S’il ne réussit pas à en choisir une, il génère une phrase japonaise qui reflète l’une des interprétations possibles. L’utilisateur peut alors valider ou transformer la reformulation qui lui est proposée. La transformation permet d’obtenir une phrase non ambiguë par construction.

Les homographes, tels que “boruto” (bolt ou volt, verrou ou volt) ou “karaa” (color ou collar, couleur ou col), sont aussi traités. L’utilisateur doit choisir le sens de l’occurrence.

Selon ses concepteurs, cet environnement interactif permet à des utilisateurs qui ne sont pas familiers avec la langue anglaise d’obtenir des traductions fiables.

e. Exemple de session :

Nous allons maintenant suivre pas à pas une session de désambiguïsation avec JETS, et plus particulièrement la phase de l’analyse qui cherche à établir les liens entre les *modifieurs* et les *modifiés*. Les treize figures qui suivent nous ont été fournies par Hiroshi Maruyama. Les couleurs sont indiquées par du texte sur les figures.

La figure 1.19 montre le panneau dans lequel on choisit la phrase à traduire. Ici, la première phrase est sélectionnée.

Après la phase d’analyse morphologique, le système effectue la première étape de l’analyse syntaxique. Le rôle de cette étape est de trouver quel groupe (bunsetsu en japonais) modifie quel(s) autre(s) groupe(s). La figure 1.20 montre l’état initial de cette étape. Toutes les relations possibles ont été calculées par le système.

Lorsque l’utilisateur place le curseur de la souris sur un groupe, celui-ci se surligne en bleu et les groupes qu’il est susceptible de modifier sont surlignés en vert. Dans ce cas, il y en a quatre. Le nombre de candidats apparaît toujours en dessous de chaque groupe. En haut à gauche de l’écran, on trouve le produit des candidats qui correspond plus ou moins au nombre de lectures de la phrase en suspens.

L’image suivante, figure 1.21, montre un autre état de l’écran lorsque l’utilisateur déplace la souris.

Les contraintes de la grammaire sont regroupées en catégories en fonction de leur degré de confiance. Les contraintes les plus fortes, comme l’interdiction d’avoir des paires croisées modifieur-modifié ont déjà été appliquées. Les contraintes plus lâches peuvent être propagées à la demande de l’utilisateur.

Dans la figure 1.22, l’utilisateur choisit la contrainte “non-multiple-object-case” parce qu’il sait que la phrase ne comporte pas de construction parallèle particulière.

La figure 1.23 montre le résultat de l’application de cette contrainte sur la phrase. On peut remarquer que certains groupes ont obtenu un modifié unique grâce à cette contrainte. Par exemple, le groupe en blanc dont le modifié, en vert, est juste à sa droite.

Maintenant, l’utilisateur cherche une autre contrainte qui pourrait diminuer l’ambiguïté. Sur la figure 1.24, on voit qu’il vérifie les modifiés possibles par le quatrième groupe de la phrase.

Figure 1.25, l'utilisateur choisit une nouvelle contrainte "topic-phrase-modifiers-main-verb"

Figure 1.26, la nouvelle contrainte a été appliquée.

Maintenant, l'utilisateur essaye de spécifier directement le modifié d'un groupe. Ici, il s'intéresse au groupe surligné en bleu qui possède deux candidats à la modification (figure 1.27).

Figure 1.28, l'utilisateur choisit le premier (celui qui est resté vert). Pour ce faire, il a d'abord effectué un clic droit sur le modificateur (en bleu), puis un clic gauche sur le modifié (resté en vert, l'autre est devenu rouge). Une nouvelle phase de propagation des contraintes est effectuée après cette sélection.

La figure 1.29 montre le résultat de cette phase de propagation de contraintes. On peut noter que le modifié du groupe situé entre les deux groupes précédemment sélectionnés a été déterminé automatiquement.

Figure 1.30, l'utilisateur choisit un autre groupe et choisit le modifié.

Après cela, figure 1.31, toutes les ambiguïtés ont été levées et l'arbre d'analyse est prêt à être transmis à la phase suivante du processus de traduction.

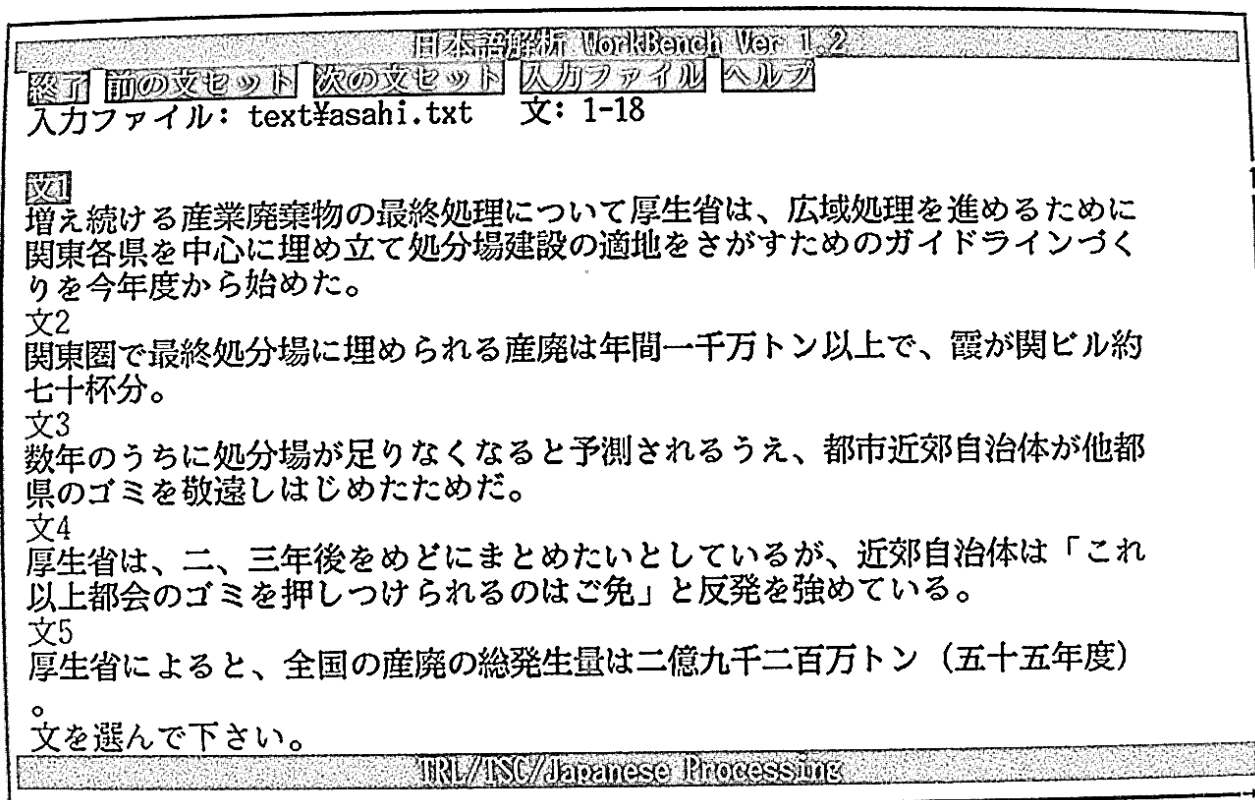


Figure 1.19 : Écran de sélection de la phrase à traduire avec JETS

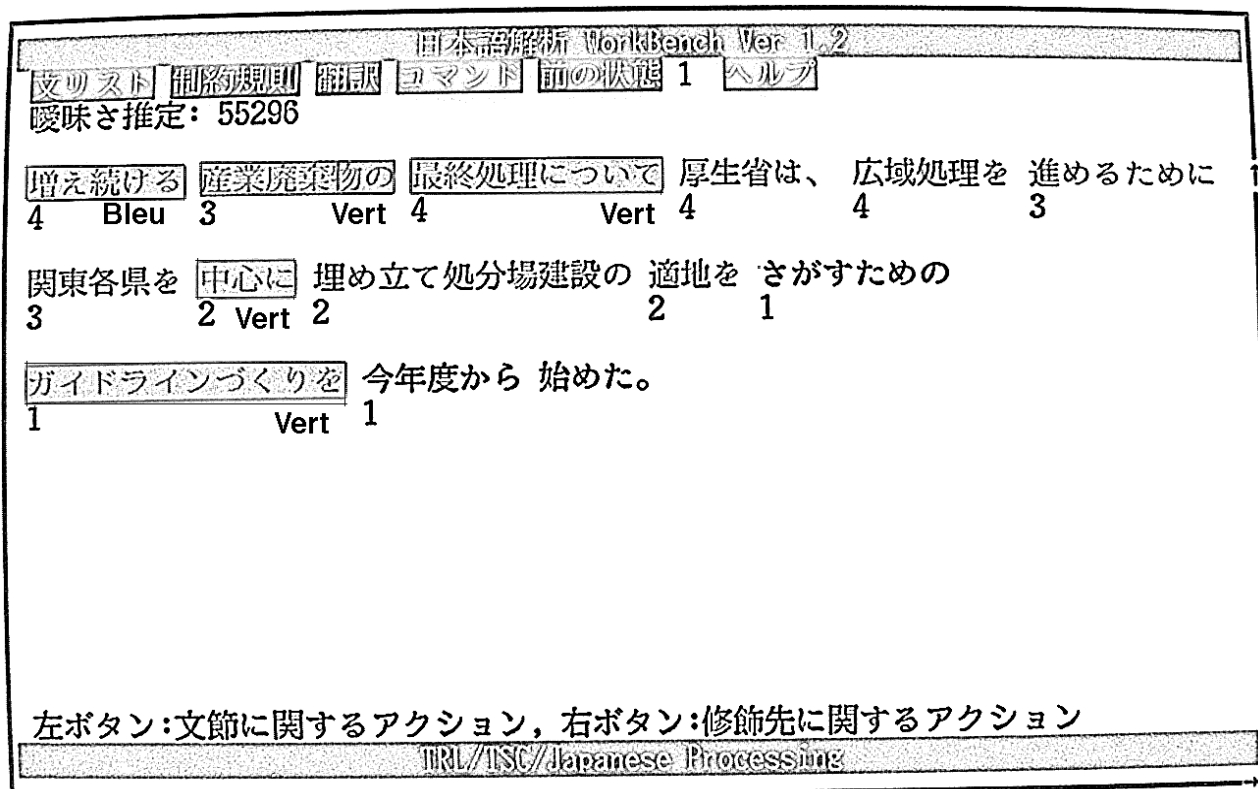


Figure 1.20 : Début de la désambiguïsation interactive avec JETS
en bleu, un groupe sélectionné ; en vert, les modifiés possibles

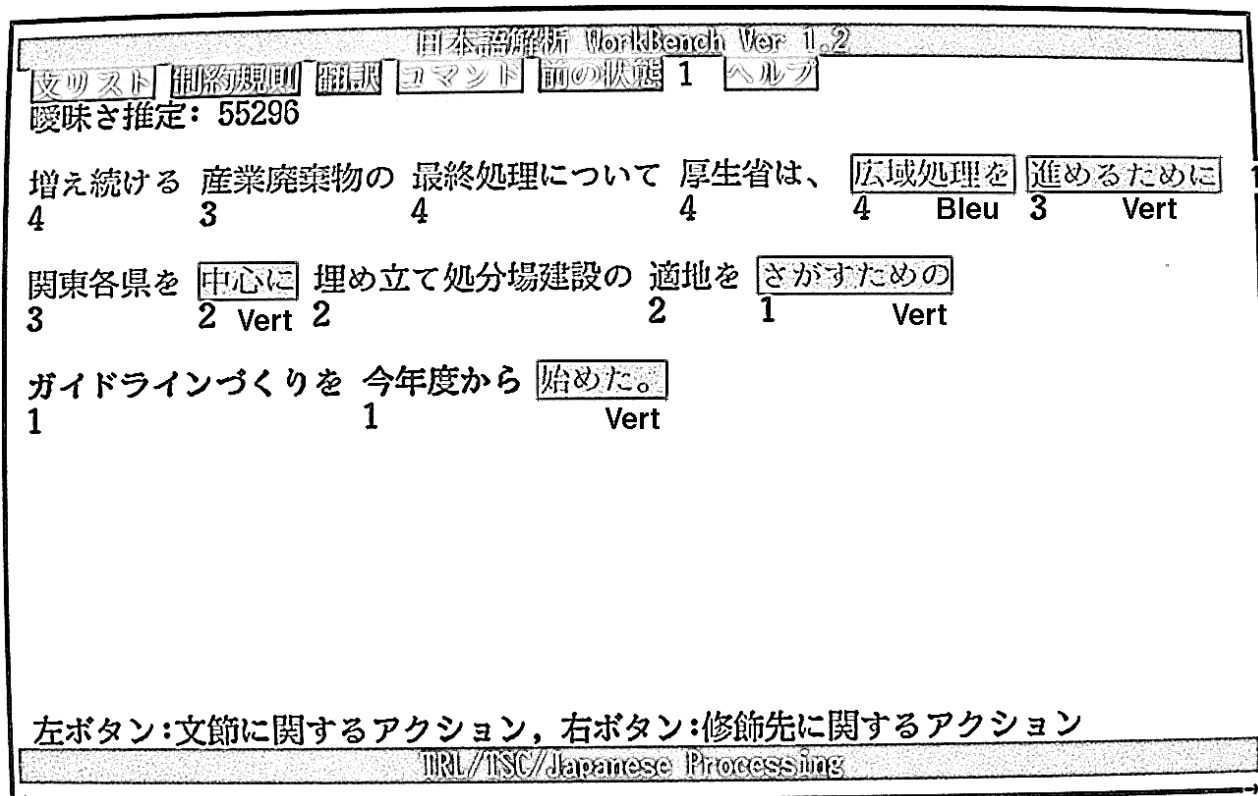


Figure 1.21 : Sélection d'un nouveau groupe

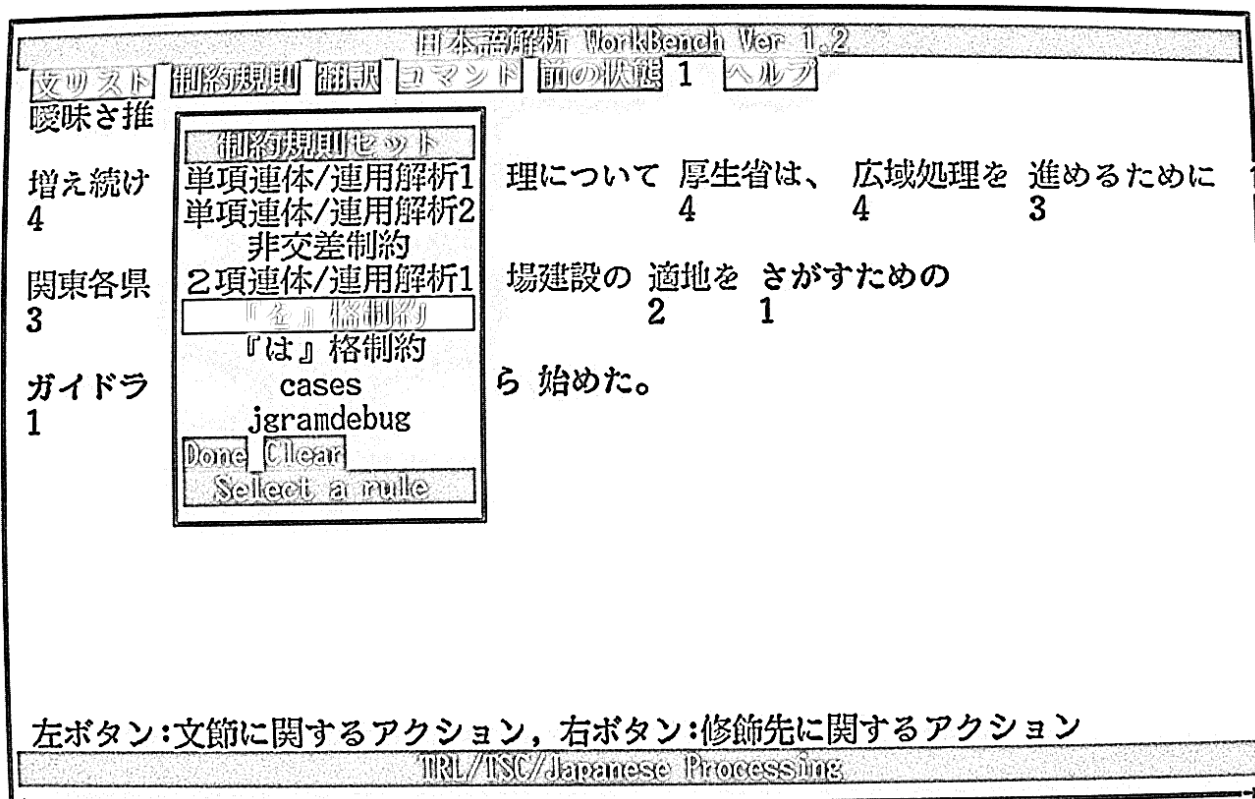


Figure 1.22 : Sélection d'une première contrainte

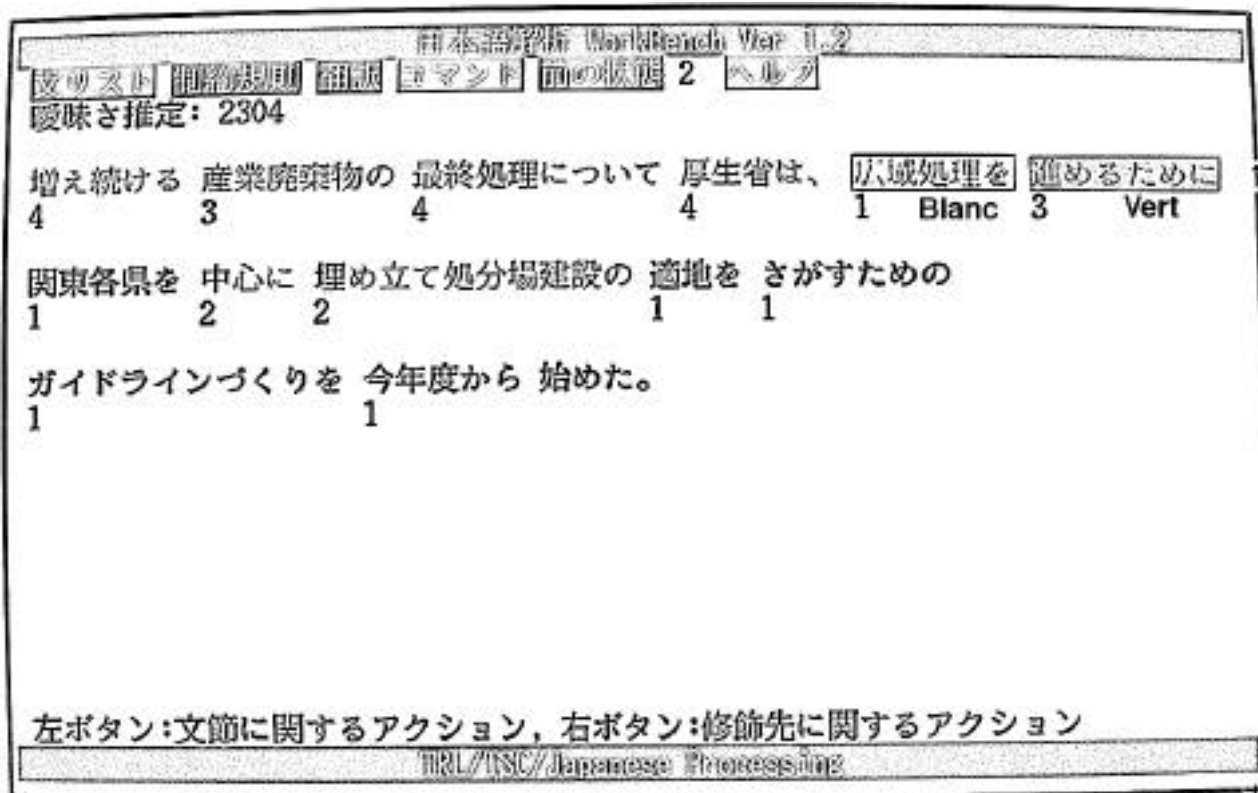


Figure 1.23 : Résultat de l'application de la première contrainte

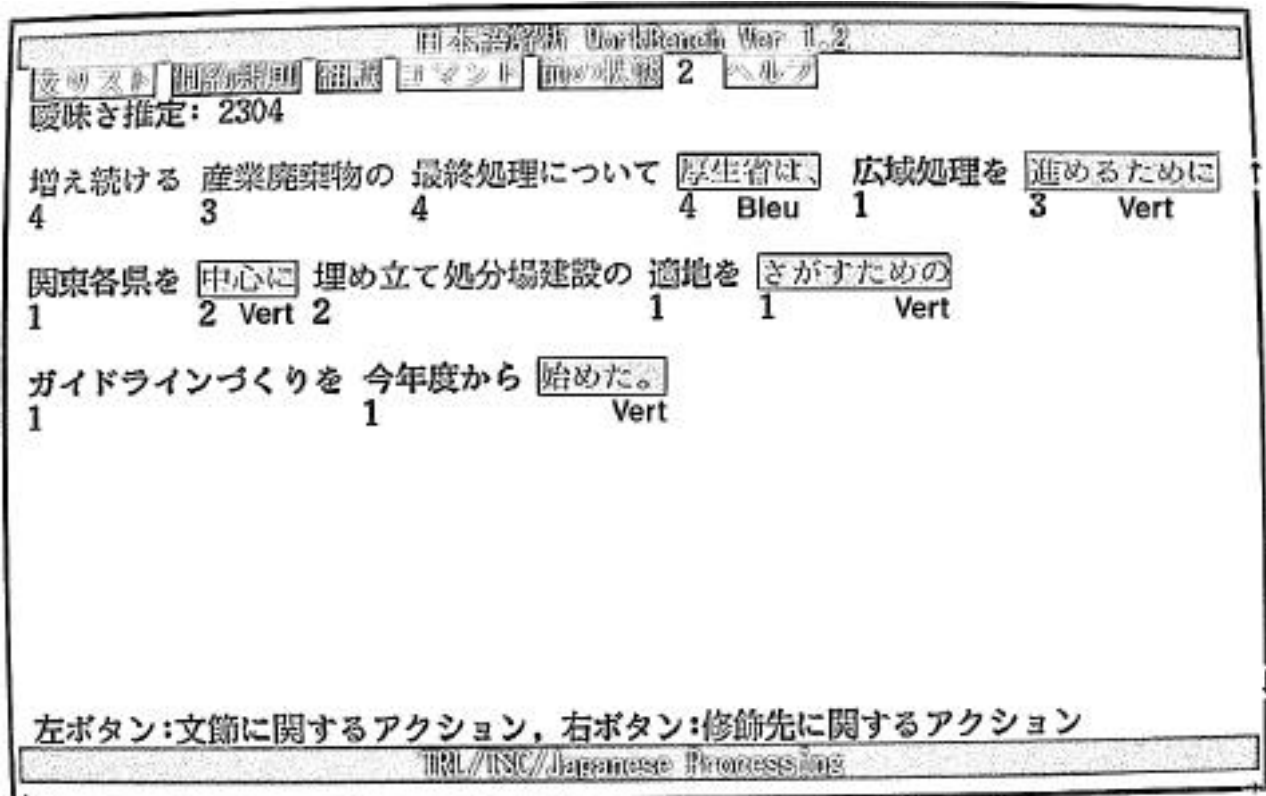


Figure 1.24 : Sélection d'un nouveau groupe pour choisir une nouvelle contrainte

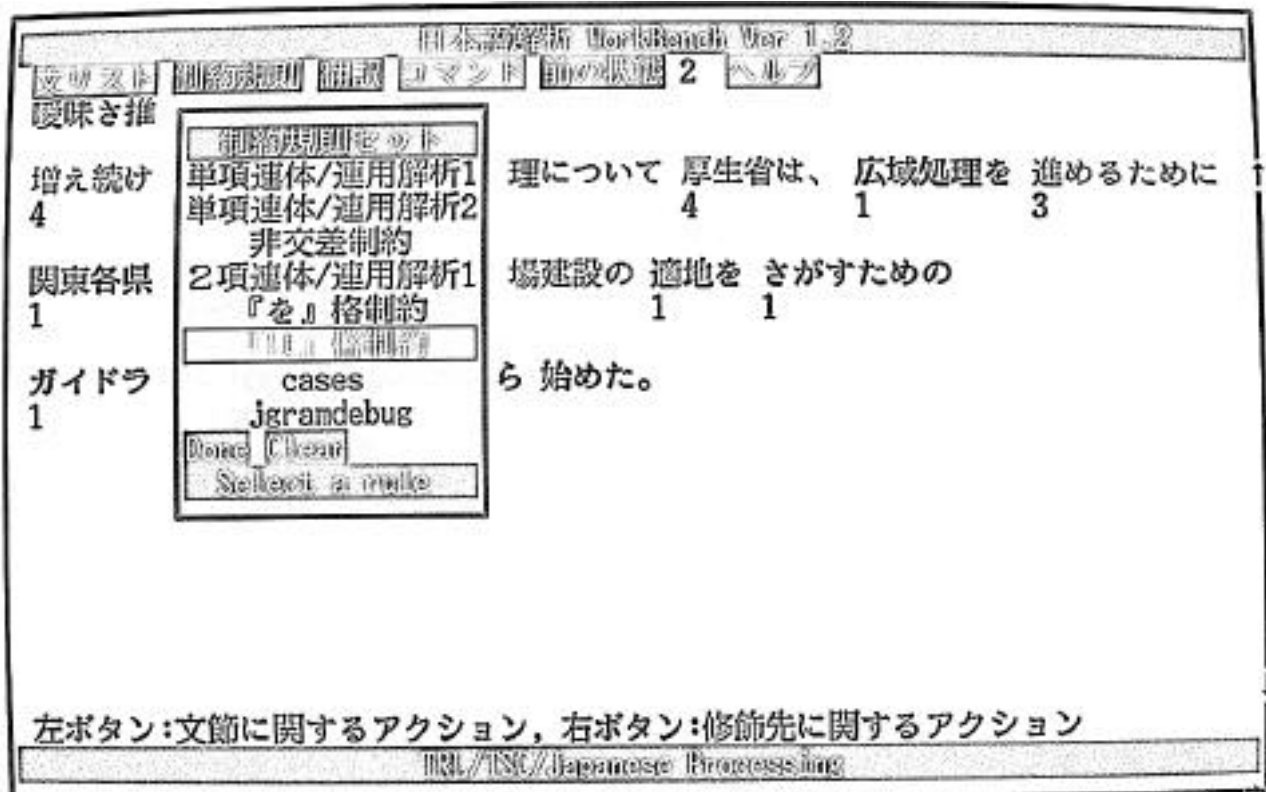


Figure 1.25 : Sélection d'une seconde contrainte

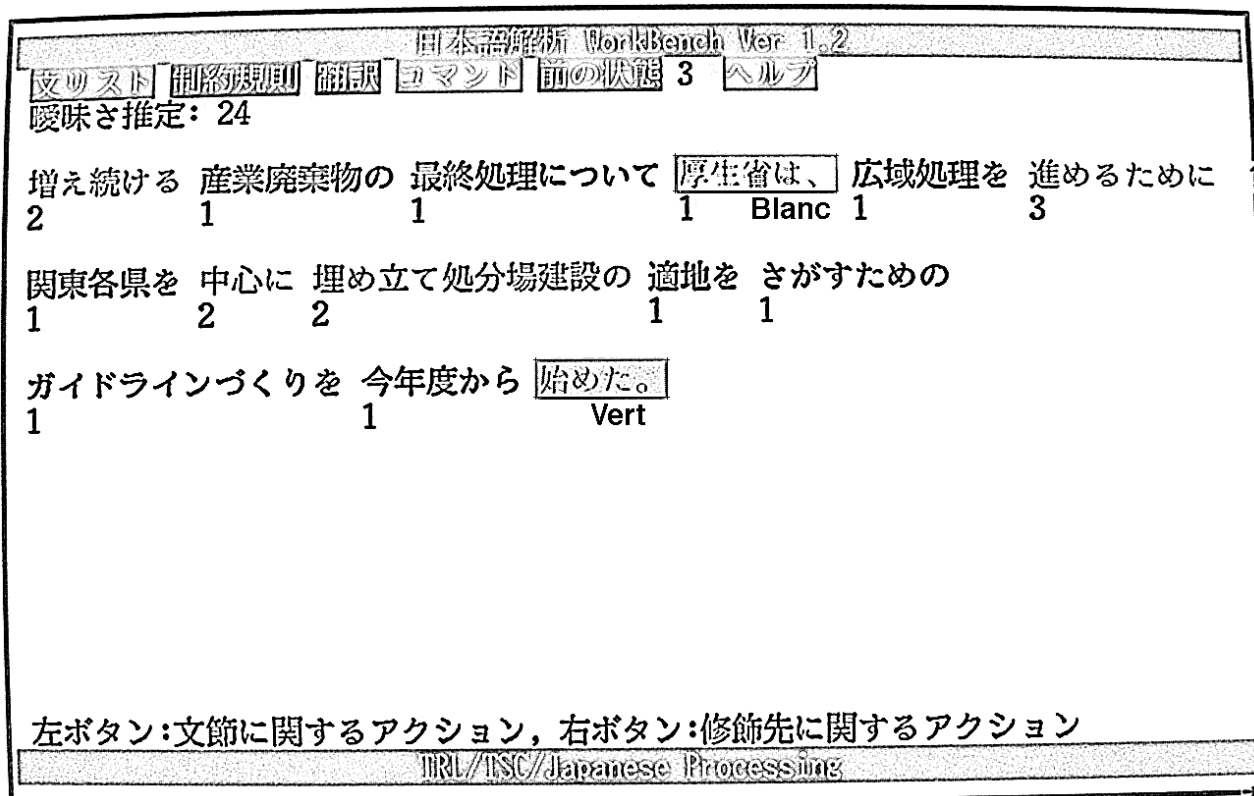


Figure 1.26 : Résultat de l'application de la seconde contrainte

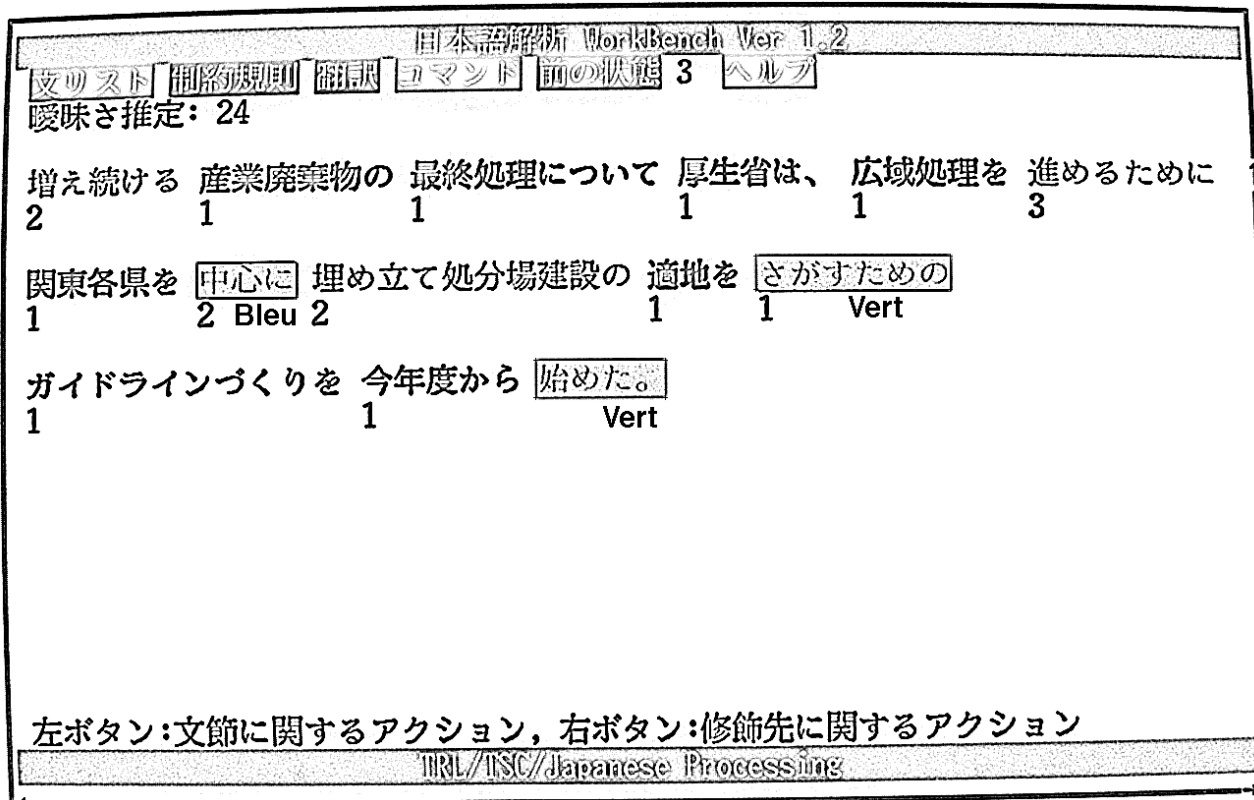


Figure 1.27 : Sélection d'un nouveau groupe pour faire un traitement manuel

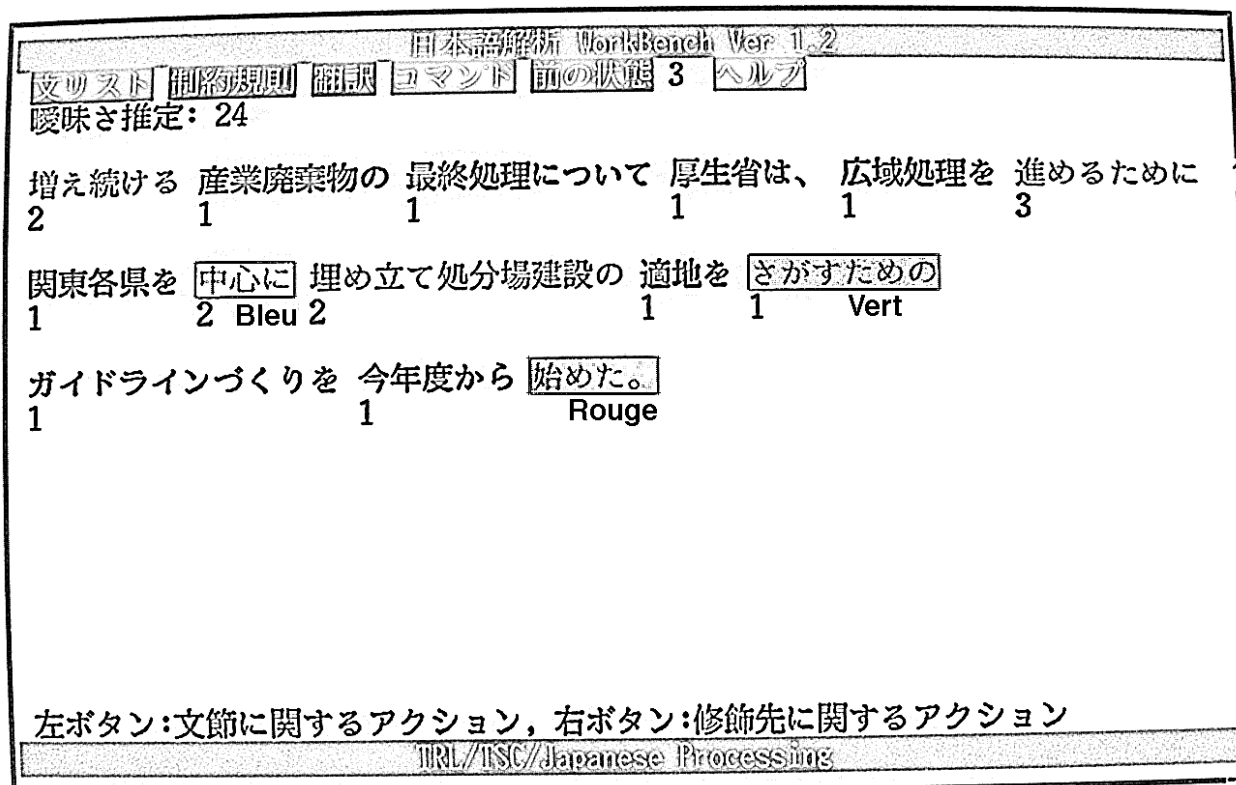


Figure 1.28 : Sélection manuelle d'un modifié (en vert)

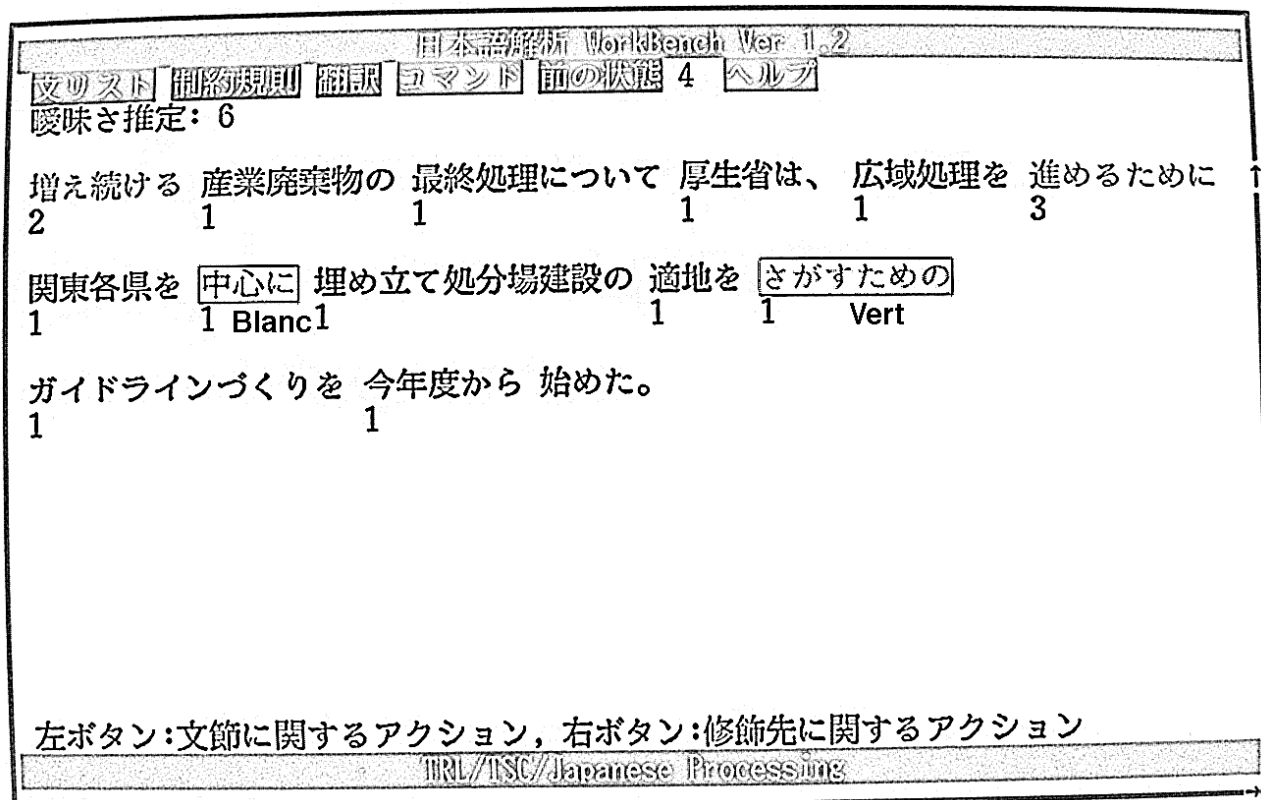


Figure 1.29 : Propagation des nouvelles contraintes imposées par la désambiguïsation précédente

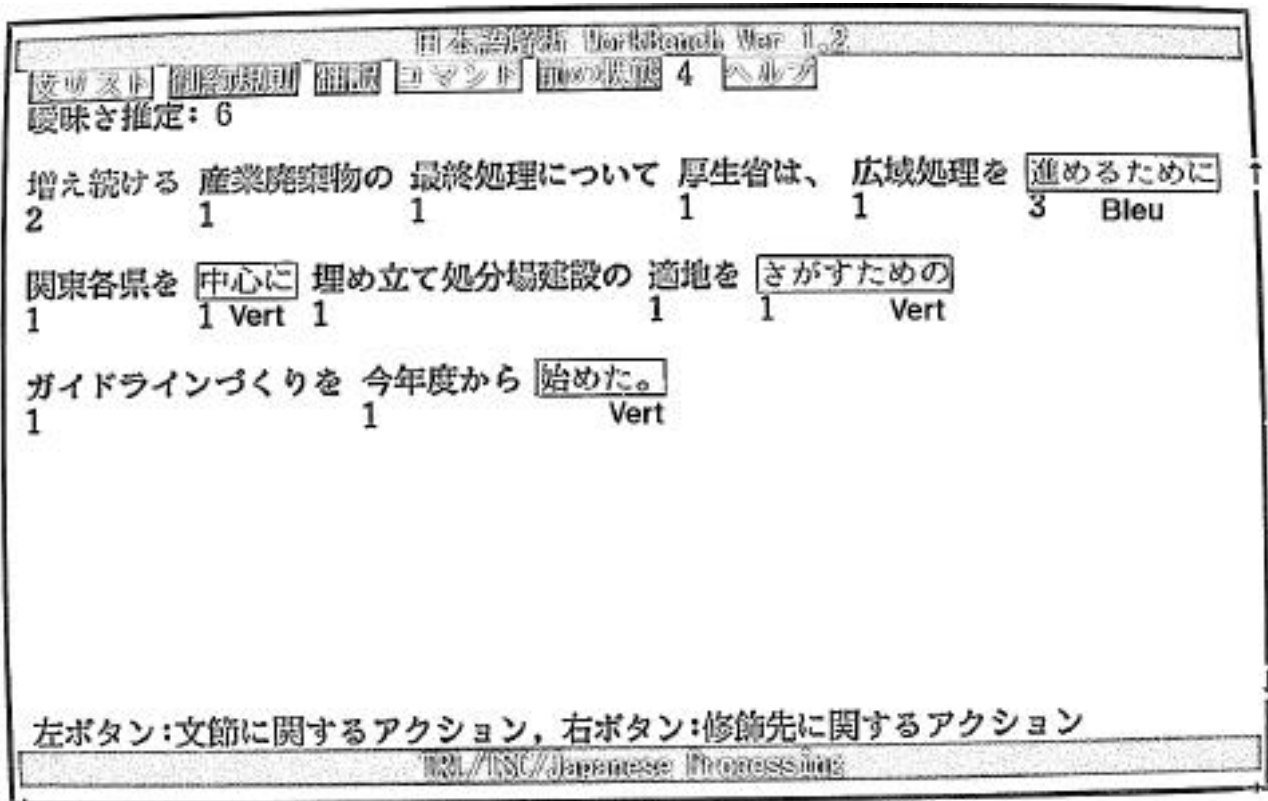


Figure 1.30 : Sélection d'un nouveau groupe

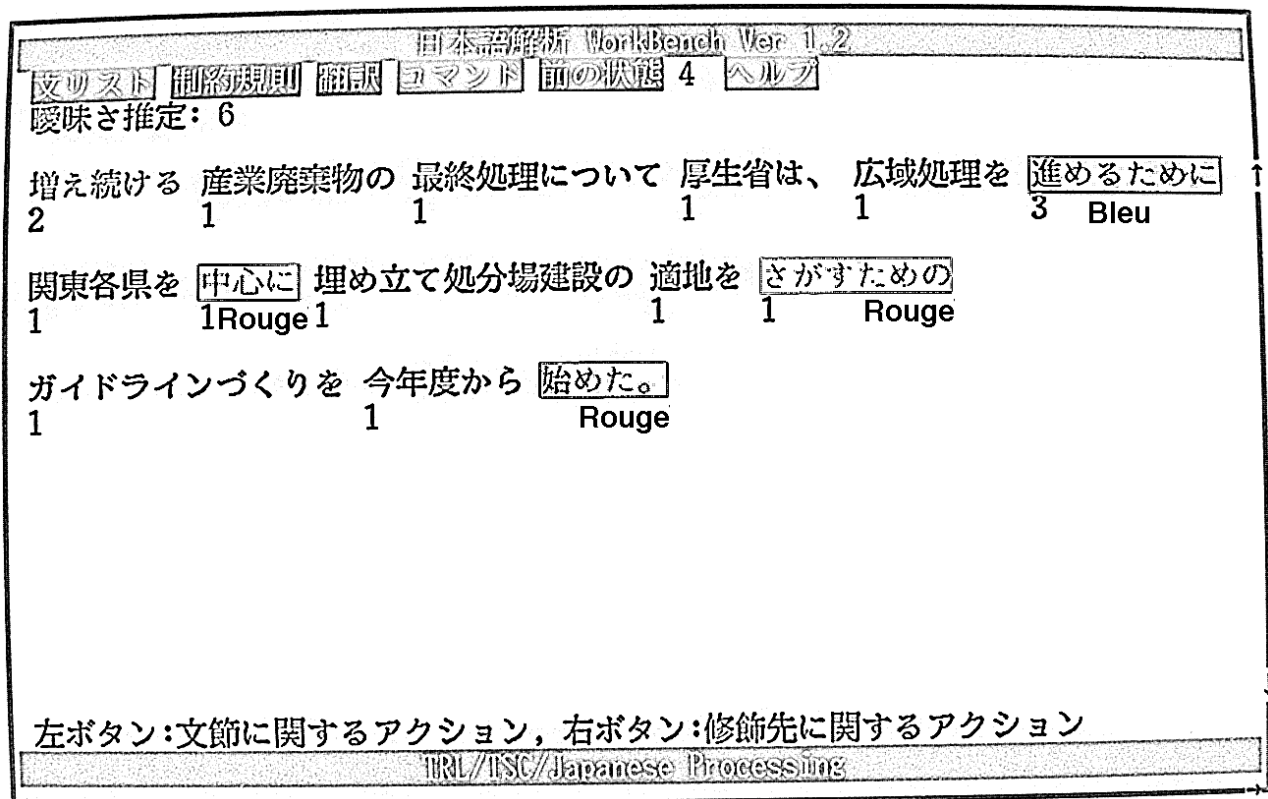


Figure 1.31 : Choix manuel du modifié, désambiguïsation terminée

f. Conclusion

Nous avons assisté à une démonstration de JETS et ne pouvons que vanter son interface originale. C'est la première fois que l'utilisateur peut manipuler presque directement le texte. La rétroaction s'adapte automatiquement en fonction de la position du pointeur. L'utilisateur peut donc "se promener sur le texte" et choisir dans quel ordre répondre aux questions (implicites), qui sont toutes en suspens. Avec KBMT-89, JETS est le seul système qui :

- propose plusieurs questions à l'utilisateur afin que celui-ci choisisse l'ordre des réponses qu'il va donner.
- modifie l'ensemble des questions qui demeurent en suspens après une réponse.

Le système offre trois niveaux de désambiguïsation : le sens des mots, le parenthésage et le rôle sémantique des groupes. Des expériences ont été conduites sur la perception de ces divers niveaux de désambiguïsation. Les comportements des utilisateurs sont très variables. Cependant, presque tous n'acceptent que la désambiguïsation sur le sens des mots, et semblent penser que la machine devrait se débrouiller avec la syntaxe.

1.3.4. ITS2 (1990-)

a. Historique

Le système ITS2 (Interactive Translation System 2) est un prototype de système interactif de traduction de phrases, en cours de développement depuis 1990 [Wehrli 1990] au Laboratoire d'Analyse et de Technologie du Langage (LATL) de l'Université de Genève. Dans son état actuel, il accepte des phrases prises dans un sous-ensemble (encore relativement restreint) de la grammaire du français pour les traduire en anglais. Le projet poursuit plusieurs objectifs, aussi bien théoriques que pratiques. Il s'agit de montrer l'apport de l'approche interactive au problème de la traduction automatique, ainsi que les avantages d'un système basé sur le modèle grammatical chomskyen du Gouvernement et Liage (Government and Binding theory, GB).

b. Fiche d'identité

Nom du projet	ITS2 (Interactive Translation System 2)
Langue(s) source(s)	français
Langue(s) cible(s)	anglais
Type de système	système de traduction interactif
Paradigme	entre la première et la seconde génération. Analyse basée sur le modèle grammatical chomskyen du Gouvernement et Liage. L'analyseur — FIPS (French Interactive Parsing System) — est décrit en détail dans [Laenzlinger & Wehrli 1991] et [Wehrli 1992]

Architecture informatique	développement sur un VAX (micro VAX), utilisation sur PC. Analyseur temps réel, processus d'analyse interrompu lorsqu'une question est en suspens
Corpus	corpus expérimental

c. Structure du système

Le système ITS2, dont l'architecture est donnée dans la figure 1.32, repose sur le modèle du transfert. On retrouve par conséquent les trois composantes fondamentales de tels systèmes : analyse, transfert, génération. Une phrase d'entrée en français (phrase source) est prise en charge par l'analyseur qui lui affecte une représentation syntaxique correspondant à une structure de surface enrichie de la grammaire GB (structure-S source)¹². Pour accomplir cette tâche, l'analyseur a besoin d'informations lexicales et grammaticales (lexiques et grammaires). Si nécessaire, il peut également demander des informations supplémentaires à l'utilisateur au moyen de dialogues de clarification.

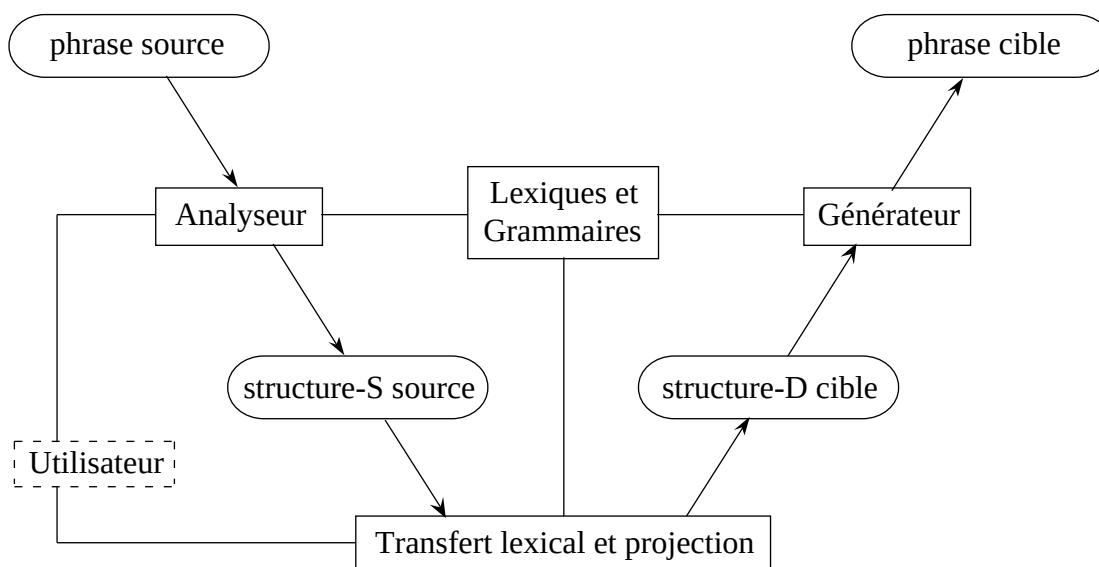


Figure 1.32 : Architecture du système ITS2

La composante de transfert est chargée d'élaborer une structure syntaxique abstraite en langue cible (structure-D) sur la base de la structure-S construite par l'analyseur. Le transfert consiste donc à mettre en correspondance à un certain niveau d'abstraction une structure source et une structure cible. Dans le système ITS2, le niveau de représentation utilisé pour le transfert s'apparente à celui de la structure profonde (structure-D) de la grammaire BG. Le transfert est dans une large mesure une question de correspondances

¹² La théorie GB distingue deux niveaux de représentation syntaxique, appelés respectivement structure-D et structure-S. Comme les structures profondes des versions antérieures de la théorie chomskyenne, les structure-D représentent, sous forme de structures canoniques, les relations entre prédicats et arguments. Plus généralement, ce niveau de représentation correspond à une projection de l'information lexicale (catégorie, sous-catégorisation, etc.) dans une structure $\bar{X}$. La structure-S représente l'organisation syntagmatique d'une phrase en tenant compte de l'ordre de surface des mots, ainsi que les traces des éléments déplacés par transformation.

lexicales entre les items lexicaux de la langue source et les items lexicaux de la langue cible, ainsi qu'une question de projection de structures sur la base d'informations lexicales.

La génération comporte deux sous-composantes, qui sont respectivement appelées la composante transformationnelle et la génération morphologique. La première a pour rôle la dérivation syntaxique des structures de surface (structure-S) à partir des structures-D par application de règles transformationnelles. Enfin, la génération morphologique est responsable du choix des représentations morphologiques des items lexicaux, ainsi que des éventuelles règles d'adaptation (élision, contraction, etc.).

d. Désambiguïsation avec ITS2

Dans le système ITS2, l'interaction entre le processus de traduction et l'utilisateur prend place à plusieurs niveaux [Wehrli 1993].

❖ Un premier niveau d'interaction se situe lors de l'analyse lexicographique, lorsque l'analyseur syntaxique "lit" la phrase source qui lui est soumise. À ce niveau, en effet, le programme peut rencontrer des mots inconnus, c'est à dire des mots qui ne figurent pas dans le lexique. Il peut s'agir de noms propres ou de noms communs qui ne se trouvent pas dans les dictionnaires, ou encore de mots mal orthographiés. Une fenêtre d'interaction est alors ouverte, exposant la phrase et mettant en évidence le ou les mots concernés. Un menu permet à l'utilisateur de choisir l'action la plus appropriée parmi:

- corriger la phrase,
- valider le mot comme un nom propre,
- créer une nouvelle entrée du dictionnaire,
- interrompre le processus.

❖ Un second niveau d'interaction prend place lors de l'analyse syntaxique. Il se peut que l'analyseur rencontre des difficultés qu'il est incapable de surmonter au moyen de ses règles et heuristiques. Un exemple typique d'une telle situation est celui des ambiguïtés d'attachement de syntagmes propositionnels, illustré ci-dessous.

- (1) a. Jean frappe le chien avec le collier.
b. Jean frappe le chien avec des taches blanches.
c. Jean frappe le chien avec le poing.

Dans la phrase (1a), le syntagme prépositionnel "avec le collier" peut se rattacher soit au syntagme nominal "le chien", et alors il est interprété comme modificateur de ce syntagme, soit au verbe "frapper", et alors il est interprété comme un complément de ce verbe. Le premier type d'interprétation est sans doute plus naturel avec la phrase (1b), le second avec la phrase (1c). Dans des cas de ce type, la fenêtre de dialogue affiche la chaîne de caractères correspondant au syntagme prépositionnel à attacher et les têtes des syntagmes auxquels il peut s'attacher. Pour la phrase (1a), cela revient à afficher le syntagme prépositionnel [pp avec le collier] et demander s'il faut l'attacher au syntagme nominal objet direct "chien" ou au verbe "frappe". Comme toujours, l'utilisateur a également la possibilité de modifier la phrase ou d'interrompre la session.

❖ Un autre type de problème qui exige une intervention de l'utilisateur est celui de l'interprétation des pronoms, comme dans les exemples suivants :

- (2) a. Jean parle à Marie de son dernier livre.
 b. Jean a appelé Paul parce qu'il lui doit beaucoup d'argent.
 c. Les enfants se sont lavés.

Contrairement aux exemples précédents, le problème de l'ambiguïté des pronoms ne se pose pas en général pendant la phase d'analyse, mais pendant celle de la génération, lorsqu'il s'agit de déterminer la forme du pronom cible.

Le possessif "son" dans l'exemple (2a) peut renvoyer soit à "Jean", soit à "Marie", soit encore à une autre personne mentionnée précédemment dans le discours. Si la langue cible est l'anglais, la forme du pronom possessif dépend du genre de son antécédent (his/her). Dans un tel cas, la fenêtre d'interaction affiche la liste des antécédents potentiels et demande à l'utilisateur de faire le bon choix.

La phrase (2b) contient deux pronoms, "il" et "lui". Chacun de ces pronoms peut prendre comme antécédent, "Jean", "Paul", ou éventuellement une tierce personne non mentionnée dans la phrase. Les principes grammaticaux de la théorie du liage éliminent la possibilité d'un même antécédent pour les deux pronoms. Il reste donc deux combinaisons de référence si l'on exclut toute référence contextuelle, six sinon. Pour la seconde question, on tirera donc partie de la réponse donnée lors de la première recherche d'antécédent.

Dans le cas de la phrase (2c), l'ambiguïté concerne l'interprétation du pronom anaphorique "se". Dans cette phrase, "se" peut être compris au sens réfléchi ou réciproque. La fenêtre d'interaction affiche l'information suivante :

Les enfants se sont lavés.

- a– les uns les autres ? (réciproque)
 b– eux-mêmes ? (réfléchi)

❖ Enfin, de très nombreuses ambiguïtés se présentent au niveau du transfert, et notamment du transfert lexical, lorsque le système doit décider comment un élément lexical de la langue source doit être traduit. On sait que la correspondance entre items lexicaux de deux langues n'est pratiquement jamais univoque. Autrement dit, il y a pratiquement toujours plusieurs possibilités de traduction pour un mot donné, même lorsque l'environnement syntaxique de ce mot est bien cerné. Il est légitime de penser que le dictionnaire bilingue contient toutes les correspondances possibles pour un item donné. On peut afficher cette liste, si l'on n'est pas face à un utilisateur qui connaît la langue cible. Comme on veut une action en langue source, on propose d'effectuer la discrimination sur la base des différentes acceptions du mot. On affichera alors le mot avec ses différents sens, en utilisant des synonymes. Par exemple, dans le cas (très simplifié) du mot "assurance", on affichera

assurance syn. confiance

assurance syn. contrat

La sélection du sens le plus approprié d'un mot, qui permet au système de choisir une traduction correcte, est basée sur des exemples ou des synonymes associés aux entrées lexicales.

e. Conclusion

ITS propose différents modes de dialogue. Le système peut travailler en mode autonome et à l'autre extrême, en mode très interactif. L'interaction syntaxique se fait déjà en langue source. L'interaction pour le transfert est encore réalisée en demandant à l'utilisateur de choisir le bon correspondant parmi une liste de mots anglais. À terme, les concepteurs visent une interaction complète en langue source.

Il faudra attendre que le système atteigne un nouveau stade de développement pour pouvoir le comparer avec les autres projets que nous avons étudiés.

Chapitre 2

Une voie nouvelle : la TAO individuelle ou “TAFD pour tous”

La TA interactive pour auteur monolingue est une idée récurrente. Bien qu'elle semble être une bonne idée, les prototypes, voire les systèmes antérieurs ont buté sur certains obstacles et n'ont pas (encore) conduit à des systèmes opérationnels.

Ces difficultés nous semblent être de plusieurs types :

- ergonomiques : utilisation de dialogues trop spécialisés ou trop prenants, fenêtres multiples,
- traductionnelles : même si le système offre une bonne ergonomie (JETS), les utilisateurs visés ne sont pas les bons,
- linguistiques : si le spectre est trop large, il y a trop de questions.

La solution interactive a d'abord été présentée en général dans le cadre d'outils d'aide au traducteur [Kay 1980 ; Melby 1982]. Souvent, un tel système propose sa traduction suivie d'une phase plus ou moins conviviale de postédition ; dans les cas les plus ambitieux — ITS et ALPS par exemple — l'interaction entre le système et l'utilisateur a lieu au cours du traitement lui-même. L'utilisateur répond aux questions concernant la désambiguïsation du texte source et des problèmes lexicaux et stylistiques du texte cible. Le système et l'utilisateur doivent l'un et l'autre connaître les deux langues. L'expérience a montré que les utilisateurs — même ceux qui étaient au début très bien disposés — finissent par devenir frustrés par un système qui n'apprend jamais rien et recommence les mêmes erreurs banales.

Au lieu d'outils de THAM (Traduction Humaine Assistée par la Machine), on a commencé, vers 1986, à concevoir des systèmes interactifs pour des utilisateurs monolingues voulant traduire leurs propres textes. Cette idée initialement proposée par M. Kay [Kay 1973], a donc subi un purgatoire de plus de quinze ans, avant d'être adoptée presque simultanément par un bon nombre de chercheurs [Carbonell & Tomita 1985 ; Johnson & Whitelock 1985 ; Schubert 1986 ; Whitelock, *et al.* 1986]. Dans un tel système, on suppose que l'utilisateur sait bien ce qu'il veut dire mais ne sait pas comment

le dire dans la langue cible, alors que le système sait bien comment traduire, mais demande à l'utilisateur de l'aider à comprendre le texte source. Dans ce scénario, il est évident que l'interaction homme-machine risque d'être assez complexe. Les différentes variantes de ces scénarios ont été regroupées sous le terme général de "TAFD", introduit à l'occasion du premier séminaire sur le sujet, DBMT-90 [Blanc & Boitet 1990], organisé par le GETA, ATR et l'UMIST au Sappey-en-Chartreuse (Isère, France).

En proposant le paradigme de "TAFD pour tous", nous espérons apporter une réponse adaptée en intégrant les meilleures idées et en proposant de nouvelles réponses à de nombreux problèmes théoriques, techniques, et ergonomiques. Le cadre de cette étude est celui du projet LIDIA.

2.1. Contraintes spécifiques de la TAFD¹³

Comme nouveau paradigme, la TAFD induit des contraintes spécifiques. Ces contraintes concernent : le matériel et le logiciel à mettre en œuvre, le linguiciel à développer et les facteurs ergonomiques que doit satisfaire un système de TAFD.

2.1.1. Contraintes matérielles et logicielles

Pour la TAFD, la station de travail du rédacteur aura un coût limité, donc une puissance de calcul et de stockage moyenne¹⁴. Il faudra probablement partager le linguiciel — complexe —, et donc y accéder via un serveur. Le système sera alors probablement distribué. On peut imaginer plusieurs scénarios de distribution.

Dans un laboratoire de recherche, dans le service de rédaction de documentation technique d'une société exportatrice, dans une entreprise de communication, on peut imaginer que l'on a un réseau de stations de travail. Sur ce réseau, on dispose d'un serveur de communication qui permet à chaque station de travail d'envoyer des requêtes à un serveur de traduction distant.

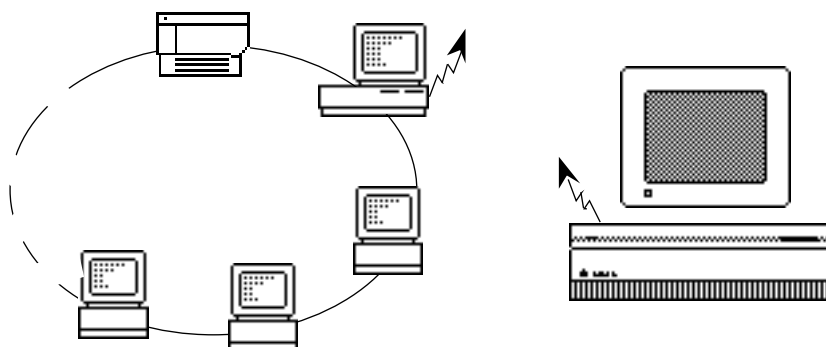


Figure 2.1 : Une première architecture de distribution des stations de rédaction en réseau accèdent directement à un serveur de traduction délocalisé via un unique serveur de communication

¹³ Ce paragraphe est une extension de [Boitet 1993], lui-même version française étendue de [Boitet & Blanchon 1993b]. Les parties reprises ou adaptées sont en style citation.

¹⁴ Il semble raisonnable de penser que l'utilisateur puisse disposer d'une machine dotée de 10 à 16 Megaoctets de mémoire vive et de 100 à 200 Mégaoctets de mémoire de masse.

On peut aussi imaginer que le serveur de traduction soit localisé.

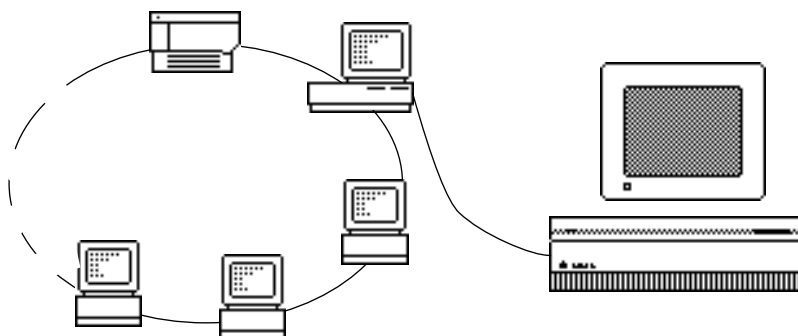


Figure 2.2 : Une seconde architecture de distribution des stations de rédaction en réseau accèdent au serveur de traduction local via un unique serveur de communication.

On peut aussi imaginer que sur un réseau comportant peu de machines dédiées à la traduction, les stations de travail prennent en charge la communication avec le serveur de traduction. On peut imaginer ce scénario en connexion directe sur un serveur local, ou en connexion via une ligne téléphonique sur un serveur délocalisé.

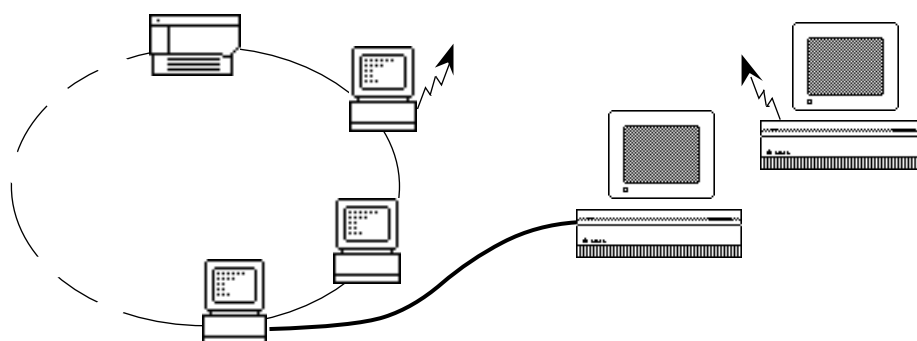


Figure 2.3 : Une troisième architecture de distribution les stations de rédaction prennent en charge le serveur de communication.

On peut aussi imaginer que la station de travail d'une personne isolée, travaillant à la maison, soit connectée à un serveur de traduction via un minitel qui sert d'interface de communication. Le serveur de communication est alors implémenté sur la station de travail, comme dans le premier scénario.

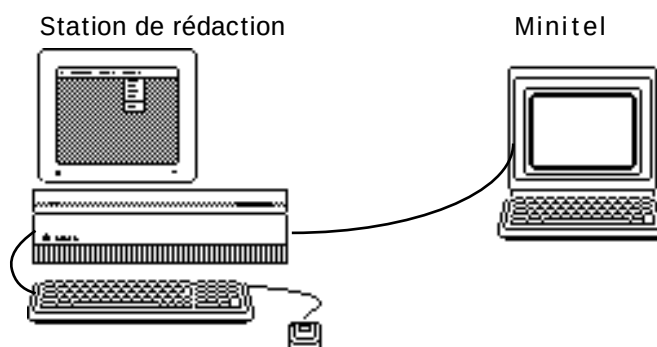


Figure 2.4 : Configuration privée pour la TAFD

Il est souhaitable qu'un système de TAFD soit inclus dans un environnement de production de documents comme fonctionnalité supplémentaire (à la JETS). Cela évitera à l'utilisateur d'avoir à apprendre un nouvel environnement de rédaction, et ne diminuera donc pas sa productivité.

Il faut aussi éviter une trop forte surcharge de l'interface du logiciel hôte par un trop grand nombre de commandes. Il ne faut pas non plus transformer la façon dont l'utilisateur interagit avec le système hôte. Pour cela il faut que l'outil de TAFD fonctionne de façon transparente et requière l'intervention de l'utilisateur sans jamais l'interrompre de façon préemptive.

La solution qui consiste à intégrer de nouveaux services est largement praticable aujourd'hui. En effet, les grands outils de production de documents offrent déjà, ou intégreront dans un proche avenir, des API (Application Program Interface) ou un mécanisme similaire. Avec de tels mécanismes, il est possible d'ajouter de nouvelles fonctionnalités à un environnement sans en modifier le comportement. Interleaf, FrameMaker, Grif, Word et HyperCard sont d'ores et déjà des hôtes possibles pour un système de TAFD.

2.1.2. Contraintes linguicielles

Dans la majorité des situations adaptées à la TAFD, il faut un système de couverture lexicale et grammaticale très large. Cela amène à poser aux théoriciens des questions 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 nécessaire* ? Typiquement, un système de TA Fondée sur la Langue contient de 3.10^4 à 3.10^5 termes, en 2 langues. Le cas de METEO (3.10^3) est atypique, à cause de son domaine très restreint. 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, 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 linguistique d'un système de TAFD accessible à un utilisateur naïf*.

2.1.3. Contraintes ergonomiques

Les aspects ergonomiques sont bien sûr importants en TAO classique. En TA du veilleur ou de l'auteur, il y a 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, ou pire en colonne. De même, on accepte plus volontiers des choix portant sur de petits groupes de mots que sur des phrases complètes. Les constructeurs de systèmes ont 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 absolument crucial, et les choix ergonomiques influencent directement l'architecture de tout le système. Les choix principaux sont les suivants :

- Le système doit-il fonctionner en temps réel, ou l'asynchronie est-elle préférable ?
- Doit-il tourner sur des ordinateurs personnels bon marché, ou sur des stations de travail ? Une architecture avec serveur est-elle possible ? Si oui, pourrait-on simplement connecter un PC au minitel ? On a donné plus haut une ébauche de réponse à ces questions.
- Comment les dialogues doivent-ils être organisés ? Est-il nécessaire et/ou possible de les conduire dans un environnement multimédia ? Plus précisément, l'utilisation de synthèse de parole peut-elle améliorer l'efficacité et la convivialité des dialogues de désambiguïsation ?
- Comment prévoir différents niveaux dans le dialogue selon que l'on s'adresse à un utilisateur débutant ou à un utilisateur confirmé ayant découvert les informations manipulées par le système et souhaitant un dialogue plus synthétique que le premier ?

2.2. Architecture linguistique et voies intermédiaires nouvelles

Le cadre nouveau de la TAFD peut amener à rechercher de nouvelles architectures linguistiques. Il ne s'agira sans doute 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.

2.2.1. Transfert multiniveau à acceptions, propriétés, et relations interlingues

Les systèmes de TAO modernes 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 GETA entre 1960 et 1970, que B. Vauquois appelait le “pivot hybride” : la structure interface source fournie au transfert contient des éléments lexicaux de la langue source, et des propriétés et relations interlingues (traits sémantiques, cas profonds, nombre, aspect, modalité, détermination abstraits...). Dans un “interlingua”, les éléments lexicaux sont de plus interlingues (les auteurs des systèmes actuels parlent de “concepts”, mais il n'y a pas toujours d'ontologie).

Le transfert (sémantique) multiniveau, au sens de Vauquois [Vauquois 1988 ; Vauquois & Boitet 1985] 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...).

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 sens¹⁵.

Remarquons aussi que “interlingue” ne signifie pas “indépendant des langues”. Par exemple, si le système travaille avec le français, l'anglais et le russe, il y aura une seule acception pour

¹⁵ 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.

“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).

2.2.2. Langages guidés

a. Séparation lexicale/grammaire

La notion de “sous-langage” a été introduite et étudiée par le linguiste R. Kittredge [Kittredge 1983 ; Kittredge 1985 ; Kittredge 1986 ; Kittredge & Lehrberger 1982] à la suite de son expérience en TAO (il fut directeur du groupe TAUM de l’Université de Montréal au début des années 70). Kittredge 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 “sous-optimisation”) était prometteuse, et les a appliqués à un certain nombre de types de textes.

Son analyse, extrêmement complète, 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 en gros 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) productibles dans des conditions fixées (par exemple, bulletins météo, appels d’offres du Secrétariat d’Etat, 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-langage 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).

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 thesaurus, en particulier la synonymie, la quasi-synonymie et l’hypéronymie, et des relations analogues aux relations lexico-sémantiques entre termes (comme entité $\mathcal{A}\mathcal{E}$ qualité, action $\mathcal{A}\mathcal{E}$ 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. En TAFD, cela peut servir à présélectionner le sens le plus probable.

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. Type de textes : “style d'énoncés”, “genre de textes” et “type de documents”

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 textes plus longs¹⁶. 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 convient¹⁷.

Style d'énoncé

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.

Genre de texte

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, qui peuvent traiter des textes beaucoup plus longs que les outils linguistiques actuels, le plus souvent limités à la phrase ou au paragraphe.

On peut envisager deux familles de solution à la définition des genres de textes.

❖ Boitet [Boitet & Blanchon 1993a] propose de définir un genre de texte comme une expression algébrique sur les styles d'énoncés et les genres de textes. Par exemple, le

¹⁶ Les termes de “microlangage” et de “sous-langage”, proposés dans [Boitet 1990b], se sont révélés surchargés et anti-intuitifs.

¹⁷ Par exemple, les grammaires hors-contexte (CFG), avec ou sans attributs, les TAG, les STCG [Boitet & Zaharin 1988 ; Vauquois & Chappuy 1985]. Pour le moment, nous utilisons ROBRA dans LIDIA-1, malgré son caractère procédural.

genre de texte S1 des paragraphes commençant par une phrase de style M1 suivie par une suite (éventuellement vide) de phrases de style M2 peut être définie par une simple expression régulière :

$\langle S1 \rangle$	$= M1 M2^*$
----------------------	-------------

Pour décrire le genre de textes d'un "document" commençant par un titre (microlangage M3), et se poursuivant par une liste non vide de paragraphes (microlangages M2 et/ou M4) et/ou de sections de même structure qu'un document, on peut de même écrire¹⁸ :

$\langle \text{Document} \rangle$	$= \langle \text{Title} \rangle \langle \text{Content} \rangle$
$\langle \text{Title} \rangle$	$= M3$
$\langle \text{Content} \rangle$	$= (\langle \text{Paragraph} \rangle \mid \langle \text{Document} \rangle)^+$
$\langle \text{Paragraph} \rangle$	$= (M2 \mid M4)^+$

❖ À partir du travail de Denos [Denos 1993], on pourrait définir les genres de textes comme des n-grammes sur les styles d'énoncés. Cette solution simple et élégante permet d'introduire des probabilités de succession sur les styles d'énoncés pour un genre de texte donné. Denos propose aussi d'introduire la notion de type de document qui permettrait de définir des paramètres plus globaux.

Il faudra plus de recherche pour trouver 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.

Type de document

Denos [Denos 1993] propose d'affecter un type de document à chaque pile HyperCard. Nous retenons cette idée, qui mérite une étude plus approfondie.

Les travaux de Laurian [Laurian 1983a ; Laurian 1983b ; Laurian 1986 ; Laurian 1987] apportent des éléments en faveur de l'existence d'une corrélation entre les propriétés d'objectif de communication et d'intention, et les propriétés de syntaxe et de style. Laurian [Laurian 1983b] suggère aussi que des informations non linguistiques peuvent être utilisées pour caractériser le type d'un document. Il s'agit des caractéristiques sociales de l'auteur et du lecteur visé, des caractéristiques du message transmis, ainsi que des emplois typographiques spéciaux, de la fréquence des sous-titres, des mots écrits en majuscules ou en italiques, des parenthèses, etc.

L'étude de Bonzi [Bonzi 1990] met en évidence un certain nombre de motifs syntaxiques qui "séparent" les textes relevant des disciplines des sciences sociales de ceux relevant des sciences exactes. Les variables qui révèlent un contraste frappant sont des caractéristiques *obligatoires* des propositions (voix, modèle, temps des verbes, types de proposition). Au contraire, les caractéristiques syntaxiques *optionnelles* semblent peu discriminantes par rapport à ces familles de sous-langages. Elles distinguent parfois des sous-langages de manière individuelle, comme si elles étaient liées à la discipline elle-même.

¹⁸ Un symbole entre crochets désigne un genre de texte, "|" l'alternation, "*" et "+" la répétition. On pourrait ajouter des conditions sur des attributs associés aux symboles principaux.

Pour le discours technique, Selinker, Trimble et Vroman [Selinker, *et al.* 1972] ont établi que les fonctions rhétoriques déterminent des choix grammaticaux au niveau de la phrase. Par exemple, le présent est utilisé pour exprimer des idées centrales, alors que le passé est utilisé pour donner des informations supplémentaires.

Tagliacozzo [Tagliacozzo 1978] a mis en évidence les différences significatives intervenant entre les textes techniques présentant une haute technicité, et les moins "pointus". Le nombre de mots-outils diffère entre les deux types de texte, et il y a un plus grand degré de subordination sur les noms dans le type "très technique".

Slocum [Slocum 1984] a étudié les couples de manuel d'instruction et de brochures de vente de deux types de produits manufacturés. Il est apparu que les manuels ont un caractère directif, alors que les brochures ont un caractère descriptif.

2.2.3. 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 révision¹⁹.

En TAFD, il nous semble que l'information ne doit pas être simpliste, mais peut-être moins fouillée que dans le cas de la TAFL (Fondée sur la Langue). Par exemple, il est sans doute inutile d'avoir un système trop riche de codes sémantiques. 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 complets.

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, comme l'aspect, 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).

2.3. Situations pour la TAFD

La TAFD n'est pas proposée comme une panacée, mais nous semble être devenue nécessaire, à cause de la croissance explosive des usages de l'informatique, de plus en plus individuelle et distribuée. Parallèlement, les progrès impressionnants réalisés ces dernières années en informatique (matériel, logiciel, interfaces) et Traitement Automatique des Langues Naturelles (on a gagné un ordre de grandeur sur la taille des

¹⁹ 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% !

bases lexicales et progressé en LSPL²⁰) nous laissent penser que ce paradigme est aussi devenu possible en pratique.

2.3.1. Limitation des paradigmes actuels

Le développement de la bureautique a fait naître de nouvelles techniques de rédaction. Depuis l'émergence de la micro-informatique, le micro-ordinateur est devenu un moyen de rédaction très utilisé.

L'arrivée de l'hypertexte permet de rédiger des documents attrayants, interactifs, et clairs en utilisant au maximum l'interaction lors des phases de conception et de rédaction.

De plus en plus, en particulier dans le contexte européen, ces documents doivent être traduits en diverses langues. Cependant, les techniques de TAO "classique" ne sont pas adaptées, soit qu'elles fournissent une qualité trop faible (TAO du veilleur), soit qu'elles nécessitent l'intervention de spécialistes polyglottes (TAO du réviseur, station de travail du traducteur), en nombre de plus en plus insuffisant au regard de l'accroissement de la demande potentielle. Rédiger en anglais n'est pas non plus une solution acceptable, ni du point de vue culturel ni du point de vue économique (maîtrise d'expression, rapidité de rédaction).

Dans ce contexte, il faudra accepter de traduire des documents traitant de n'importe quel sujet, en atteignant une qualité suffisante pour ne pas avoir besoin de révision. Or, on sait depuis Bar-Hillel (1960) [Bar-Hillel 1960] que cela est impossible sans une compréhension profonde.

Mais nous ne pouvons viser une *compréhension "explicite"* et suivre, par exemple, la voie du CMT [Nirenburg & al. 1989], qui consiste à coder une grande quantité d'informations sur un domaine donné, puisque l'on ne connaîtra pas, a priori, le domaine du document traité.

On ne peut pas non plus suivre la voie de la *compréhension "implicite"* (TAO du réviseur) qui consiste à réduire le type des textes traités à un sous-langage, pour lequel on développe des heuristiques fondées sur l'observation de régularités formelles. En effet, on ne peut ici se borner à un seul sous-langage. En tout état de cause, les heuristiques liées à un sous-langage ne permettent jamais de résoudre les problèmes à 100%, d'où la nécessité de postéditer pour arriver à la qualité souhaitée.

Reste alors la voie de la *compréhension "indirecte"*, dans laquelle c'est le rédacteur qui aide le processus de traduction à travers un dialogue de standardisation et de clarification (lexicale, syntaxique et sémantique). La connaissance qu'un auteur a de ce qu'il veut dire excédera toujours, d'ailleurs, celle d'un système fondé sur l'expertise d'un domaine, aussi perfectionné soit-il.

2.3.2. 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 $\nrightarrow$ n, comme pour la dissémination de documentation technique, ou n $\nrightarrow$ n, comme dans des projets internationaux) ;
- l'entrée ou le domaine ne doivent pas être trop contraints ou contrôlés (sinon, mieux vaut utiliser la TA Fondée sur la Langue : TAFL, ou la TA Fondée sur la Connaissance : TAFC) ;

²⁰ LSPL : Langage Spécialisé pour la Programmation Linguistique.

- 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 annotations de désambiguïsation, etc.), et en lui laissant si possible le choix entre plusieurs média.

2.3.3. Situations traductionnelles adaptées à la TAFD

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 DON²¹, par exemple dans les 9 langues de la CEE (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 orale aussi bien qu'écrite ;
- l'échange télématique de notes et de documents de travail dans des projets internationaux.

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), ou encore la préparation d'une présentation vidéo dans plusieurs langues ;
- 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) semblerait indiquée.

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, résultant d'une négociation avec un expert. C'est par exemple le cas de lettres officielles ou formelles, qui ont des structures très différentes dans différentes cultures. Somers & Tsujii (Somers & al. 1990) ont proposé le terme de "traduction sans texte source", mais il est peut-être plus exact de parler de génération multilingue de textes que de TAFD.

²¹ Disque optique numérique (CD-ROM), contenant près de 600 Mo.

Chapitre 3

LIDIA-1, une maquette de TAFD

Pour mettre en pratique l'idée de la TAFD pour tous, le GETA a lancé, voici quatre ans, le projet LIDIA (Large Internationalisation des Documents par Interaction avec l'Auteur). Nous présentons maintenant le résultat du travail de spécification de la première maquette : LIDIA-1. Cette maquette doit permettre de simuler de manière convaincante les fonctionnalités de traduction du poste du rédacteur, d'étudier les problèmes linguistiques nouveaux, et d'expérimenter diverses architectures logicielles. L'état actuel de la maquette décrit au dernier paragraphe de ce chapitre permet de se faire une idée du travail accompli.

3.1. Situation concrète retenue²²

Les choix que nous avons faits pour la maquette concernent les langues manipulées, le type de document à traduire et l'architecture du système.

3.1.1. Une langue source, plusieurs langues cibles

Pour LIDIA-1, on suppose qu'un ingénieur français crée de la documentation technique sous la forme d'une pile HyperCard [Apple Computer 1989], sur un Macintosh de puissance moyenne, et aide le système à la traduire en anglais, allemand et russe. Le choix des langues est uniquement dû aux compétences disponibles, et le choix d'une situation monosource et multicible permet évidemment de réduire le coût de l'opération en n'ayant qu'une analyse à construire ! Il y a des arguments moins contingents pour les deux autres choix.

3.1.2. Traduction de piles HyperCard de documentation technique

Avec l'arrivée d'HyperCard™, les hypertextes [Daniel-Vatonne 1990 ; Nielsen 1990 ; Savoy 1990] sont sortis des laboratoires. Pour un prix très modique, n'importe qui peut réaliser des documentations vivantes, des animations, etc., avec image et son en prime. Certaines

²² Le § 3.1. est une reprise et une extension de [Boitet 1993], lui-même extension en français de [Boitet & Blanchon 1993b]. Les parties reprises ou adaptées sont en style citation.

documentations de voitures (Renault, Peugeot) sont diffusées sur Disque Optique Numérique sous HyperCard (8.000 pages de documentation, en 10 langues, avec figures et éventuellement messages oraux, sous forme phonétique codée, 30 à 40 fois plus compacte que le signal).

Du point de vue ergonomique, l'hypertexte privilégie l'interaction. Par conséquent, on peut penser qu'un rédacteur sera plus prêt à accepter une interaction linguistique sous hypertexte que sous traitement de texte. Dans le premier cas, on reste dans la logique de l'outil. Dans le second, il faut changer ses habitudes, ce qui est souvent fort difficile.

Du point de vue linguistique, les parties textuelles sont bien isolées, et connexes, au contraire des traitements de texte ou des logiciels de PAO, où l'on trouve un mélange de codes de formatage, de texte, de figures, de formules, le tout parfois présenté de façon non connexe (tableaux avec tabulations...). De plus, les fragments de texte d'une pile HyperCard typique (champs et boutons) sont petits, et grammaticalement très homogènes : on peut parler de "types de fragments" beaucoup mieux que de "types de documents". Par exemple, un champ donné peut contenir, dans chaque carte, un titre avec verbe à l'infinitif ("Naviguer dans HyperCard"), un élément de menu fonctionnant comme un nom dans d'autres contextes ("Quitter" — "...cliquer sur Quitter"), un ordre à l'infinitif ("Prendre la disquette"), une explication simple sans sujet pour la première phrase ("vous permet de quitter l'application et..."), ou encore un titre simple sans verbe ("Rotation à gauche"), etc.

Le concept de TAFD n'impose pas l'utilisation d'hypertexte. On pourrait également le mettre en œuvre dans le cadre de "texteurs" ou de "documenteurs" plus classiques. Certains, comme WinText™, contiennent déjà des annotations indiquant, outre la fonte, le corps et le relief, la langue naturelle de segments de texte. En ajoutant le type de fragment, et en développant une interface analogue, on pourrait les étendre pour y introduire les fonctions précédentes. Malgré tout, il faudrait se limiter à des texteurs ou documenteurs programmables (comme Interleaf™), ou modifier les codes sources. Le même problème se pose d'ailleurs aux concepteurs de "postes du traducteur".

3.1.3. Architecture distribuée et traitements asynchrones

Nous désirons que la station de rédaction soit un micro-ordinateur convivial et largement diffusé, d'où le choix du Macintosh sous HyperCard. Nous désirons aussi réutiliser la puissance de notre générateur de systèmes de TAO Ariane-G5 (sur Ariane, le lecteur pourra consulter [Boitet 1982]) pour écrire les parties "lourdes" du traitement linguistique. Or, il ne tourne que sur mini IBM (Eurolang l'a en partie porté sur SUN en 1992), mais pas sur Macintosh. Quand bien même il serait disponible, son exécution, même en tâche de fond, serait trop lente sur ce type de matériel, et augmenterait inévitablement les temps de réponse.

D'autre part, l'exemple du système CRITIQUE [Heidorn, *et al.* 1982 ; Jensen, *et al.* 1993 ; Richardson 1985] a montré qu'un traitement asynchrone convenablement organisé pouvait donner un système très convivial. Nous avons donc opté pour un traitement distribué entre la station de rédaction et un serveur de TA, et un fonctionnement asynchrone, pour ne pas pénaliser le rédacteur par des interruptions forcées ou des attentes interminables. Tout doit rester sous son contrôle.

Enfin, ce type d'organisation devient de plus en plus réaliste, avec la disponibilité de réseaux télématiques puissants et relativement peu coûteux. Il est tout à fait possible d'imaginer un système de TAFD où le micro se connecte au serveur à des intervalles réglables par l'utilisateur, et où la connexion ne dure que quelques secondes, tout comme un utilitaire de messagerie.

Les schémas suivants (figures 3.1 et 3.2) montrent les deux organisations retenues pour la maquette LIDIA-1.

Dans le premier cas, la station de rédaction — un Macintosh II fx — héberge le serveur de communication avec le serveur de traduction — un IBM 4361 —, qui est local. La carte Avatar permet de se connecter avec l'IBM au moyen d'un câble coaxial.

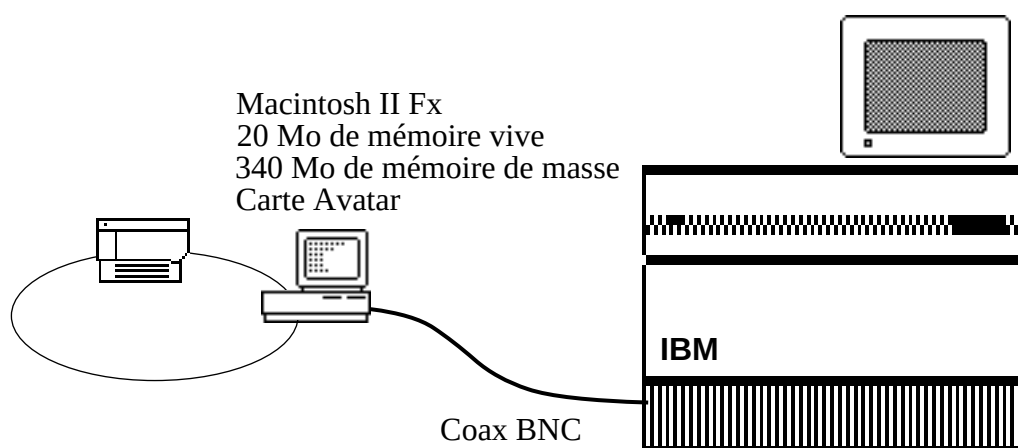


Figure 3.1 : Station de traduction hébergeant le serveur de communication.

Dans la seconde version, une machine dédiée — un Macintosh SE — héberge le serveur de communication. Une station de rédaction envoie des requêtes au serveur de communication qui se charge de les transmettre au serveur de traduction. Le serveur de communication se charge aussi de tous les échanges de fichiers occasionnés par les traitements.

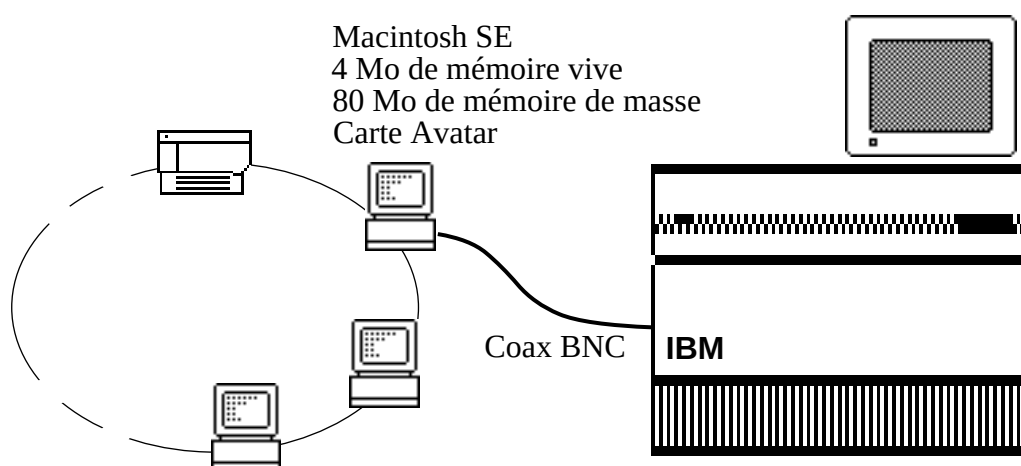


Figure 3.2 : Serveur de communication autonome

3.2. Choix fonctionnels pour le maquettage

Pour réaliser le maquettage, nous avons fait certains choix fonctionnels, liés d'abord aux acquis du GETA — Ariane-G5 —, mais aussi aux moyens que nous voulions mettre en œuvre au sujet des modalités d'interaction à proposer entre le système et l'utilisateur; et finalement à l'organisation du processus de traduction lui-même.

3.2.1. Utilisation du générateur Ariane-G5 sur IBM²³

a. Principes généraux

Ariane-G5 est un générateur (G) de systèmes de TAO reposant sur cinq langages spécialisés pour la programmation linguistique (LSPL). Chacun de ces langages est compilé, et les tables internes produites sont fournies en paramètres au "moteur" du langage.

Bien qu'il soit particulièrement bien adapté à l'approche transfert, il ne l'impose pas. À part les limites d'implémentation, la seule contrainte forte est que les structures de représentation des unités de traduction doivent être des arbres décorés.

Ariane-G5.3 tourne sous VM/ESA(CMS)²⁴, sur architecture 370/390 (gros ordinateurs, minis, gros micros), et représente environ 400.000 lignes sources, plus 30.000 lignes de messages par langue de dialogue²⁵. Les exécutables occupent environ 10 Mo sur un "minidisque" d'une machine virtuelle particulière. Pour qu'une machine virtuelle soit une "machine utilisateur", il suffit d'effectuer une connexion logique au minidisque Ariane. Ariane gère alors sur le minidisque utilisateur deux bases de données spécialisées, l'une contenant les linguiciels et l'autre les textes (au maximum 3.000 langues sources et langues cibles, 1.000 "corpus", et 10.000 textes par corpus).

b. Types de traitements

En Ariane-G5, la traduction se fait en trois "étapes" successives : analyse, transfert et génération. Chaque étape est réalisée en au moins deux et au plus quatre "phases" successives, reliées ou non par des "articulations", qui sont en première approximation de simples "changements de coordonnées".

En analyse, on trouve successivement les phases :

AM (analyse morphologique)	obligatoire,	écrite en ATEF ;
AX(analyse expansive X)	facultative,	écrite en EXPANS ;
AY(analyse expansive Y)	facultative,	écrite en EXPANS ;
AS (analyse structurale)	obligatoire,	écrite en ROBRA.

En transfert, on trouve successivement les phases :

TL (transfert lexical)	obligatoire,	écrite en EXPANS ;
TX (transfert expansif X)	facultative,	écrite en EXPANS ;
TS (transfert structural)	obligatoire,	écrite en ROBRA ;
TY (transfert expansif Y)	facultative,	écrite en EXPANS.

²³ Les points a. et b. sont repris de [Boitet 1990a]

²⁴ VM/ESA est un hyperviseur qui simule un ensemble de "machines virtuelles", chacune tournant sous un système d'exploitation propre. Par exemple, on peut faire tourner en même temps des machines virtuelles sous MVS/TSO, sous CMS, et sous AIX. CMS est un système d'exploitation interactif mono-utilisateur puissant, en général utilisé pour le développement de logiciels ou pour des applications interactives.

²⁵ Ariane-78.4 avait deux versions parallèles, en français et en anglais. Ariane-G5 est programmé de façon totalement multilingue, mais la seule version complète des messages est actuellement en français.

En génération, on trouve successivement les phases :

- GX (génération expansive X) facultative, écrite en EXPANS ;
 GS (génération syntaxique) obligatoire, écrite en ROBRA ;
 GY (génération expansive Y) facultative, écrite en EXPANS ;
 GM (génération morphologique) obligatoire, écrite en SYGMOR.

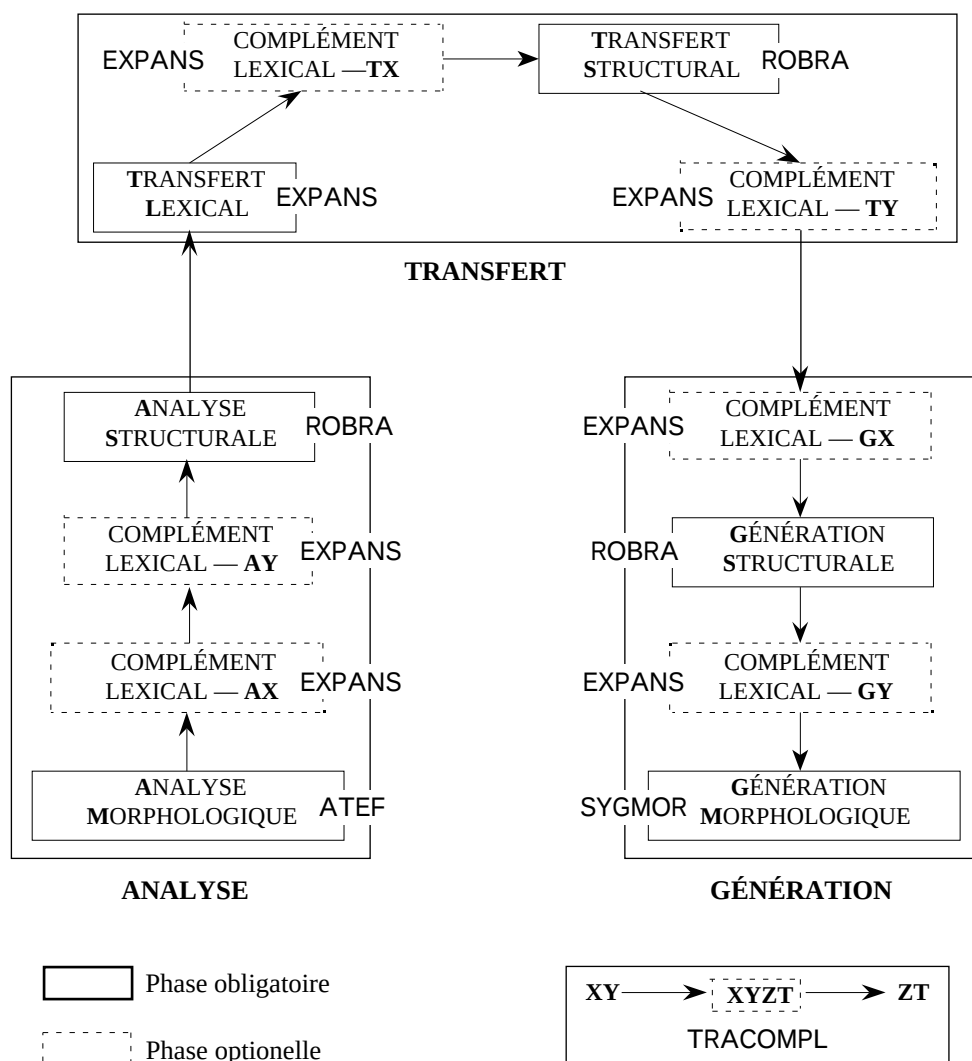


Figure 3.3 : Organisation logique des processus linguistiques en Ariane-G5

L'ordre de ces phases à l'intérieur de chaque étape est imposé. Par conséquent, les "articulations" possibles, toutes écrites en TRACOMPL, sont AMAX, AMAY, AMAS, AXAY, AXAS, AYAS, ASTL, puis TLTX, TLTS, TXTS, TSTY, TSGX, TSGS, TYGX, TYGS, et enfin GXGS, GSGY, GSGM et GYGM. On n'a en fait besoin d'écrire une phase d'articulation que si l'on veut composer deux phases provenant de logiciels utilisant des types de décorations hétérogènes.

c. Structure produite par l'analyseur

L'analyseur utilisé dans le cadre du projet LIDIA produit une structure multirésolution, multiniveau et concrète.

i. Multisolution

La structure produite est dite “multisolution” car, pour chaque phrase, elle contient toutes les analyses vérifiant le modèle syntagmatique, syntaxique et logico-sémantique des grammaires utilisées. Cela est nécessaire pour que l’interaction permette d’obtenir l’analyse qui correspond aux intentions de l’auteur.

ii. Multiniveau

La structure produite est dite “multiniveau” car les nœuds portent des décorations complexes qui représentent divers niveaux d’interprétation : le niveau des classes syntaxiques et des classes syntagmatiques, le niveau des fonctions syntaxiques, et enfin le niveau des relations logiques et sémantiques.

- Le niveau des classes syntaxiques (classes terminales telles que – nom commun : NCOM, – verbe : VB, – préposition : PREP, – démonstratif : DEM, etc.) et des classes syntagmatiques (classes non terminales telles que – groupe nominal : GN, – phrase verbale : PHVB, etc.), donne le parenthésage de la phrase en groupes syntagmatiques. Les feuilles de l’arbre portent les classes syntaxiques et les nœuds non terminaux portent les classes syntagmatiques.
- Le niveau des fonctions syntaxiques donne le rôle syntaxique de chaque nœud dans le groupe auquel il est attaché directement (par exemple – gouverneur : GOV, – régisseur : REG, – sujet : SUJ, – circonstant : CIRC, – complément : COMP, etc.). Avec ce niveau, on peut très simplement transformer l’arborescence en une arborescence de “dépendance syntaxique”.
- Le niveau des relations logiques et sémantiques peut être considéré comme celui de la construction logique de la phrase, mettant en évidence les places des arguments attachés aux unités lexicales prédicatives et donnant l’interprétation sémantique des compléments et circonstants (par exemple – premier argument d’un prédicat : ARG0, – second argument d’un prédicat : ARG1, – circonstanciel instrumental : INST, – déterminant : DET, etc.). Il s’agit de syntaxe profonde et non d’une interprétation logique.

La figure 3.4 montre l’arbre d’analyse de la phrase “l’élève calcule cette intégrale par la méthode des résidus”, tel qu’il est classiquement calculé au GETA²⁶.

iii. Concrète

La structure produite est dite “concrète”, car on retrouve directement le texte analysé (*en italique* dans la figure 3.4) via le mot des feuilles de l’arbre. Cette propriété facilite la construction d’un processus de génération.

²⁶ La géométrie des arbres produits, introduite par B. Vauquois en 1974, correspond avant la lettre à celle qui est obtenue avec des analyseurs fondés sur le formalisme “Head-driven Phrase Structure Grammar” (grammaire syntagmatique guidée par la tête).

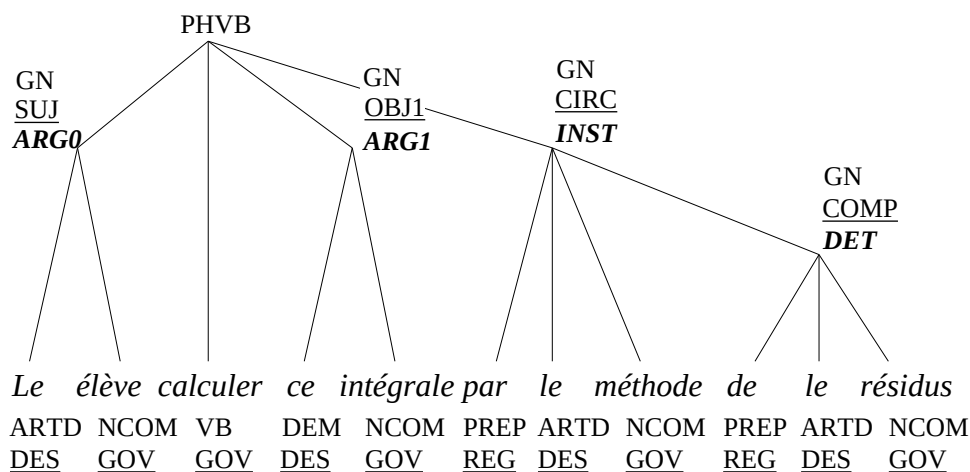


Figure 3.4 : Un arbre d'analyse multiniveau

légende :

CLASSE SYNTAXIQUE et SYNTAGMATIQUE,

FONCTION SYNTAXIQUE,

RELATION LOGIQUE et SÉMANTIQUE

3.2.2. Interface et dialogue de désambiguïsation

Comme un système de TAFD doit s'intégrer de manière transparente dans l'environnement hôte, on accède aux traitements LIDIA avec des menus et des palettes d'outils. Les outils LIDIA se comportent comme les outils HyperCard. La communication des informations vers l'utilisateur se fait au moyen de dialogues et d'icônes.

Pour le dialogue de désambiguïsation, nous avons choisi d'exploiter une seule modalité. Nous utilisons des dialogues qui présentent des choix à l'utilisateur, ces choix étant libellés en langue naturelle.

Nous avons aussi choisi de faire appel à des mécanismes simples pour produire ces dialogues. En d'autres termes, nous ne ferons pas appel à des processus linguistiques complexes tels que la passivation ou la négation des énoncés.

Nous utilisons des définitions extraites de la base lexicale par acception PARAS [Sérasset & Blanc 1993]. Nous produisons des paraphrasages qui se traduisent par la permutation des occurrences des phrases ambiguës ou par le remplacement d'occurrences ambiguës. Le seul outil "linguistique" que nous utilisons est un générateur de formes fléchies d'adjectifs.

3.2.3. Organisation des traitements linguistiques²⁷

Expliquons brièvement les traitements principaux illustrés dans la figure 3.5.

²⁷ Le paragraphe 3.2.3. est une reprise en français de [Boitet & Blanchon 1993a] et publiée dans [Boitet 1993].

1. Après l'étape de standardisation,

- les textes doivent être orthographiés correctement, et être conformes aux paramètres stylistiques associés au type de leur conteneur (nous intégrons le correcteur 101 de Machina Sapiens) ;
- les préférences terminologiques (cf. supra) doivent avoir été plus ou moins imposées par le *module de préférence lexicale*, selon le degré de normalisation terminologique souhaité ;
- les syntagmes figés se comportant de façon spéciale (comme l'item de menu Cacher les bulles d'aide dans "Cacher les bulles d'aide arrête l'aide par bulles") doivent être marqués (l'utilisateur doit aider le *module de syntagmes figés spéciaux* à établir la liste des syntagmes figés spéciaux de la pile) ;
- tous les champs et boutons doivent porter un type de texte²⁸, permettant de contrôler le correcteur stylistique, puis de guider l'analyse (d'où un module de *catégorisation textuelle*).

2. Le texte standardisé est alors analysé sur le serveur. La *mmc-structure source* produite (multisolution, multiniveau et concrète) est transformée en une forme portable (en lisp) et lisible (directement par les développeurs) et envoyée au Macintosh.

3. La mmc-structure source est utilisée pour produire le dialogue de désambiguïsation sur le Macintosh. Le processus de désambiguïsation la transforme en une *umc-structure source* non-ambiguë (unisolution, multiniveau et concrète) correspondant à l'analyse choisie par l'auteur²⁹.

4. Cette umc-structure source est alors "abstraite", ou "réduite" à une *uma-structure source* (unisolution, multiniveau et abstraite).

5. À partir de la uma-structure source, le système Ariane-G5 produit les *gma-structures cibles* (génératives, multiniveaux et abstraites), en utilisant les transferts adéquats. Une gma-structure est plus "générale" et plus "généralisante" qu'une uma-structure, car ses niveaux de surface (fonctions syntaxiques, catégories syntagmatiques...) peuvent être vides, et sinon ne sont que des préférences indiquées par le transfert.

6. Pour chaque langue cible, la génération structurale produit à partir de la gma-structure cible une *uma-structure cible* homogène avec ce que serait le résultat de l'analyse (et de la désambiguïsation) du texte cible qui sera généré. Cette étape consiste à choisir la paraphrase à générer en calculant les niveaux de surface³⁰ et à choisir une première approximation de l'ordre des mots à partir des niveaux plus profonds (relations logiques et sémantiques, traits sémantiques, etc.).

²⁸ Dans le cas de textes "incomplets", par exemple si le sujet de la première phrase est contenu dans un autre champ (comme dans des tables contenant des noms de commandes et leurs explications), ce module demande aussi comment construire le texte complet.

²⁹ "Concrète" signifie que le texte original peut être retrouvé à partir de la structure de façon directe (par un parcours préfixe des feuilles pour des structures de constituants et par un parcours infixé de tous les nœuds pour des structures de dépendances). Les nœuds et/ou les arcs de la structure peuvent contenir aussi bien de l'information "de surface" que de l'information "profonde" (organisation en prédicats/arguments, relations sémantiques...). Dans les structures "abstraites", les négations, les auxiliaires, les articles, etc., peuvent avoir été supprimés en tant que nœuds et être représentés dans les décorations, certains éléments éliminés peuvent avoir été insérés, l'ordre peut avoir été normalisé, etc.

³⁰ En particulier, les fonctions syntaxiques, les classes syntagmatiques, et les classes morphosyntaxiques, ces dernières en fonction des schémas dérivationnels des unités lexicales.

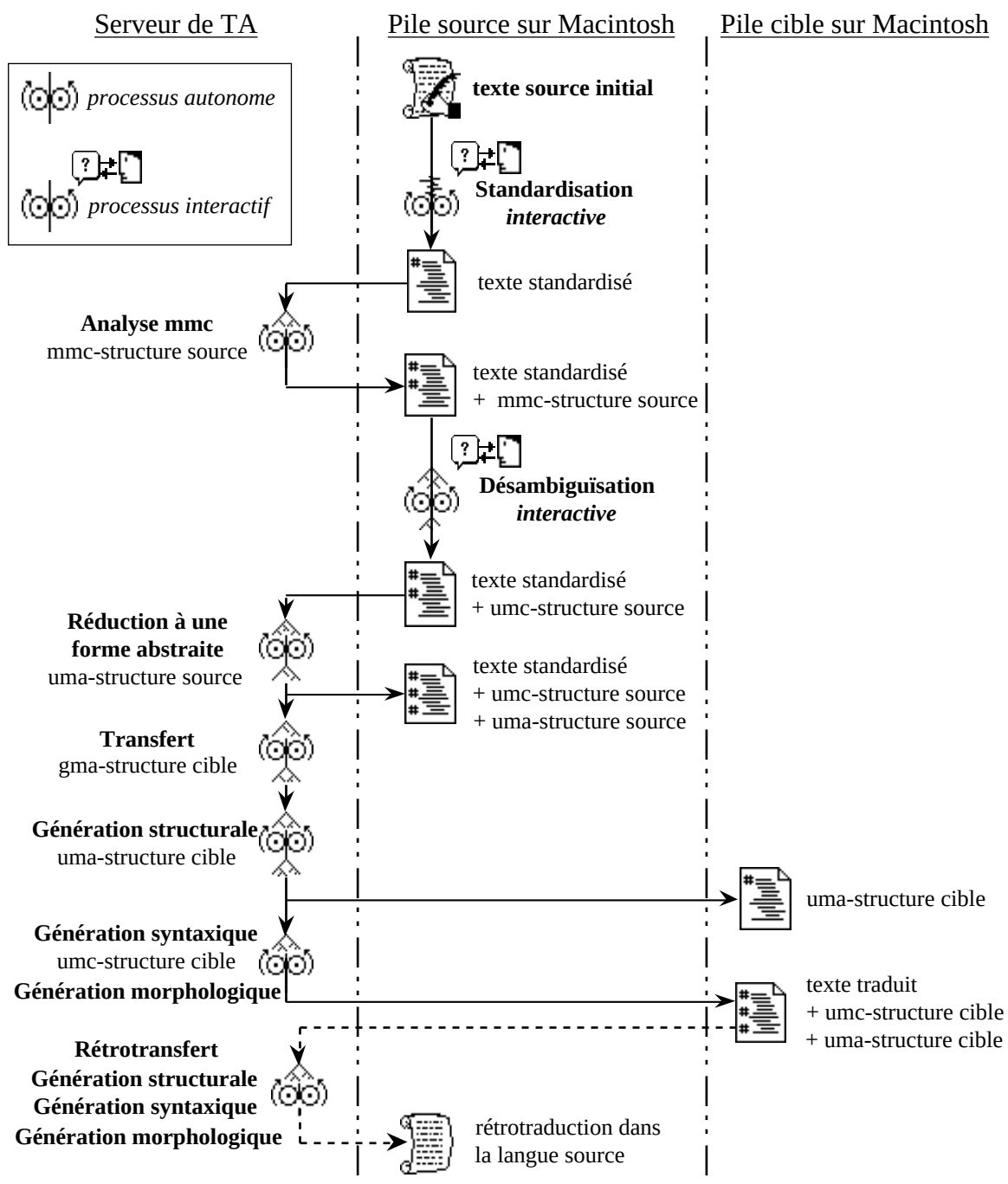


Figure 3.5 : Organisation générale du processus de traduction en LIDIA-1

7. Le processus de traduction se termine par les générations syntaxiques et morphologiques. Quand tous les objets ont été traduits, on obtient la ou les piles images dans la ou les langues cibles.

8. Les uma-structures cibles peuvent être utilisées comme point de départ de *rétrotraductions* permettant à l'auteur (monolingue) de contrôler les traductions.

L'idée est que le système, traduisant par exemple de français en russe, retradise au rédacteur ce qu'il va produire en russe, lui permettant ainsi de contrôler le résultat sans connaître un mot de russe. C'est d'ailleurs souvent ce qui se passe entre un interprète et son client.

Comme on suppose que la génération ne pose pas de problèmes, et qu'on veut éviter d'avoir à écrire tous les analyseurs dans une situation monosource et multicible, on ne repart pas du texte final en russe, mais de son uma-structure produite par la génération structurale. La rétrotraduction n'a pas à être identique à l'original pour que la traduction soit bonne, tout comme en traduction et surtout en interprétation humaine : on pourrait par exemple conserver actif et passif dans un sens (F-R), et les échanger dans l'autre (R-F) : il n'y aurait même pas de point fixe.

3.3. État d'avancement

Au moment de la rédaction de ces lignes, la maquette était en cours d'achèvement et devait donner lieu à une démonstration deux mois plus tard. Ce qui suit est donc une projection.

3.3.1. Supports informatiques

Du point de vue informatique, tout ce qui était prévu devrait être disponible, à savoir,

- le serveur LIDIA (sur le Macintosh),
- le serveur de communication (sur le Macintosh),
- le serveur de traduction (sur l'IBM).

Au moment de la rédaction, il restait à tester les mécanismes de communication physique entre le réseau physique des Macintoshs et le réseau virtuel des “machines virtuelles serveur” Ariane-G5 sur l'IBM.

3.3.2. Linguiciels de TA pure

Du point de vue linguiciel, la maquette devrait être complète, du français vers l'anglais, l'allemand et le russe, avec les trois rétrotraductions. Bien qu'il s'agisse d'une petite maquette, les dictionnaires et grammaires ne sont absolument pas ad hoc, et cherchent au contraire à refléter, pour les termes et les constructions retenues, la complexité réelle que rencontrerait un véritable système.

3.3.3. Standardisation et désambiguïsation

Le travail de N. Denos n'ayant pu être repris à temps, le typage textuel est très rudimentaire. Le traitement des syntagmes figés est également embryonnaire. Par contre, le reste de la standardisation est assez complet. La désambiguïsation fait l'objet d'un travail particulièrement important, même si, nous le redirons plus loin, nous n'avons pas traité tous les types d'ambiguïté.

Seconde Partie

La maquette LIDIA-1 – vue externe

Introduction

Pour construire notre maquette, nous avons supposé que le rédacteur construisait une pile sous HyperCard, de façon “traduisible”, et décidait à partir d’un certain moment d’activer LIDIA.

Son environnement initial ne diffère d’HyperCard que par la présence d’un nouveau menu LIDIA. Par contre, une fois que LIDIA est activé, le rédacteur a accès à des outils supplémentaires (actions LIDIA, palette, témoins liés aux champs textuels et aux boutons à traduire).

L’interface n’est pas modale : chaque objet indique (éventuellement) qu’il est en attente de réponses de l’utilisateur à des questions provoquées par une certaine étape du traitement.

La première grande étape concerne la standardisation, avec quatre phases, la vérification orthographique et grammaticale, la vérification terminologique, le marquage de syntagmes spéciaux et le typage textuel.

La seconde grande étape concerne la clarification, c’est-à-dire la désambiguïsation à différents niveaux. À ce sujet, nous proposons une classification exemplifiée des ambiguïtés qu’il nous a semblé important de traiter au niveau d’une maquette. Pour démontrer la maquette, nous avons construit une pile traduisible qui permet de traiter le corpus que nous avons choisi.

Chapitre 4

Utilisateur et utilisation

L'accès aux traitements LIDIA se fait via HyperCard dans un environnement logiciel particulier. Le contrôle des traitements de traduction est pris en charge par le *serveur LIDIA* qui est hébergé par la station de travail du rédacteur³¹. Pour mener la traduction à terme, le serveur LIDIA utilise un serveur de communication qui permet de gérer les échanges de requêtes avec un serveur de traduction.

Les traitements que l'on peut effectuer sur un objet avant d'obtenir sa traduction sont multiples et facultatifs. Les outils pour les réaliser peuvent être choisis par l'utilisateur, qui peut ainsi définir :

- le profil de la tâche, en choisissant, parmi les différents traitements proposés, ceux qui seront effectivement réalisés lors d'un traitement standard ;
- le profil des ressources lexicales, en choisissant un correcteur orthographique et des dictionnaires et thesaurus personnels ;
- le profil des serveurs, en choisissant les serveurs de traduction et de communication ;
- le profil de l'utilisateur, en choisissant des modes d'interaction avec le système.

Les traitements LIDIA sont mis en œuvre en utilisant les objets d'interface exploités dans HyperCard, à savoir des menus, des palettes flottantes et des curseurs outils. La rétroaction est assurée au moyen de messages et de zones actives sur l'écran, comme habituellement sous HyperCard.

4.1. Environnement de travail

L'environnement de travail définit la configuration logicielle de la station de travail, l'accès au système et la gestion des piles.

³¹ La délocalisation du serveur LIDIA n'a pas été envisagée dans ce travail. Nous verrons, lorsque nous décrirons l'implémentation, que cette délocalisation ne pose aucun problème.

4.1.1. Configuration logicielle de la station de travail

L'environnement de travail LIDIA comporte plusieurs outils plus ou moins transparents pour l'utilisateur.

Le premier de ces outils est bien sûr HyperCard, qui permet à l'utilisateur de rédiger son hypertexte et d'accéder aux fonctionnalités LIDIA.

Le second outil indispensable est le serveur LIDIA, implémenté en Macintosh Common Lisp. Le contrôle du processus de traduction avec la standardisation et la clarification est pris en charge par cet environnement. Cet outil est transparent pour l'utilisateur.

Si l'utilisateur le souhaite, il peut aussi accéder à Parax, qui est une maquette de base de données lexicales multilingues, indépendante de toute théorie linguistique et indépendante de toute application (cf. § 7.2.).

La station de rédaction doit communiquer avec le serveur de traduction. Si la station de rédaction n'est pas équipée d'une carte de communication, le serveur de communication est hébergé par une autre machine sur le réseau des stations de travail. Si la station de rédaction est dotée d'une carte de communication, l'utilisateur peut choisir d'héberger le serveur de communication ou de le laisser tourner sur une autre machine s'il est disponible ailleurs. Le serveur de communication est également transparent pour l'utilisateur.

4.1.2. Accès au système

Lorsque l'auteur lance HyperCard, il choisit son niveau d'utilisation dans la carte Préférences de la pile "Home". Sur cette même carte, il peut aussi choisir d'Utiliser LIDIA. Lorsque cette fonctionnalité est activée, le menu LIDIA apparaît dans la barre de menus.

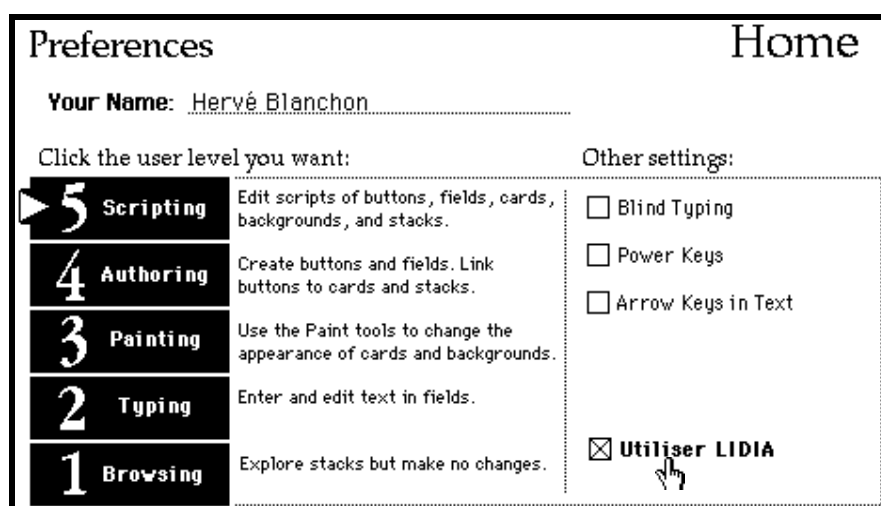


Figure 4.1 : Bouton d'accès aux traitements LIDIA

Si l'utilisateur travaille sur une nouvelle pile, ou sur une pile pour laquelle il n'a pas encore défini les préférences LIDIA, ce menu lui permet de les définir (cf. infra).



Figure 4.2 : Menu LIDIA minimum

Si l'utilisateur travaille sur une pile pour laquelle il a déjà défini les préférences LIDIA, le menu correspond aux paramètres de traitement de la pile.



Figure 4.3 : Menu LIDIA après définition de préférences

À chaque menu est associée une palette flottante qui permet d'accéder à tous les traitements.

4.1.3. Piles

Une pile est un document HyperCard qui possède certaines des caractéristiques d'un hypertexte .

a. Vocabulaire

Trois caractéristiques principales distinguent un document HyperCard d'un document textuel classique. Un document HyperCard peut contenir des sons, des images et des films, et peut être consulté de manière non linéaire (grâce à des liens hypertextuels) selon divers scénarios d'interaction entre l'utilisateur et le document.

Les hypertextes [Nielsen 1990] ont donc un pouvoir explicatif et descriptif augmenté par rapport à des documents textuels habituels. Ils ouvrent de nouvelles perspectives pour la présentation des documentations techniques, des modes d'emploi, des catalogues, etc.

Un document HyperCard se présente comme un ensemble de cartes supportant des informations. Cet ensemble est appelé une pile. Les cartes sont reliées entre elles par des liens hypertextuels, selon un graphe de relations. Ces liens sont matérialisés sur chaque carte par des "boutons" ou autres zones de la carte, qui permettent d'accéder immédiatement à une autre carte.

Il existe 5 sortes d'objets HyperCard.

Pile : c'est le document HyperCard, il contient un ensemble de fonds et de cartes.

Fond : une pile peut comporter plusieurs fonds. Un fond peut être partagé par plusieurs cartes (il contient donc des éléments communs à plusieurs cartes).

Carte : elle est superposée à un fond (elle a la même taille que le fond). On voit les fonds par transparence à travers la carte.

Ce que l'utilisateur voit sur l'écran est la superposition d'une carte sur un fond. Ainsi, toutes les cartes du bestiaire figure 4.4 sont conçues selon le même modèle, et partageant le même fond.

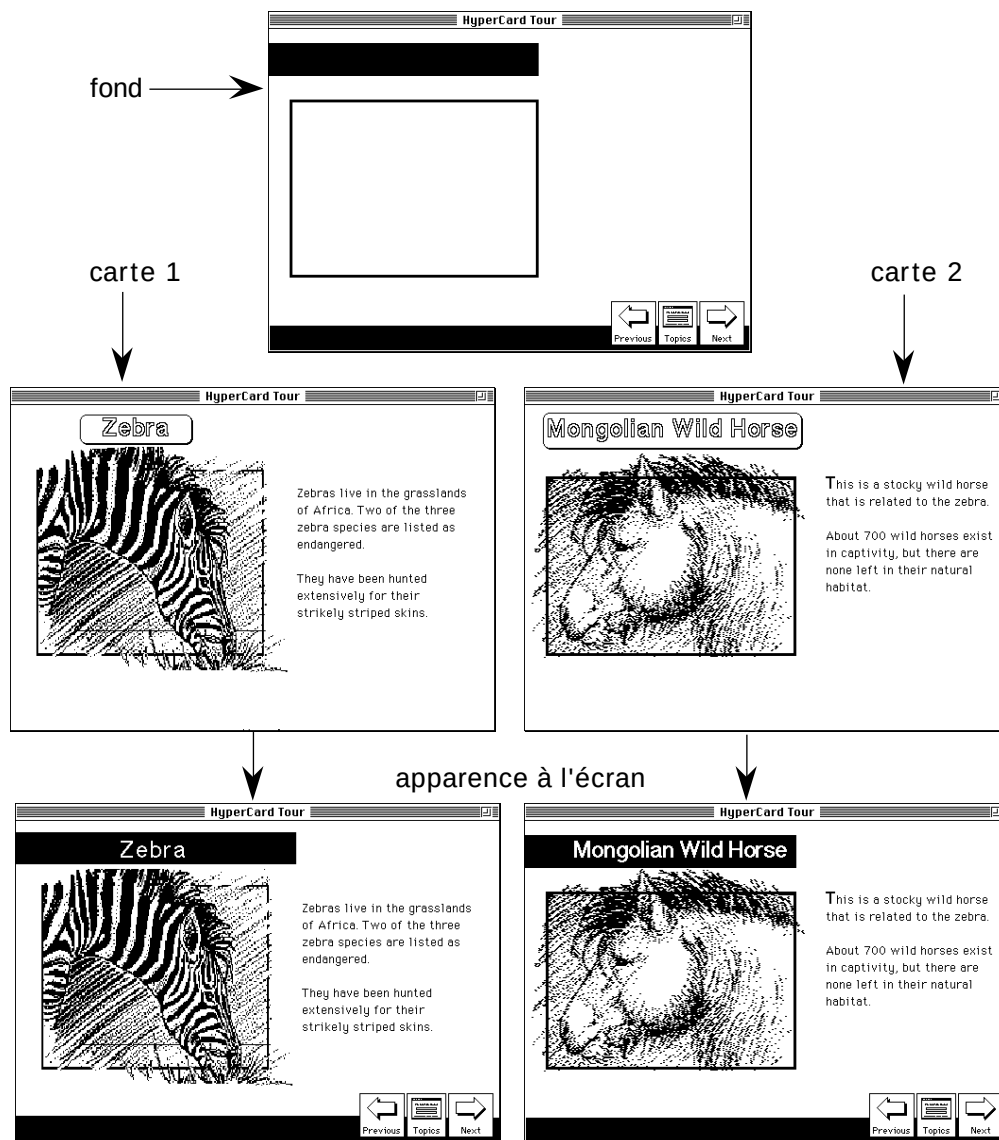


Figure 4.4 : Schéma de construction d'une pile HyperCard

Bouton : c'est une zone "active" de la carte contenant un texte non éditable directement (le nom du bouton peut être édité en passant par le menu). Un bouton peut appartenir à un fond ou à une carte.

Champ de texte : c'est une zone contenant un texte éditable (sauf s'il est verrouillé). Il peut appartenir à un fond ou une carte. Un champ de texte vide peut appartenir à un fond, son contenu étant spécifique à une carte.

On peut inclure des dessins sur les cartes et sur les fonds. Le schéma suivant illustre la combinaison des objets HyperCard telle que la voit l'utilisateur.

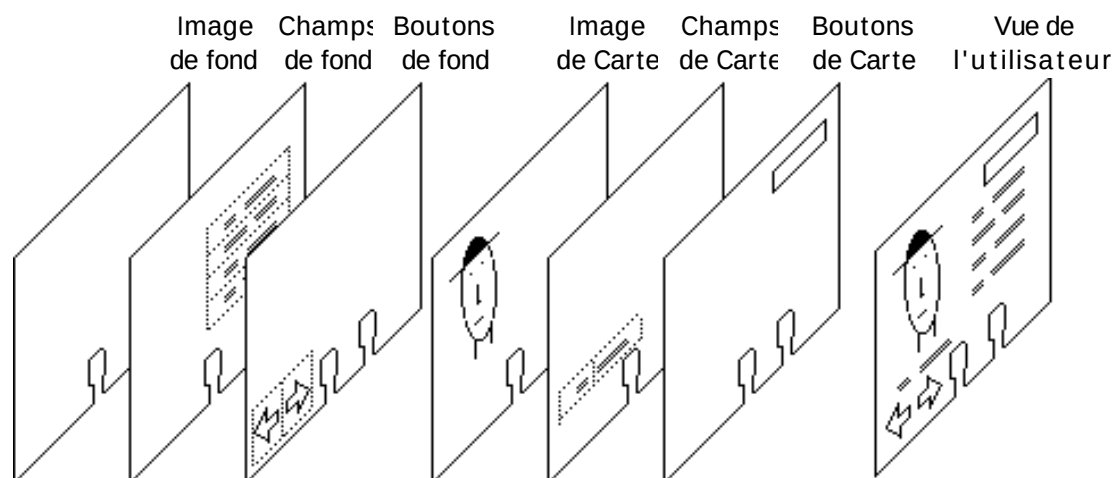


Figure 4.5 : Les objets HyperCard

b. Conditions de traductibilité

Pour être exploitable complètement dans une langue différente de la langue de conception, une pile doit vérifier certaines contraintes.

D'abord, il est clair que nous ne pouvons traduire que le nom des boutons et le contenu des champs textuels, qui sont les seuls objets auxquels on a vraiment accès. L'utilisateur peut "dessiner" du texte sur ses cartes, mais celui-ci ne peut être traduit car il est codé dans la représentation du dessin de la pile sous la forme d'un ensemble de pixels allumés ou éteints.

Il est fréquent qu'un document hypertexte émette des messages en réponse aux actions de l'utilisateur. Si ces messages sont codés dans les scripts, on ne peut en général automatiser leur traduction.

Par exemple, la commande HyperTalk :

answer "Vous souhaitez des explications : "with "simples" or "détaillées" produit le dialogue suivant :

Pour qu'un tel dialogue puisse être traduit, il faudrait qu'il soit produit par la commande suivante :

answer card field x of card t with card field y of card t or card field z of card t, où card field x of card t contient "Vous souhaitez des explications :", card field y of card t contient "simple", et card field z of card t, "Détaillées".



Il est possible que des objets (boutons, champs) apparaissent ou disparaissent. Il faudra prévoir un scénario de traitement pour ces objets. Dans ce cas, l'une des solutions que l'on peut envisager est que le serveur LIDIA les fasse réapparaître temporairement

pour les besoins de la traduction. Quoiqu'il en soit, la gestion de cette situation est complexe et mérite d'être étudiée avec attention.

Finalement, comme le nom des boutons peut être traduit, il est recommandé de ne pas référencer, dans les scripts, les boutons par leur nom mais par l'identificateur unique qu'HyperCard associe à chaque objet de la pile.

c. Localisation physique

Les piles traduites sont produites sur la station de travail du rédacteur. La pile complète est construite lorsque tous les objets de la pile source ont été traduits. L'utilisateur peut, lorsqu'il le désire, demander la construction de la pile traduite pour voir où en est le système. Avec HyperCard, la construction incrémentale de la pile traduite est impossible. Lorsqu'un nouvel objet est créé dans la pile source, il faut recommencer complètement la construction de la pile cible.

4.2. Personnalisation du système

Pour personnaliser les traitements à effectuer, le système permet d'associer un fichier de préférences LIDIA à chaque pile à traduire. Ce fichier est rangé dans un dossier prédéterminé.

Un dossier LIDIA est créé sur la station de rédaction. Ce dossier comporte six sous-dossiers :

- Boîte aux lettres contient les fichiers arrivés depuis le serveur de communication, que celui-ci soit hébergé ou non par la station de rédaction ;
- Poste contient les fichiers en partance vers le serveur de communication ;
- Préférences contient les fichiers préférences de toutes les piles en cours de traduction avec LIDIA ;
- Serveur de communication contient les fichiers en partance vers le serveur de traduction ;
- Serveur de communication OU contient les fichiers en retour du serveur de traduction.



Figure 4.6 : Organisation des dossiers LIDIA

- Pour le typage contient les fichiers de définition et d'exemples des types d'énoncés et des genres de textes.

Sur les stations qui servent uniquement de serveur de communication, on retrouve seulement les dossiers relatifs au serveur de communication — inclus dans un dossier LIDIA. Sur une station de rédaction utilisant un serveur de communication, ces deux derniers dossiers sont vides, le serveur de communication assurant le stockage des fichiers.

Lorsque l'utilisateur a choisi d'utiliser LIDIA, il peut définir ou modifier un fichier de préférences pour chaque pile qu'il crée ou manipule.

Lorsqu'il choisit l'item "préférences..." dans le menu LIDIA, un dialogue apparaît. Dans la fenêtre du dialogue peuvent s'inscrire quatre panneaux qui permettent de régler quatre profils de paramètres : le profil de la tâche, le profil des préférences lexicales, le profil des serveurs et le profil de l'utilisateur. L'utilisateur passe de panneau en panneau à l'aide d'un menu surgissant que l'on voit ci-dessous, inactif — à gauche — et activé avec un clic tenu — à droite.

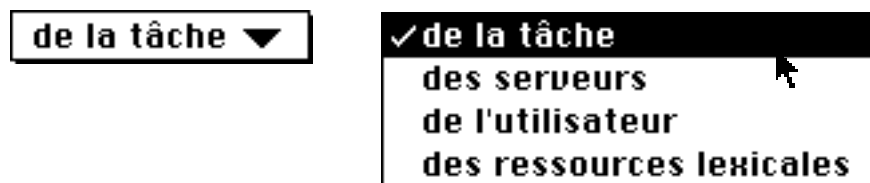


Figure 4.7 : Menu surgissant pour choisir un panneau de profil.

4.2.1. Profil de la tâche

Avec le profil de la tâche, on détermine :

- les langues cibles vers lesquelles on souhaite traduire,
- l'unité de traduction,
- les différentes étapes du traitement.

La maquette permet de traduire du français vers l'anglais, l'allemand et le russe simultanément. S'il le souhaite, l'utilisateur peut bien sûr choisir n'importe quel sous-ensemble de ces trois langues.

La structure d'un document HyperCard impose trois unités de traduction, les objets de base de l'hypertexte — les boutons et les champs —, les cartes, et l'hypertexte complet — la pile.

On peut aussi envisager comme dans CRITIQUE [Heidorn, *et al.* 1982 ; Ravin 1988] un mode de fonctionnement Automatique dans lequel le système traite tous les objets qui n'ont pas subi de modification depuis un certain temps. Cette dernière option a été discutée au sein de l'équipe mais n'a pas été retenue pour la maquette car elle exige des ressources logicielles supplémentaires. D'autre part, nous ne sommes pas convaincus que dans la situation traductionnelle que nous envisageons, le système doive effectuer des traitements qui peuvent nécessiter l'intervention de l'utilisateur sans que celui-ci les ait explicitement demandés. Ce dernier scénario pourra être étudié avec un simulateur afin de vérifier si l'investissement que nécessite sa mise en œuvre soit utile.

Comme nous l'avons déjà dit (§ 3.2.3.), nous envisageons d'appliquer le processus de traduction proprement dit sur un texte standardisé, c'est-à-dire un texte "plus facile à traduire". C'est pourquoi nous proposons plusieurs autres traitements en plus de la

traduction proprement dite : l’orthographe et/ou la grammaire et/ou le style, la terminologie, les formules figées, et le typage textuel³².

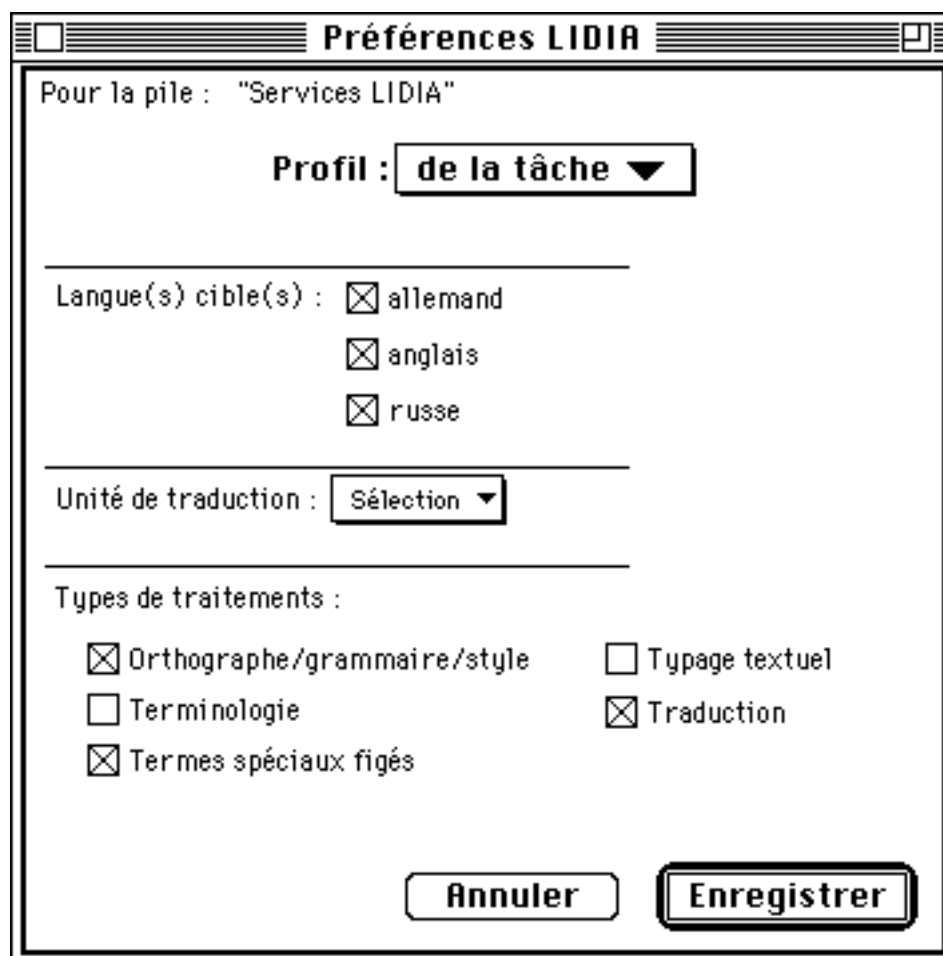


Figure 4.8 : Préférences LIDIA “Profil de la tâche”

Le panneau “profil de la tâche” (figure 4.8) ainsi que les trois autres (figures 4.9, 4.12, 4.14), est présenté vierge à l’utilisateur si c’est la première fois que celui-ci règle les préférences d’une pile. Si l’utilisateur modifie des préférences³³, les panneaux apparaissent avec les anciennes valeurs.

En remplissant le panneau précédent, l’utilisateur a choisi de traduire la pile Services LIDIA vers l’anglais, l’allemand et le russe. Il veut traduire des objets choisis au fur et à mesure. Parmi les traitements de standardisation, il a choisi la vérification orthographique et le traitement des termes spéciaux figés.

³² Ces traitements sont détaillés dans le chapitre 5 consacré à la standardisation.

³³ Les préférences peuvent être changées en cours de traitement. Lorsque l’on sélectionne un objet de la pile, le fichier de préférences est consulté pour créer “un plan de vol” qui est associé à l’objet qui en franchit les différentes étapes. Différentes familles d’objets peuvent avoir des plans de vol différents.

4.2.2. Profil des ressources lexicales

Le panneau “profil des ressources lexicales” (figure 4.9) permet de choisir :

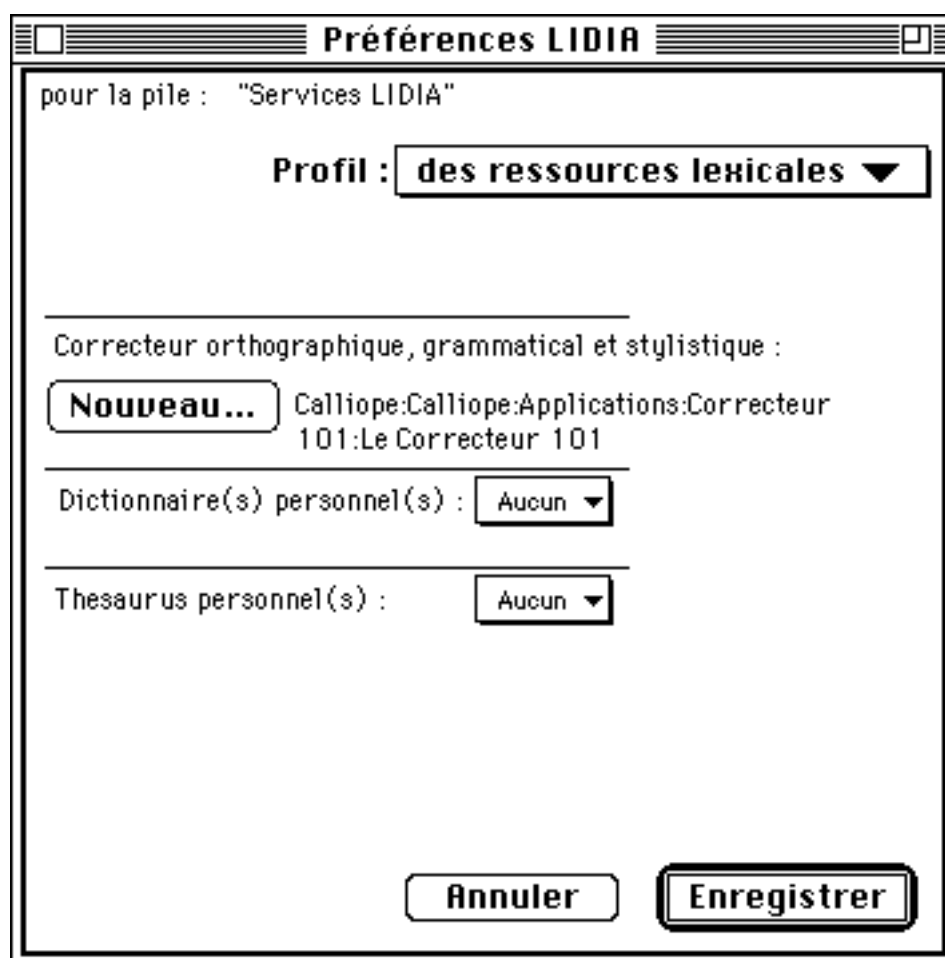


Figure 4.9 : Préférences LIDIA “Profil des ressources lexicales”

- le correcteur d’orthographe et/ou de grammaire et/ou de style.
- les dictionnaires de l’utilisateur³⁴.
- les thesaurus de l’utilisateur³⁵.

Ici, l’utilisateur a choisi le correcteur orthographique “Le Correcteur 101” de la société canadienne Machina Sapiens, dont on voit le chemin d’accès, avec un dialogue standard de sélection de document (figure 4.10). Il n’a pas défini de dictionnaires ni de thesaurus personnels.

³⁴ Ces dictionnaires pourront être utilisés par le correcteur orthographique, pour mettre à jour la base lexicale par acceptions et les dictionnaires utilisés lors du processus de traduction. On songe bien sûr aux mots inconnus que va inévitablement rencontrer le système. Nous reviendrons sur ce problème dans le paragraphe 5.2.

³⁵ Ces thesaurus devraient être utilisés, lors de la phase de vérification terminologique, comme références pour accepter ou refuser les termes utilisés par le rédacteur (cf § 5.4.).

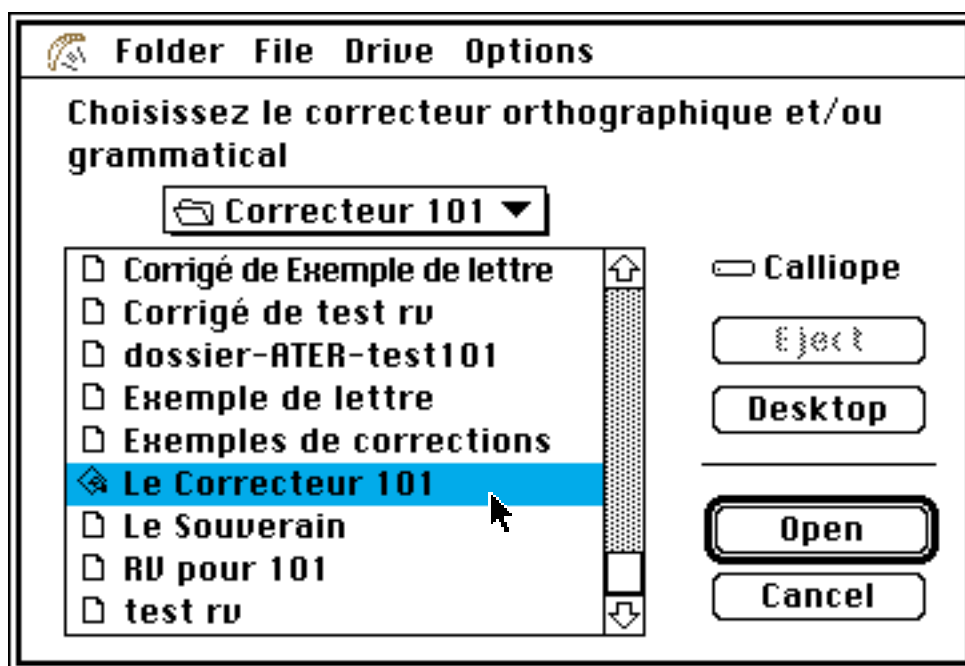


Figure 4.10 : Préférences LIDIA : sélection du correcteur orthographique

4.2.3. Profil des serveurs

Le panneau “profil des serveurs” (figure 4.12) permet de choisir le serveur de communication et le serveur de traduction.

Pour l’instant, un seul serveur de traduction est disponible. La sélection est vue comme le choix d’un item dans une liste à n éléments. L’interface d’accès est présentée figure 4.11.

Le serveur de communication est sélectionné par un dialogue de sélection complexe (figure 4.13). L’utilisateur doit d’abord choisir une zone du réseau (en bas à gauche), puis une machine physique sur ce réseau (en haut à gauche), et enfin le programme qui fait fonction de serveur de communication. Ici, l’utilisateur a choisi le programme RemoteMacMainFrame sur la machine Serveur Avatar dans la zone GETA. Ainsi, le logiciel de la station de rédaction sait avec quel programme communiquer. Le programme serveur de communication doit connaître le nom des machines qui lui demandent des services.

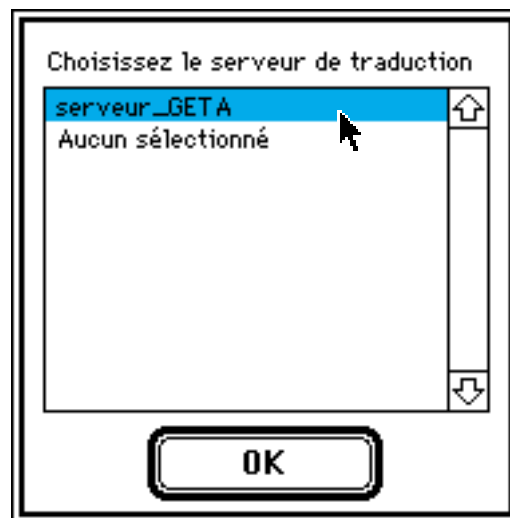


Figure 4.11 : Sélection du serveur de traduction

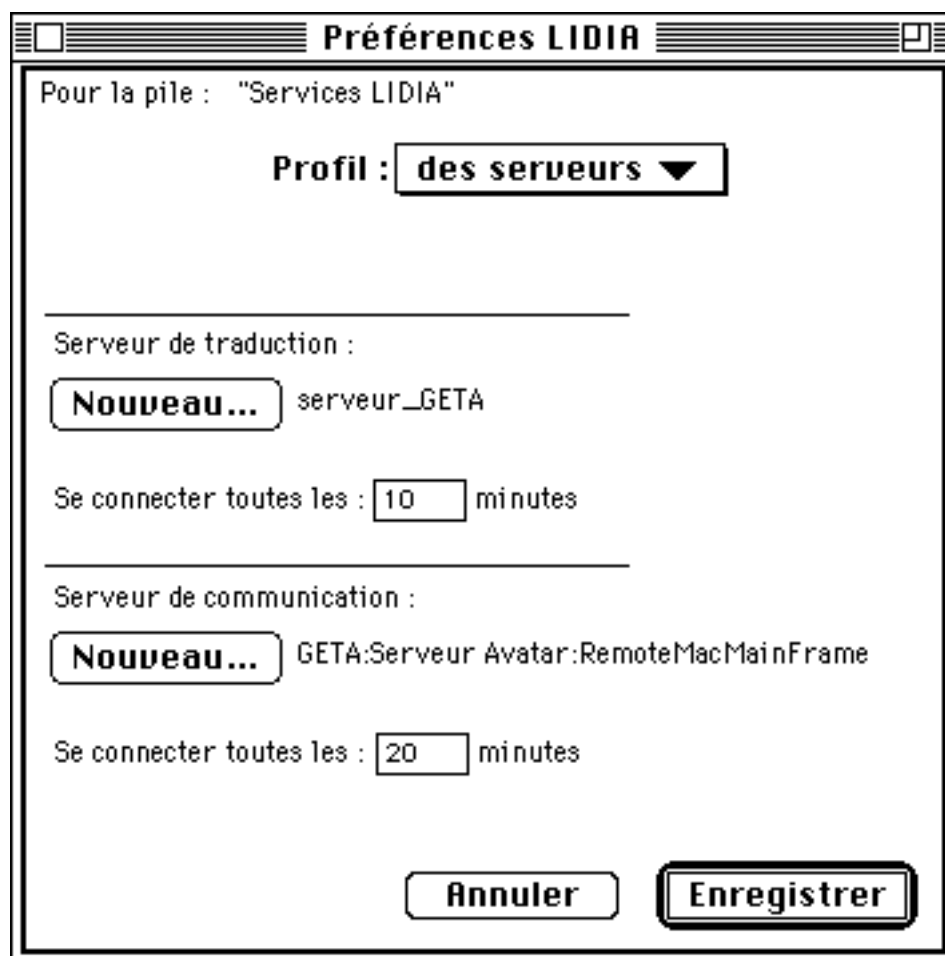


Figure 4.12 : Préférences LIDIA "Profil des serveurs"

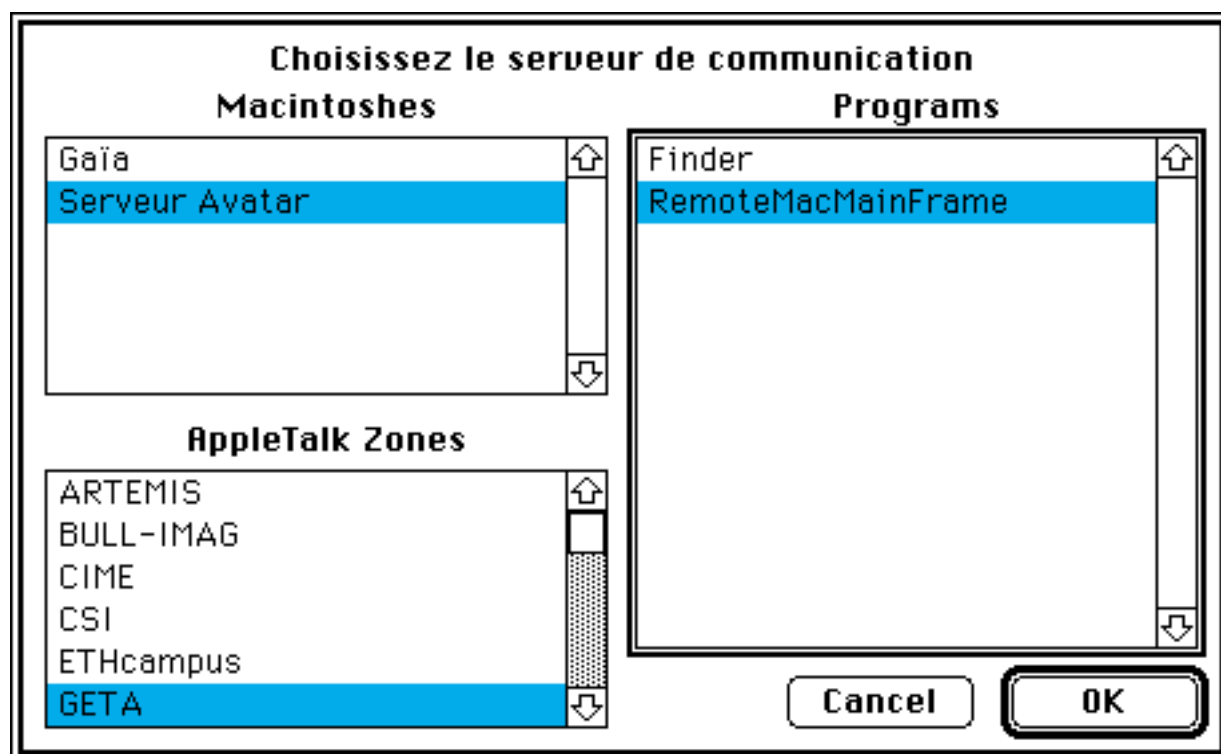


Figure 4.13 : Préférences LIDIA : sélection du serveur de communication

4.2.4. Profil de l'utilisateur

Le quatrième panneau, “profil de l'utilisateur” (figure 4.14), permet de choisir le type du dialogue de désambiguïsation et le type du document (dépendant de l'institution), et de définir comment le système va rétroagir avec l'utilisateur.

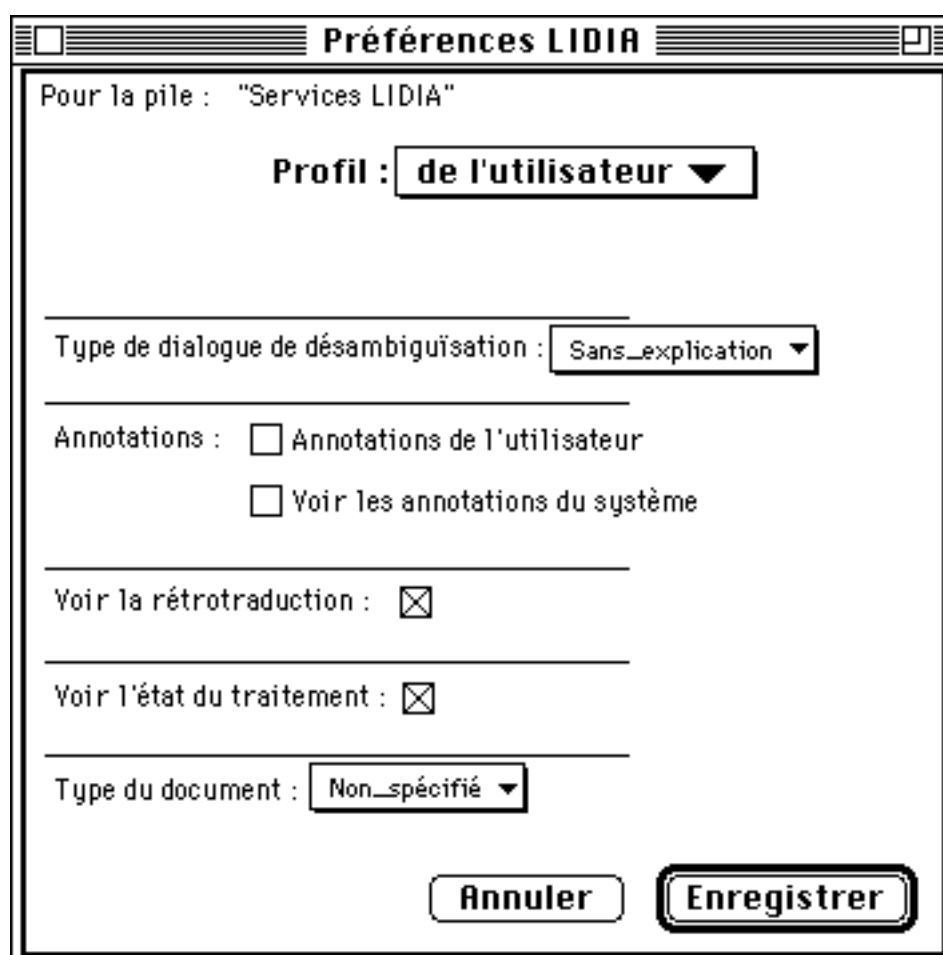
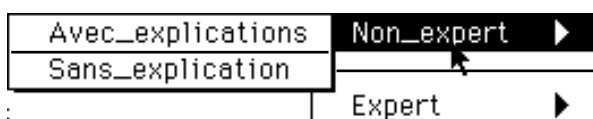


Figure 4.14 : Préférences LIDIA “profil de l'utilisateur”

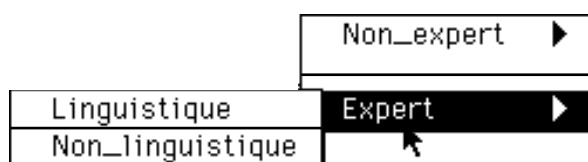
a. Type de dialogue de désambiguïsation

Nous avons imaginé quatre façons de libeller le dialogue de désambiguïsation, deux pour le Non_expert et deux pour l'Expert. Dans la maquette, seul le dialogue sans_explication pour le Non_expert est produit.

Pour un utilisateur non expert du système, nous proposons deux types de dialogue plus ou moins détaillés. Les dialogues Avec_explications sont plus complets et permettent d'accéder à des informations sur le type d'ambiguïté que le dialogue cherche à résoudre, avec des exemples. Les dialogues Sans_explication sont plus compacts et ne permettent pas d'accéder à l'aide en ligne.



Nous distinguons deux classes d'utilisateurs experts : les linguistes et les autres. L'utilisateur expert non linguiste a l'habitude de travailler avec le système. Il a commencé à le découvrir en utilisant le dialogue avec explication. Il a envie d'en savoir plus. On lui propose donc des dialogues où l'on insère des informations linguistiques calculées par le système. Bien sûr, ce n'est pas sur la base de ces informations qu'il peut réellement choisir l'item du dialogue qui correspond à ce qu'il veut dire, mais il se familiarise avec le système. Cela lui permettra de pouvoir insérer des annotations aux endroits pertinents dans le texte qu'il rédige afin de mieux aider l'analyse. À l'utilisateur linguiste, on présentera sous une forme lisible les arbres produits par l'analyseur. Ce niveau peut aider à la mise au point du système.



b. Annotations du système³⁶

Pour favoriser la découvrabilité du système, il nous semble important de montrer à l'utilisateur, s'il le désire, certaines des informations calculées par l'analyseur. Nous songeons bien entendu à des informations qui sont à la portée d'un utilisateur non linguiste.

Il pourrait y avoir des formes annotées plus ou moins complexes et complètes selon le type des informations retenues pour la présentation.

Pour l'utilisateur, la forme annotée du texte suivant :

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.

sera, selon les informations que nous avons retenues pour la maquette

```
{ { 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 étapes.&n,comp { (.&lp
analyse.&n,app ,.&ponc transfert.&n,coord et.&cjcoord
génération.&n,coord ).&rp } } } ..&ponc } { { Chaque.&art étape.&n,suj
} est.&v,aux constituée.&v,phvb { d'&prep une.&art suite.&n,comp {
de.&prep { différentes.&adj,epit } phases.&n,comp { de.&prep
traitement.&n,comp } } } ..&ponc } { { Chaque.&art phase.&n,suj }
est.&v,phvb { relative.&adj,atsubj { à.&prep l'&art emploi.&n,obj1 {
d'&prep un.&art LSPL.&np,comp { précis.&adj,epit } } } } ..&ponc }
```

Cette forme annotée est obtenue par projection systématique de certaines informations contenues dans la structure umc obtenue après désambiguïsation (n'apparaissent pas ici le sens de chaque occurrence).

Le point principal est que le système d'annotations présentées à l'utilisateur concerne plusieurs niveaux de description linguistique, mais est incomplet à chaque niveau, parce qu'aucune notion non familière ne doit apparaître. Par exemple, "verbe" est une notion familière

³⁶ extrait de [Boitet 1993]

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.

Comme l'utilisateur pourra voir les annotations, il pourra aussi fournir des annotations au système. Si les annotations sont suffisamment pertinentes, elles favoriseront le travail de l'analyste.

Si l'utilisateur coche la case Voir les annotations du système, chaque objet pour lequel les annotations sont prêtes signalera à l'utilisateur la disponibilité de ces informations. Si cette option n'est pas sélectionnée, l'utilisateur peut voir les annotations en les demandant explicitement.

c. Annotations de l'utilisateur

Nous formulons l'hypothèse qu'un utilisateur ayant une expérience suffisante du système et plus particulièrement des ambiguïtés reconnues pourrait assister l'analyste en lui donnant des informations ponctuelles de même forme que celles produites par le système.

Ces informations pourraient concerner :

- la classe syntaxique des occurrences,
- le sens des occurrences,
- le parenthésage, dans le cas des coordinations et subordinations ambiguës,
- l'usage d'un ensemble d'occurrences comme tournure,
- la fonction syntaxique d'un groupe d'occurrences,
- etc.

L'insertion de ces annotations se fait pendant la rédaction. Elles sont automatiquement supprimées lorsque la traduction du texte est obtenue et satisfaisante (i.e. validée par l'utilisateur). Dans la maquette, l'utilisateur peut indiquer la classe syntaxique de chaque occurrence, le parenthésage et la fonction syntaxique d'un groupe d'occurrences, ainsi que les tournures.

d. Contrôle de la traduction

Pour contrôler la traduction que produit le système, l'utilisateur peut en demander une rétrotraduction. S'il ne demande pas une rétrotraduction systématique dans les préférences LIDIA, il pourra cependant demander des rétrotraductions ponctuelles.

e. État d'avancement des traitements

Enfin, pour savoir comment progresse le processus de traduction, l'utilisateur peut demander une rétroaction permanente du système. S'il ne choisit pas cette option, il peut interroger ponctuellement les objets en cours de traitement.

f. Type de document

Cette fonctionnalité n'est pas implémentée.

4.3. Mise en œuvre des traitements

Dans un souci d'intégration à l'outil de base du rédacteur — ici HyperCard —, nous avons réalisé les fonctionnalités LIDIA en utilisant les mêmes outils d'interface que le logiciel hôte. On utilise donc la barre de menus, les palettes flottantes, les boutons et les textes flottants.

4.3.1. Menus

Deux nouveaux menus intitulés LIDIA et Messages, sont ajoutés dans les barres de menus standard d'HyperCard.

a. Menu LIDIA

Le menu LIDIA change en fonction des préférences LIDIA de la pile en cours de traitement. Si aucune préférence n'est définie, ce menu permet simplement de les définir. Sinon, le menu se compose de cinq items persistants et de trois items conditionnels.

i. Items persistants

Traiter la sélection

Lorsque l'auteur sélectionne cet item, le curseur se transforme en sélecteur LIDIA (✓), qui permet de désigner l'objet qui va être traduit en cliquant dessus.

Traitement spécial...

Lorsque l'auteur sélectionne cet item, le curseur se transforme en sélecteur paramétrique (✓), qui permet de désigner l'objet sur lequel appliquer les traitements spéciaux. Un dialogue de paramétrage apparaît en fonction de l'état courant de l'objet.

- Si l'objet sélectionné n'a pas encore été traité, l'utilisateur peut choisir les traitements à effectuer en mode normal. Dans ce cas, la séquence originale des traitements est associée à l'objet et les traitements commencent (figure 4.15 ci-dessous).
- Si l'objet sélectionné a déjà été traité avec LIDIA (la traduction a été conduite à son terme), l'utilisateur peut choisir les traitements à effectuer en mode reprise.

Lorsque le dialogue de description du traitement spécial en mode reprise apparaît, le bouton Traduction est remplacé par le bouton Désambiguïsation (figure 4.16). En fin de traitement, on dispose des données qui permettent de n'effectuer que ce traitement sans avoir à refaire une phase d'analyse.

L'utilisateur peut donc choisir de ne refaire que la désambiguïsation. S'il décide qu'un traitement antérieur doit être refait, le bouton Désambiguïsation est bien sûr remplacé par le bouton Traduction (figure 4.17).

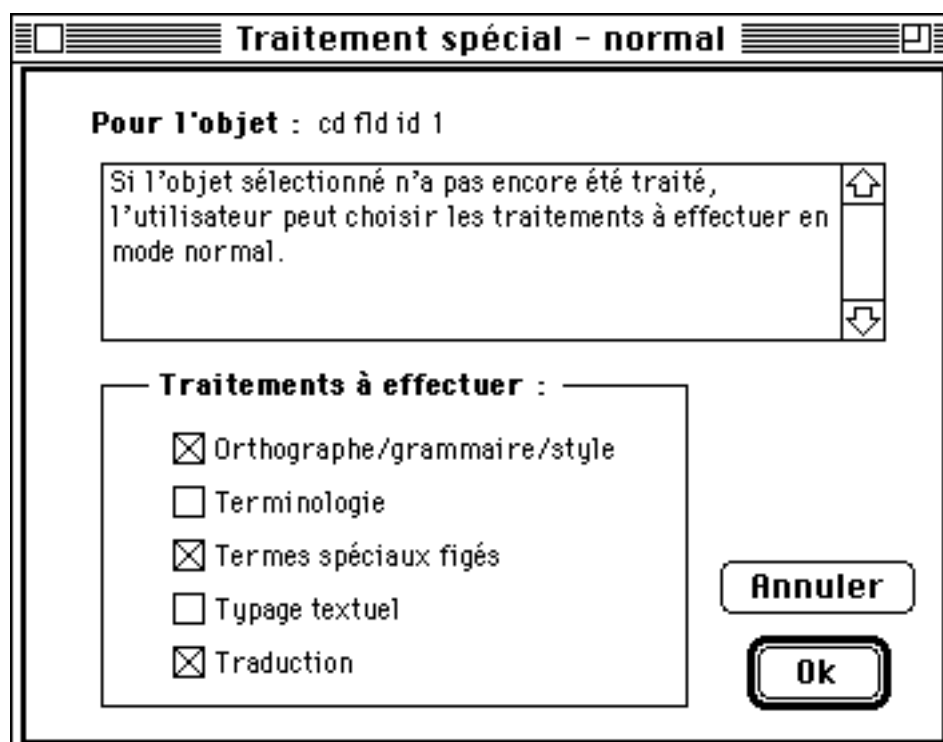


Figure 4.15 : Traitement spécial en mode normal

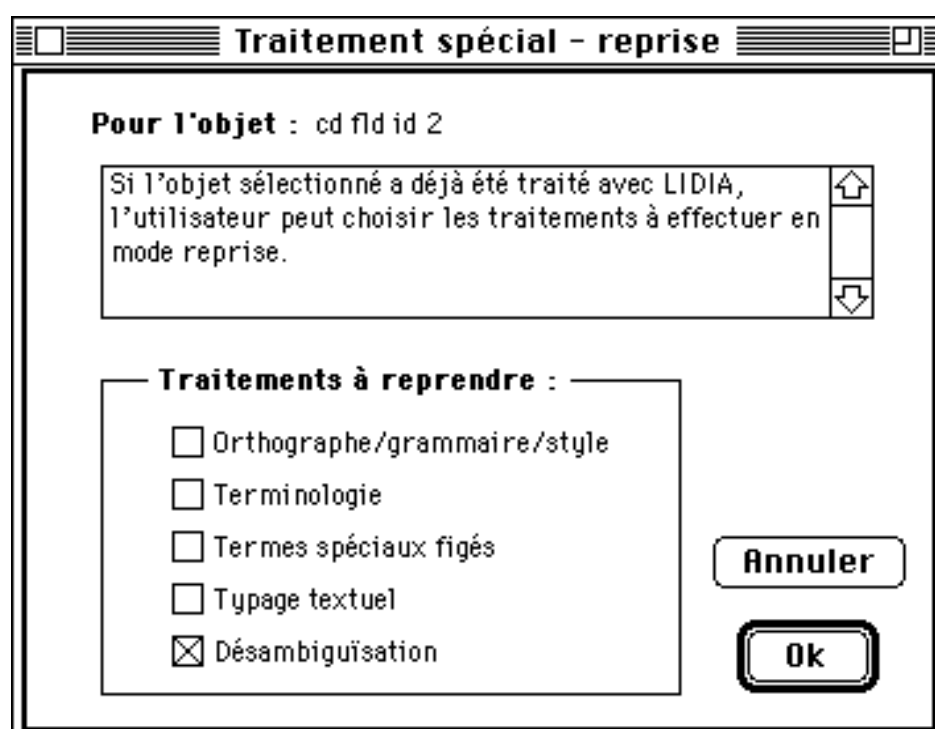


Figure 4.16 : Traitement spécial avec reprise de la désambiguïsation

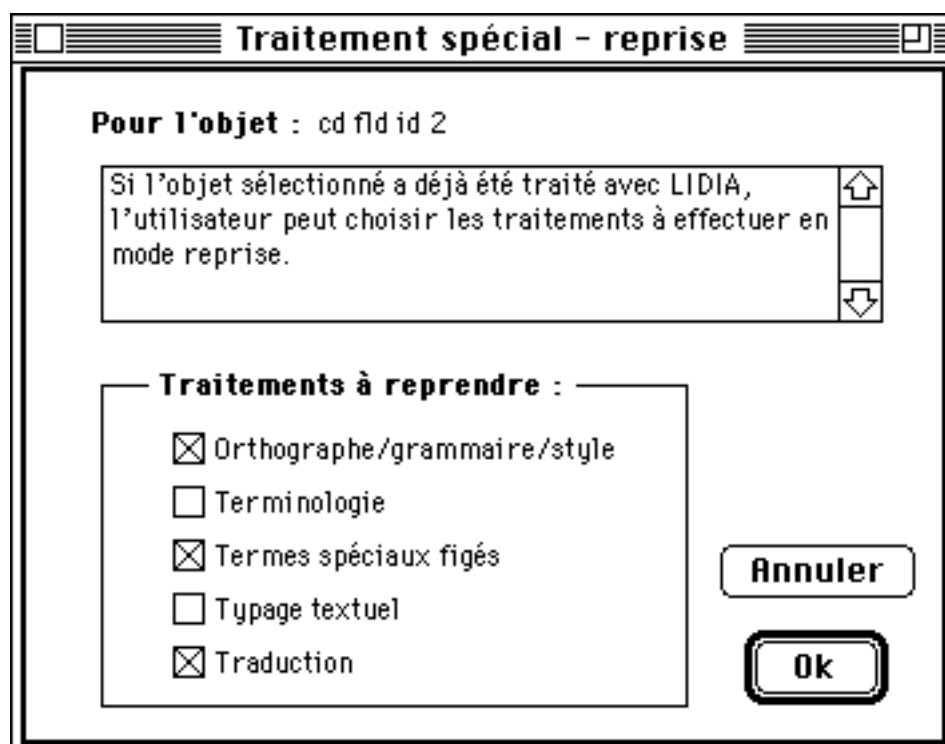


Figure 4.17 : Traitement spécial... reprise complète

Palette LIDIA

Lorsque l'auteur sélectionne cet item, une palette flottante apparaît. Elle permet de choisir tous les items importants du menu et de se déplacer dans la pile courante.

Préférences...

Lorsque l'auteur sélectionne cet item, le dialogue de réglage des préférences décrit au § 4.2. apparaît.

Construire piles cibles

Cet item permet à l'utilisateur de demander la construction des piles dans les langues visées. Nous avons ajouté cet item pour simplifier la maquette. Dans une version plus complète du système, les piles traduites devraient être produites automatiquement.

ii. Items conditionnels

Les items conditionnels sont liés aux choix effectués par l'utilisateur dans le panneau "profil de l'utilisateur" du dialogue de réglage des préférences.

Voir les annotations

Lorsque l'auteur sélectionne cet item, le curseur se transforme en interrogateur d'annotation (🔍). L'utilisateur sélectionne l'objet qu'il veut interroger. Si le processus de désambiguïsation a été effectué, le texte annoté apparaît dans une fenêtre flottante que l'utilisateur peut déplacer et fermer.

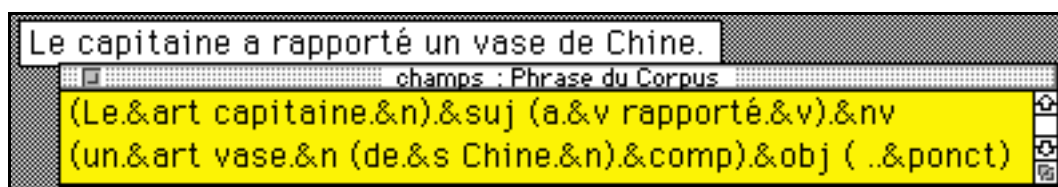


Figure 4.18 : Textoïde montrant les annotations

Voir la rétrotraduction

Lorsque l'auteur sélectionne cet item, le curseur se transforme en interrogateur de rétrotraduction (↔). L'utilisateur sélectionne l'objet qu'il veut interroger. Si le processus de traduction a été effectué, le texte rétrotraduit apparaît dans une fenêtre flottante que l'utilisateur peut déplacer et fermer.

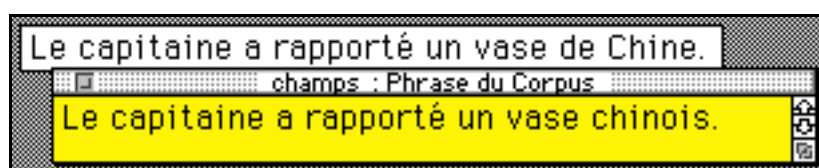


Figure 4.19 : Textoïde montrant la rétrotraduction

Voir l'état du traitement

Lorsque l'auteur sélectionne cet item, le curseur se transforme en inspecteur (🔍). L'utilisateur sélectionne l'objet qu'il souhaite inspecter. Une fenêtre flottante apparaît (figure 4.20) et montre quel est l'état d'avancement des traitements. Les traitements déjà effectués apparaissent en corps normal, le traitement en cours, en gras, et les traitements à venir, en italique.

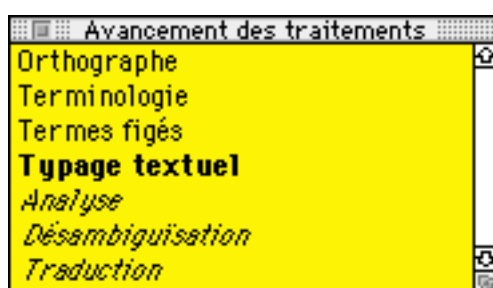


Figure 4.20 : État d'avancement des traitements

La figure 4.21 montre le menu le plus complet dont l'utilisateur puisse disposer.

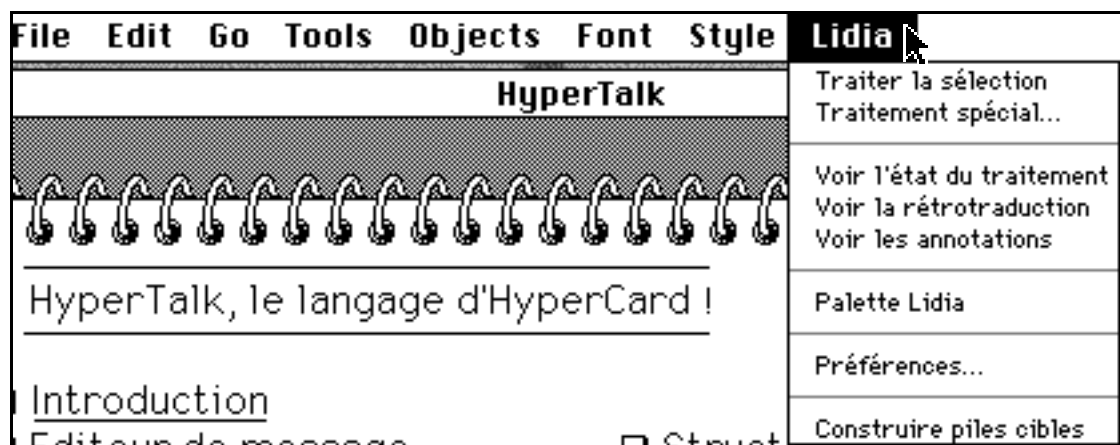


Figure 4.21 : Menu LIDIA complet

b. Menu Messages

Le menu message contient la localisation des objets pour lesquels l'intervention de l'utilisateur est requise. Dans [Blanchon 1990a], nous proposons deux façons d'organiser les entrées dans ce menu, par carte ou par traitement. Ces deux accès sont possibles, le menu se configurant automatiquement sur requête de l'utilisateur.

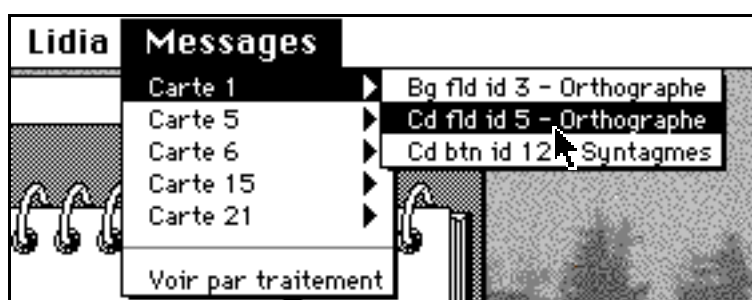


Figure 4.22 : Menu Messages : accès par carte

Le premier ordre que l'on peut établir sur les objets qui requièrent l'intervention de l'utilisateur est un ordre par carte. Au premier niveau du menu Message, on voit les cartes sur lesquelles il y a des problèmes. Au second niveau, pour chaque carte, on montre quels objets requièrent une intervention et quel type de problème est à résoudre.

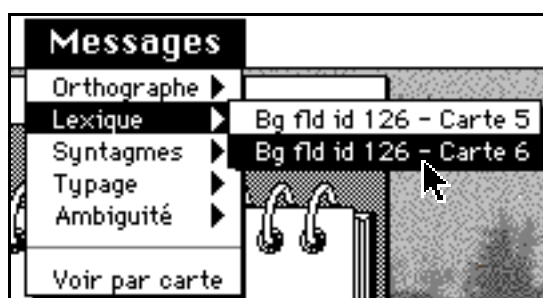


Figure 4.23 : Menu Messages : accès par traitement (1)

Le second ordre que l'on peut établir sur les objets qui requièrent l'intervention de l'utilisateur est un ordre par type de traitement. Au premier niveau du menu Message, on voit alors les types de traitement qui sont en attente. Au second niveau, pour chaque type de traitement, on donne la localisation des objets qui requièrent l'intervention de l'utilisateur.

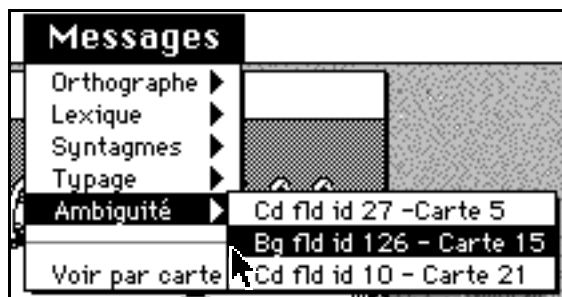


Figure 4.24 : Menu Messages : accès par traitement (2)

Comme un conteneur de fond (Bg) peut être partagé par plusieurs cartes avec des contenus différents, on constate ici (figure 4.23) que deux questions concernent le champ de fond dont l'identificateur est 12.

L'utilisateur peut, à son gré, basculer entre les deux présentations de ce menu en sélectionnant l'item Voir par carte ou Voir par traitement qui apparaît en fin de menu.

Lorsque l'utilisateur sélectionne l'un des items d'un sous-menu, il est transporté sur la carte où se situe le problème. Il peut alors décider d'intervenir immédiatement ou de remettre son intervention à plus tard.

4.3.2. Palettes

À chaque menu est associée une palette qui permet d'accéder directement aux items du menu Lidia. On a ajouté quatre outils qui permettent de se déplacer dans la pile courante.

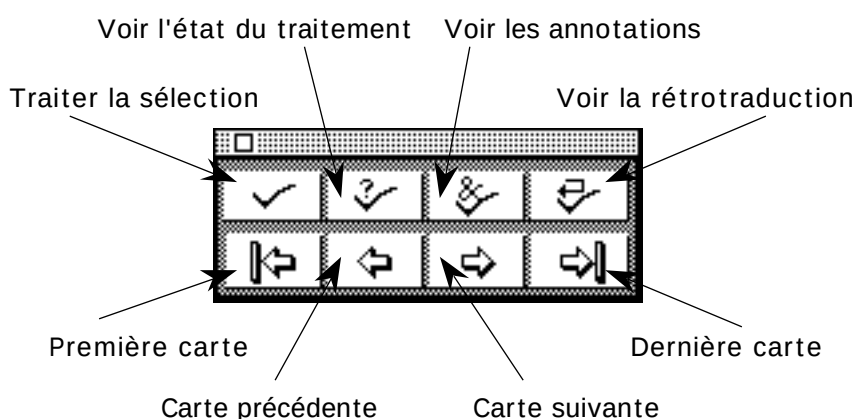


Figure 4.25 : Palette Lidia correspondant au menu de la figure 4.21

La figure 4.25 montre la palette la plus complète dont peut disposer l'utilisateur. Cette palette correspond au cas où les préférences utilisateur attachées à la pile ne prévoient pas de voir automatiquement les annotations du système, l'état d'avancement du

traitement d'un objet et la rétrotraduction. Chacun des outils permet d'interroger l'objet sélectionné.

Huit palettes sont disponibles. Toutes contiennent l'outil Traiter la sélection ainsi que l'une des huit projections des outils Voir l'état du traitement, Voir les Annotations et Voir la rétrotraduction.

4.3.3. Rétroaction

Nous avons vu que l'utilisateur peut régler la rétroaction par trois paramètres booléens : Voir les annotations du système, Voir la rétrotraduction, Voir l'état du traitement. À chacun de ces paramètres est associé une classe de boutons qui se superposent aux objets capables de fournir l'information désirée.

L'état du traitement est disponible à tout moment. On y accède via le bouton .

Les annotations sont disponibles lorsque la phase de désambiguïsation interactive est terminée. On accède à la forme annotée via le bouton .

La rétrotraduction est disponible lorsque la phase de génération syntaxique (phase 7 figure 3.5) est achevée. On y accède via le bouton .

Il y a aussi des étapes du processus de traduction qui nécessitent l'intervention de l'utilisateur. C'est le cas des phases de standardisation et de la phase de désambiguïsation interactive.

Une étape du processus de standardisation peut échouer — la vérification de l'orthographe, de la terminologie et du type de document³⁷ — ou avoir des informations à demander — concernant l'emploi de termes figés, et la première étape du typage³⁸.

Pour une étape donnée, un objet signale à l'utilisateur que son intervention est nécessaire dans le menu Message et par un bouton posé sur sa surface. La difficulté qu'il y a à trouver une icône (de petite taille) qui corresponde à une métaphore de la tâche nous a conduit à utiliser des icônes textuelles.

Une vérification orthographique qui échoue est signalée par : .

Une vérification du typage textuel qui échoue est signalée par : .

Une vérification terminologique qui échoue est signalée par : .

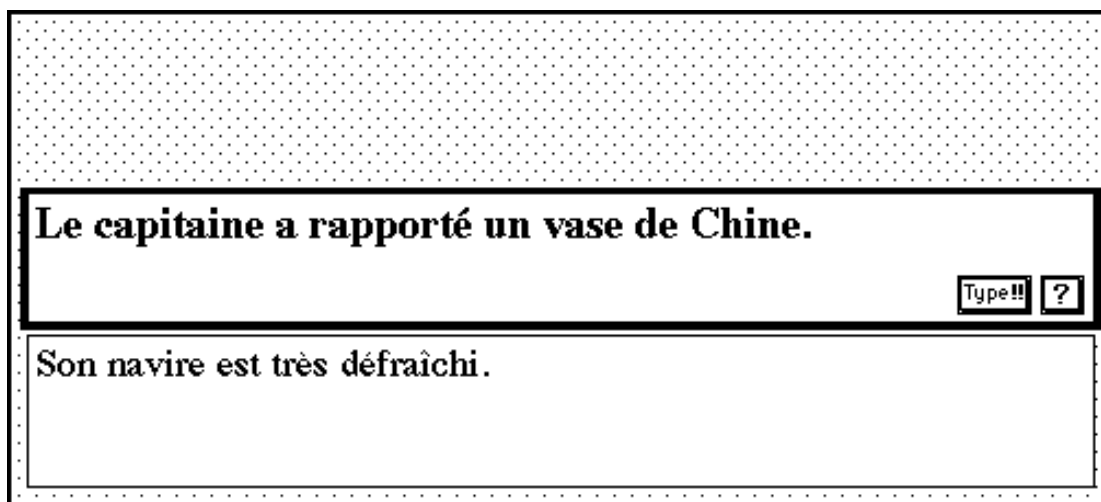
Une hésitation sur l'emploi d'une formule ou d'une tournure est signalée par : .

Enfin, lorsque l'analyse d'un objet a besoin d'être désambiguïsée, celui-ci le signale dans le menu Message et par le bouton suivant : .

³⁷ Il faudra bien, en effet, vérifier que le genre de texte ou le style d'énoncé affecté à un objet du document permet son analyse.

³⁸ Le système demande d'abord à l'utilisateur de donner un type à l'objet. Ce type est ensuite vérifié. Si le système constate que l'objet n'est pas du type prédit par l'utilisateur, il faut retyper, ou modifier le contenu textuel.

Voici quelques exemples chronologiques de rétroaction.

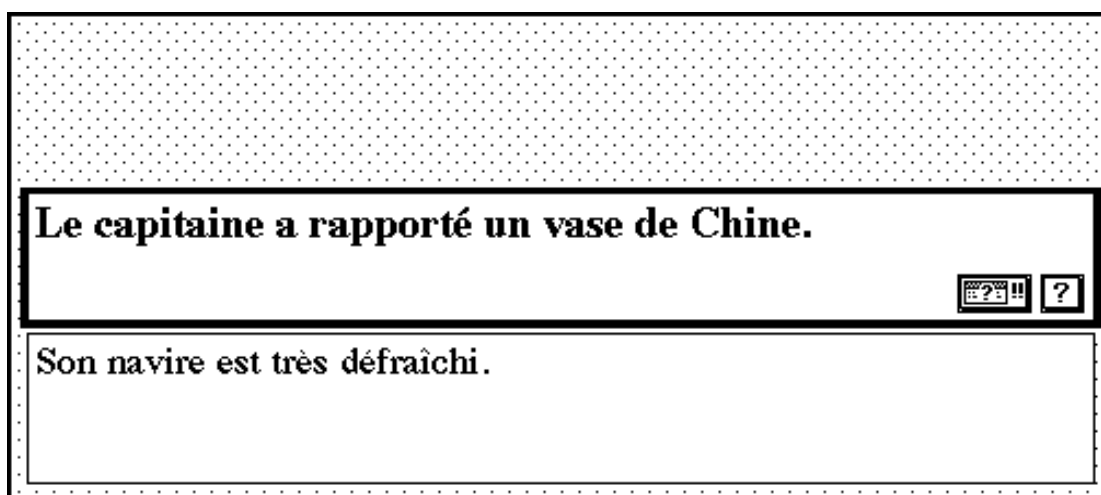


Le capitaine a rapporté un vase de Chine.

Type!! ?

Son navire est très défraîchi.

Figure 4.26 : État du traitement et demande de typage

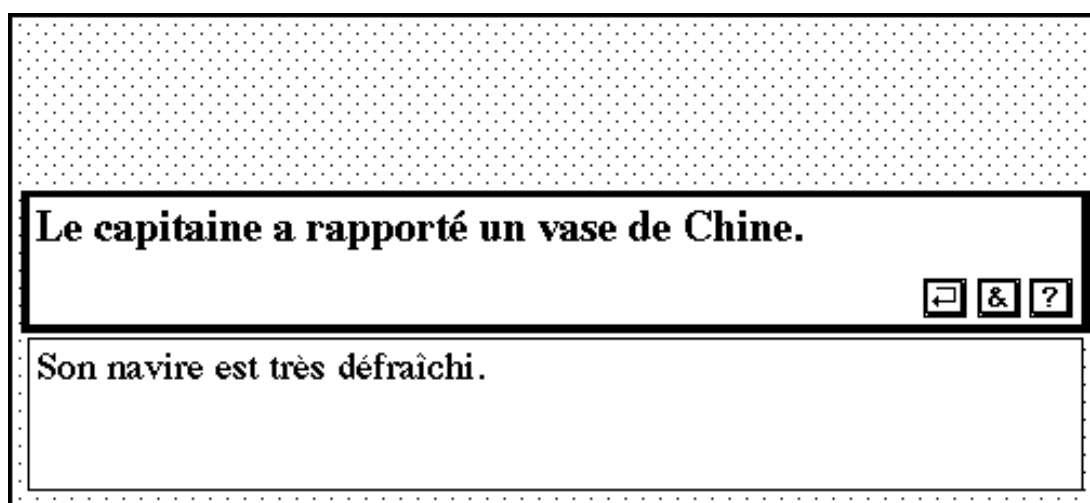


Le capitaine a rapporté un vase de Chine.

?? ?

Son navire est très défraîchi.

Figure 4.27 : État du traitement et demande de clarification



Le capitaine a rapporté un vase de Chine.

← & ?

Son navire est très défraîchi.

Figure 4.28 : État du traitement, annotations et rétrotraduction

Chapitre 5

Standardisation

Lorsque l'utilisateur a défini les préférences du système, il peut demander la traduction des objets de son hypertexte lorsque ceux-ci ont atteint leur stade définitif. Dans ce chapitre, nous examinons les traitements de standardisation. Ces traitements, s'ils sont tous appliqués, doivent permettre :

- d'obtenir un texte correct du point de vue de l'orthographe et de la syntaxe,
- de vérifier que l'auteur utilise la terminologie de son domaine,
- de dire au système que certaines suites de mots sont employées comme des tournures, qu'il s'agisse de "syntagmes spéciaux figés" spécifiques à la pile ("Changer de style...") ou de tournures habituelles (termes et prédicats composés),
- d'associer à chaque objet un "genre de texte" ou un "style d'énoncé", afin de sélectionner les règles appropriées de la grammaire d'analyse, afin de diminuer le degré d'ambiguïté de son contenu textuel.

5.1. Orthographe

Pour effectuer la correction orthographique d'objets HyperCard rédigés en français, nous utilisons dans la maquette un correcteur orthographique commercial ayant fait ses preuves. Dans un système réel, on pourrait aussi développer les fonctions de corrections à partir de la même base de connaissances linguistiques que celle utilisée pour la TA, de façon à garantir un comportement homogène du système dans les différentes phases.

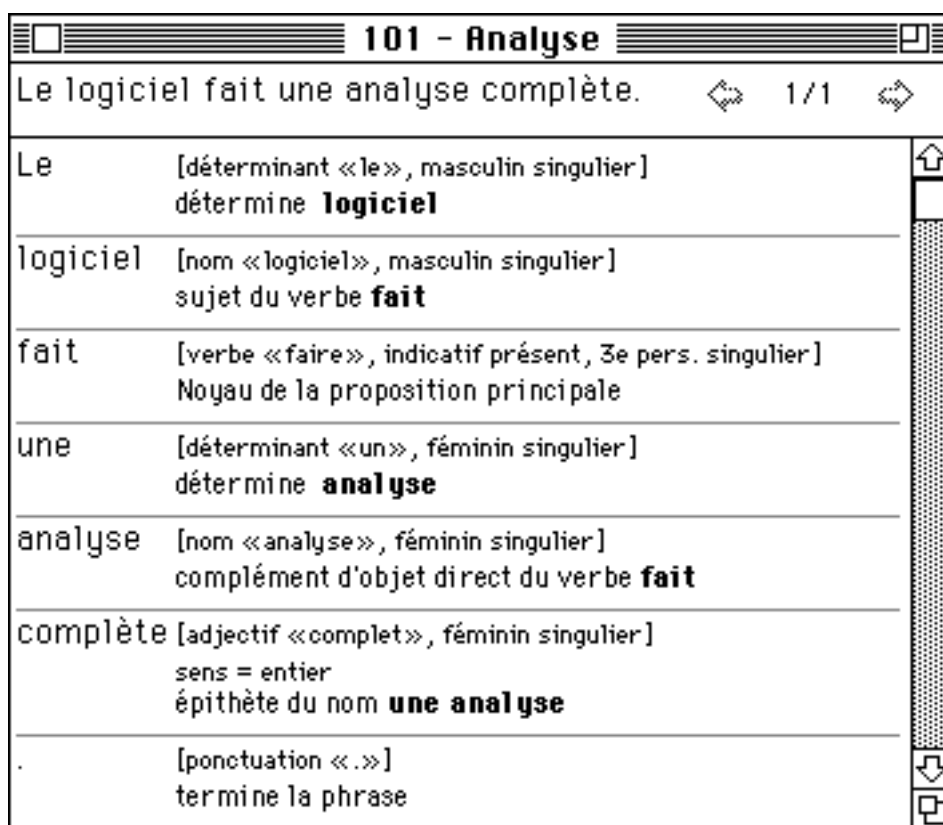
Nous avons partiellement intégré à LIDIA-1 le correcteur orthographique Le Correcteur 101 de la société canadienne Machina Sapiens. Pour faire une intégration complète, il aurait fallu modifier le code source du logiciel, mais la société Machina

Sapiens ne nous y a pas autorisé. En attendant un changement d'architecture de 101³⁹, nous avons utilisé le logiciel tel quel pour la maquette.

5.1.1. L'outil 101 et son intérêt pour LIDIA

Le Correcteur 101 est un logiciel de correction lexicale et grammaticale du français. En s'appuyant sur un dictionnaire et une grammaire du français, 101 est capable de détecter et corriger des fautes d'orthographe, d'accord, d'élision, de ponctuation et autres difficultés de la langue française. On lui soumet un texte rédigé, et on corrige les fautes interactivement, phrase par phrase, ou encore on laisse 101 vérifier tout le texte et produire la liste des passages incorrects associés aux corrections à apporter.

101 est, à notre connaissance, le seul correcteur à montrer le résultat d'analyse qu'il produit. Cette fonctionnalité est un avantage pour LIDIA, car elle permet au rédacteur de voir un résultat d'analyse sous une forme lisible et ainsi de s'initier à l'analyse grammaticale et aux annotations LIDIA.



101 - Analyse	
Le logiciel fait une analyse complète. 1/1	
Le	[déterminant «le», masculin singulier] détermine logiciel
logiciel	[nom «logiciel», masculin singulier] sujet du verbe fait
fait	[verbe «faire», indicatif présent, 3e pers. singulier] Noyau de la proposition principale
une	[déterminant «un», féminin singulier] détermine analyse
analyse	[nom «analyse», féminin singulier] complément d'objet direct du verbe fait
complète	[adjectif «complet», féminin singulier] sens = entier épithète du nom une analyse
.	[ponctuation «.»] termine la phrase

Figure 5.1 : Résultat d'analyse produit par 101

Le fonctionnement d'un analyseur de correcteur orthographique est complètement inversé par rapport à celui d'un analyseur utilisé dans un processus de traduction

³⁹ La société Machina Sapiens envisage de séparer le moteur d'analyse de son interface en le rendant "scriptable" et donc accessible depuis n'importe quelle application de manière transparente. Cette version autonome de l'analyseur devrait être en test en février ou mars 1994. Communication personnelle de Claude Coulomb, le 26 octobre 1993.

automatique. Pour 101, les occurrences du texte doivent être remises en question, tandis que, pour l'analyseur LIDIA, les occurrences du texte sont supposées correctes.

Pour la phrase "Il prend des classeurs et des cahiers noirs.", le correcteur répond : "Aucune erreur détectée⁴⁰".

Pour la phrase "Il prend des classeurs et un cahier noirs", le correcteur répond : "Phrase originale apparemment correcte ; mais cette phrase peut être écrite d'au moins deux façons". Le système propose aussi la lecture "Il prend des classeurs et un cahier noir."

Les dictionnaires de 101 contiennent des informations utiles pour le préindexage des mots inconnus. Les informations ainsi saisies peuvent servir de base à une insertion du nouveau vocabulaire dans la Base Lexicale Multilingue par Acceptions.

Les catégories des mots que l'on peut ajouter dans les dictionnaires sont : Nom de famille, Prénom, Lieu géographique, Raison sociale, Nom Commun, Adjectif, Adverbe, Ethnie habitat et Interjection.

The screenshot shows a software window titled "Nouveau mot:" with the word "multimédias" entered. Below this, there are two dropdown menus: "Catégorie:" set to "Adjectif" and "Dictionnaire:" set to "Temporaire".

Below the dropdowns, there are two columns of checkboxes. The first column has checkboxes for "Singulier" and "Pluriel", both of which are checked. The second column has checkboxes for "Masculin" and "Féminin", both of which are checked. To the right of these checkboxes are two input fields for the word forms: "multimédia" and "multimédias".

Below the checkboxes, there is a section titled "On peut dire:" with two example sentences, each preceded by an unchecked checkbox:

- ☐ Il est multimédia que Lise est aimable.
- ☐ Je trouve multimédia que Paul soit aimable.

At the bottom right of the window are two buttons: "Annuler" and "Ajouter".

Figure 5.2 : Saisie d'un nouvel adjectif avec 101

⁴⁰ Les phrases ont été soumises à 101 et les réponses du logiciel sont reprises intégralement.

Figure 5.3 : Saisie d'un nouveau nom commun avec 101

5.1.2. Scénarios envisagés

En attendant une version autonome du moteur de l'analyseur 101, nous avons imaginé deux scénarios.

Avec un analyseur scriptable⁴¹, l'appel se ferait de façon transparente pour l'utilisateur. Si l'analyseur rencontre un problème, on le signale à l'utilisateur via le menu Message et le bouton **Faute!!** que nous avons décrit au § 4.3. Lorsque l'utilisateur décide de répondre à une question, l'analyseur et son interface prennent le contrôle. Lorsque la correction est terminée, et si nécessaire, le texte corrigé remplace le texte erroné et les traitements de l'objet se poursuivent.

Avec le correcteur, tel qu'il existe actuellement, le scénario est bien différent. L'objet qui doit être corrigé simule la faute via le menu Message et le bouton **Faute!!**. Lorsque la correction est terminée et si nécessaire, le texte corrigé remplace le texte erroné et les traitements de l'objet se poursuivent. La grande différence avec le scénario précédent réside dans le fait que, si l'utilisateur a choisi la correction orthographique, tous les objets ont le bouton Faute!! actif. L'utilisateur peut donc être interrompu pour corriger un texte exact puisque l'on ne peut pas faire de test de correction avant de faire appel à l'utilisateur.

⁴¹ Sur la notion de script, se reporter au § 8.2.2.

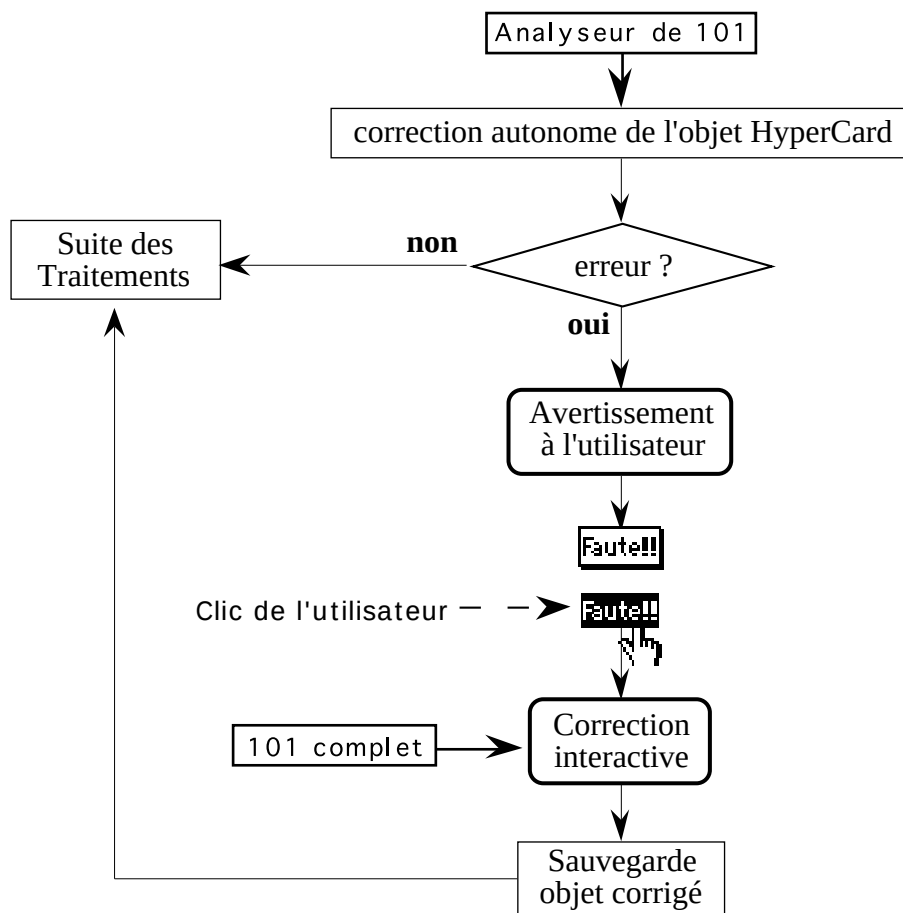


Figure 5.4 : Correction orthographique, scénario imaginé

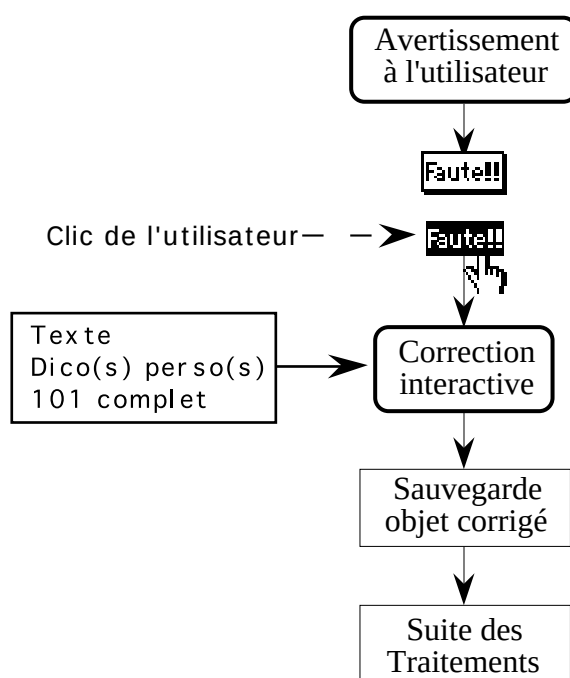


Figure 5.5 : Correction orthographique, scénario réalisé

5.1.3. Correction

Les schémas suivants illustrent l'interaction avec 101.

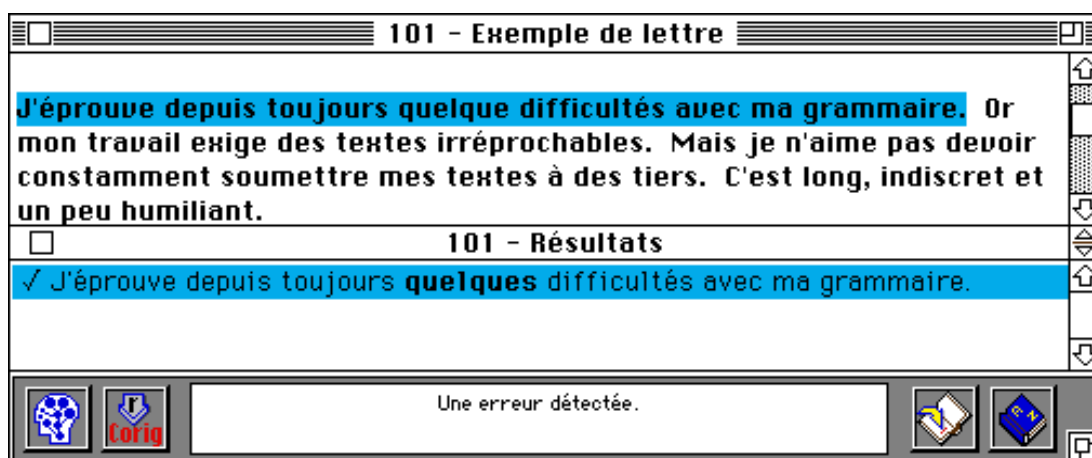


Figure 5.6 : 101 détecte une erreur

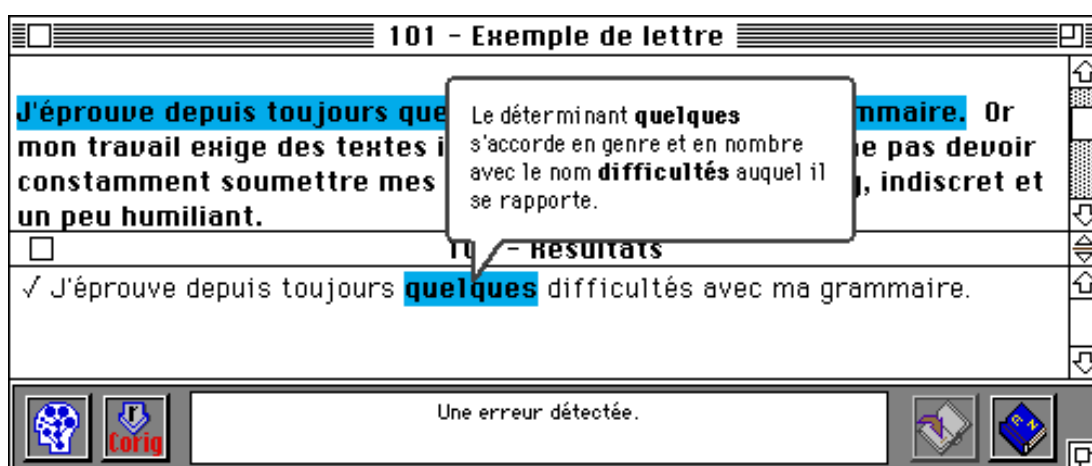


Figure 5.7 : 101 explique l'erreur

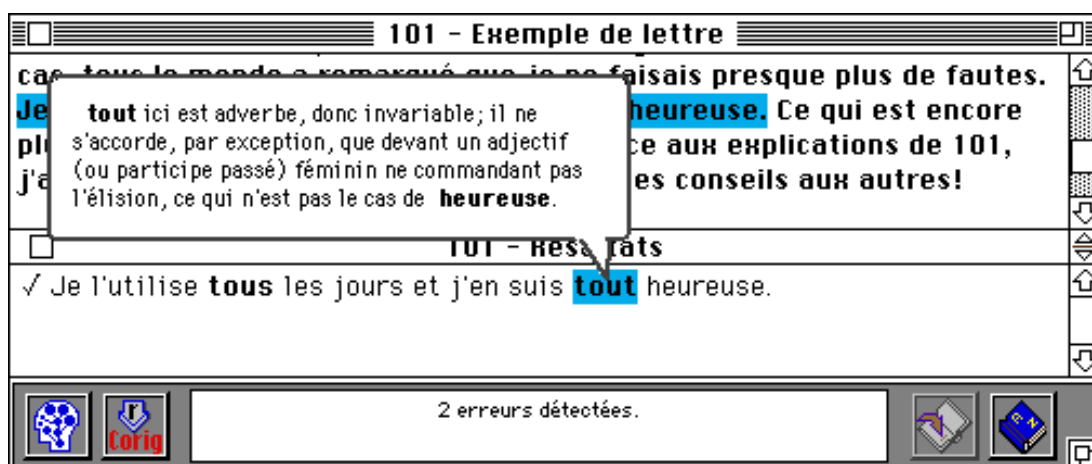


Figure 5.8 : 101 trouve une autre erreur

5.2. Préférences lexicales

Le module de préférences lexicales permet à un auteur de rédiger avec le vocabulaire exact du domaine de son document, ou avec le vocabulaire obligatoire dans sa société. Dans certaines situations, il existe des termes uniques pour désigner certains objets ou certaines actions. Cette univocité nous intéresse en traduction automatique, puisqu'elle diminue le degré de polysémie des mots.

Pour standardiser lexicalement un texte, on compte utiliser les thésaurus personnels de l'utilisateur dans lesquels on trouvera, pour chaque terme, le terme qui doit être employé dans le document courant. On peut par exemple imaginer que tous les synonymes de "avion" doivent se transcrire avec le mot "appareil" dans un certain contexte, et par "aéronef" dans un autre.

L'entrée d'un tel module se compose du texte à traiter et des thésaurus personnels de l'utilisateur. Ces thésaurus contiennent, pour chaque mot obligatoire, la liste des mots qu'il doit remplacer. Chaque mot du texte est lemmatisé, et la forme canonique obtenue est recherchée dans les thésaurus. Si elle y figure comme forme obligatoire, on ne fait rien.

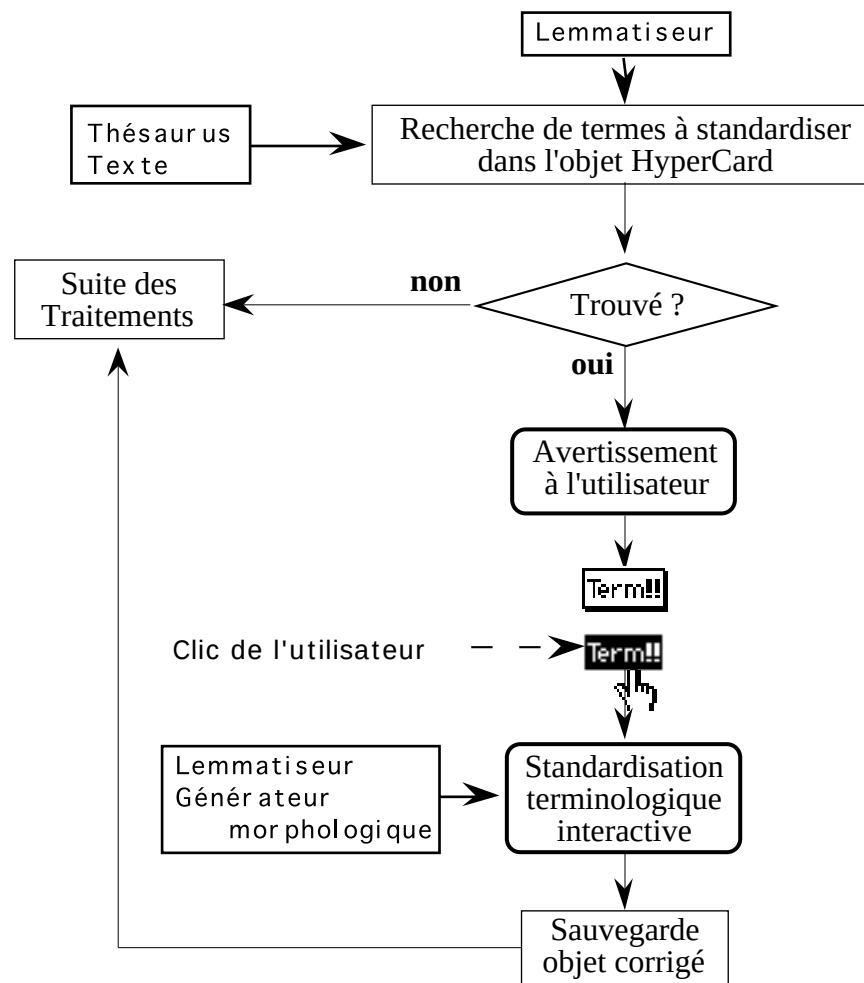


Figure 5.9 : Standardisation terminologique, scénario imaginé

Si l'occurrence doit être remplacée, deux cas peuvent se produire :

- Le mot standard a le même genre et le même nombre que le mot qu'il doit remplacer. Dans ce cas,
on produit la forme fléchie de la forme obligatoire correspondante, puis on remplace l'occurrence du texte.
- Le mot standard diffère en genre et/ou en nombre avec le mot qu'il doit remplacer. Dans ce cas,
le remplacement implique des modifications syntaxiques de la phrase non localisées au mot à remplacer. Dans ce cas, l'une des solutions envisageables est d'effectuer le remplacement en demandant à l'auteur de propager les contraintes syntaxiques⁴².

Le remplacement d'une occurrence du texte origine est bien entendu soumis à l'accord de l'utilisateur, car un mot ordinairement interdit peut être nécessaire dans certaines conditions. Soit par exemple, un texte rédigé au sein de la société X qui parle des produits de la société X et de la société Y. Les produits de la société X doivent être dénotés par des termes en usage chez X, les produits de la société Y peuvent être notés par des termes en usage chez Y qui sont en conflit avec les normes de X.

Lorsque l'auteur accepte d'effectuer la standardisation terminologique sur un objet — en cliquant sur le bouton **Term!!** ou en sélectionnant l'objet dans le menu Message —, un dialogue apparaît, dans lequel figure notamment le texte à standardiser.

Le traitement commence dès que le dialogue apparaît. L'occurrence surlignée est en cours de traitement. Si le module décide que cette occurrence devrait être remplacée, deux cas peuvent se produire; selon que le remplacement induit ou non des modifications syntaxiques de surface. Dans le premier cas un message d'alerte est affiché et on présente le mot préféré à l'utilisateur.

La figure 5.10, le système propose à l'utilisateur de remplacer le mot “avion” par le mot “appareil”. L'utilisateur peut choisir de remplacer “avion” par “appareil”, ou bien d'ignorer la suggestion :

- s'il choisit Remplacer, le système remplace l'occurrence et continue l'examen du texte. Ici, il n'y a pas d'autre modification à effectuer, et le traitement se termine (figure 5.11).
- S'il choisit Ignorer, le traitement se termine (figure 5.12).

⁴² On pourrait aussi songer à effectuer ce traitement de standardisation sur l'arbre retenu après la phase de désambiguïsation interactive. Après le remplacement de l'occurrence non standard, il faudrait effectuer une phase de propagation de contraintes et faire les transformations lexicales qui s'imposent.

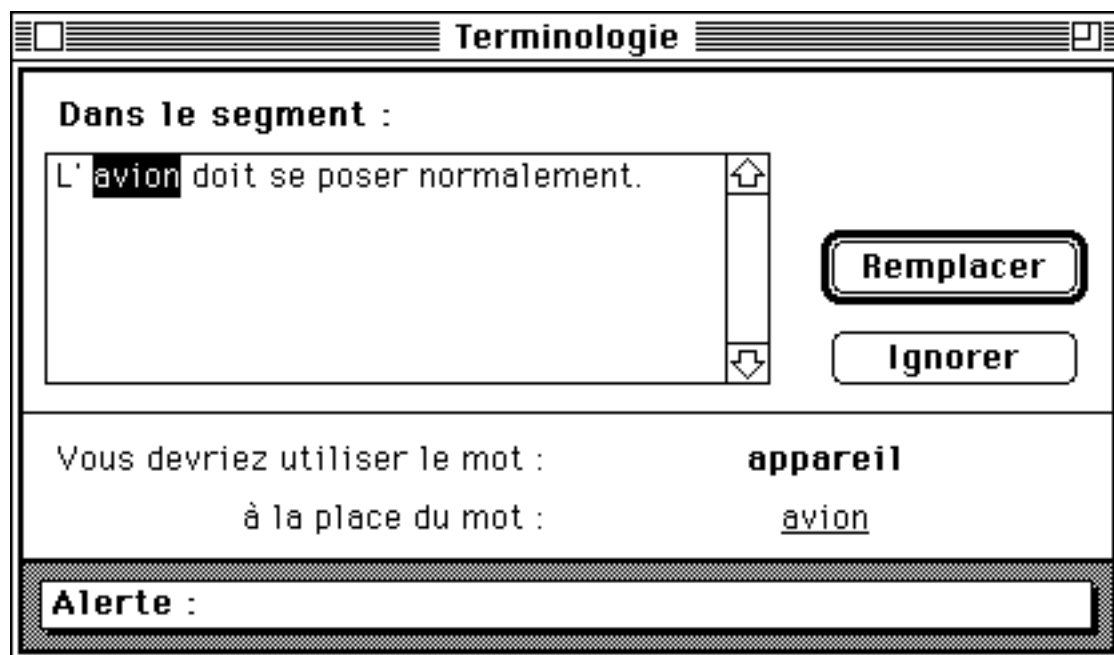


Figure 5.10 : Détection d'un terme non standard

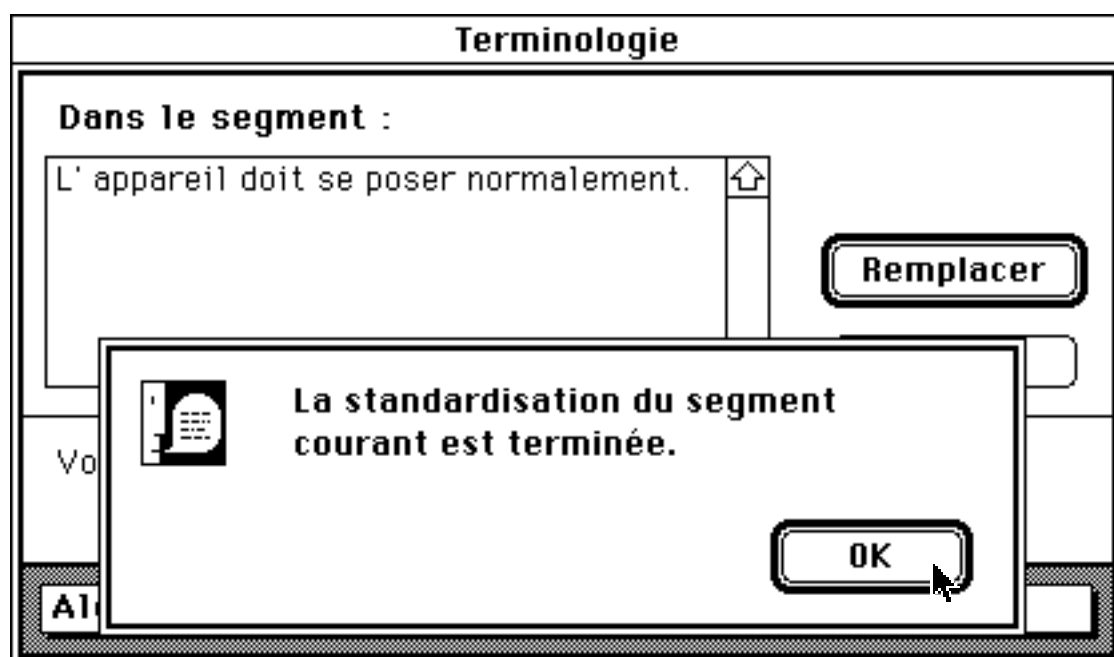


Figure 5.11 : Validation de remplacement par un terme standard sans modification morphologique

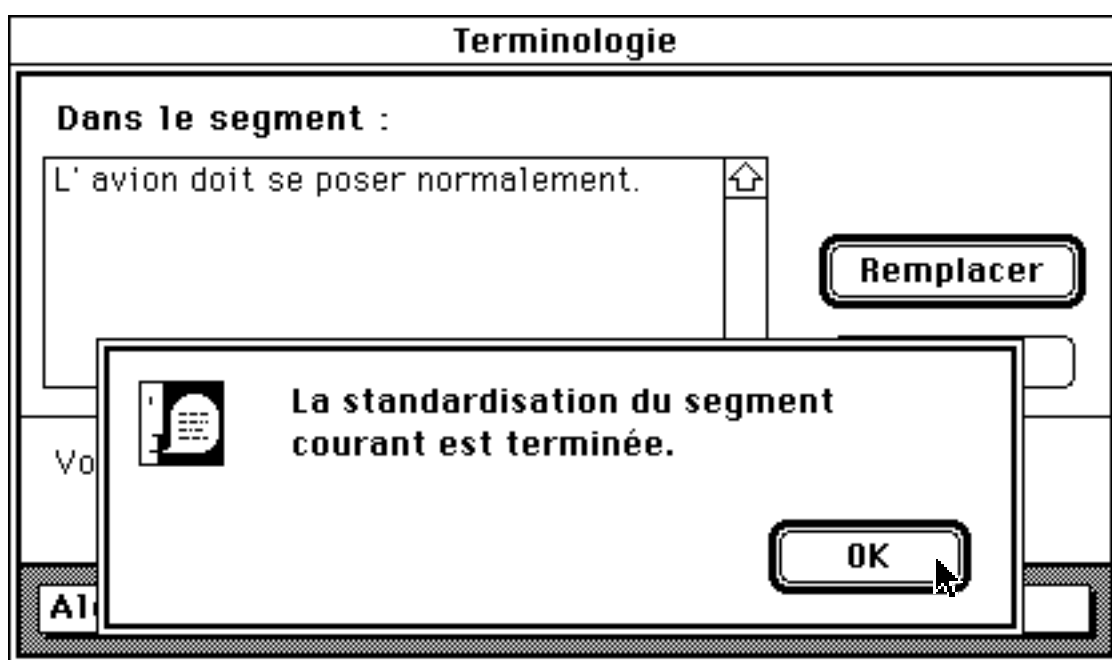


Figure 5.12 : Maintien de l'utilisation d'un terme non standard

Si le remplacement induit des modifications syntaxiques, on affiche une alerte qui signale la nature des conflits. Dans l'exemple suivant, on veut remplacer "production" par "produits". Le bouton Remplacer est transformé en bouton Éditer (figure 5.13).

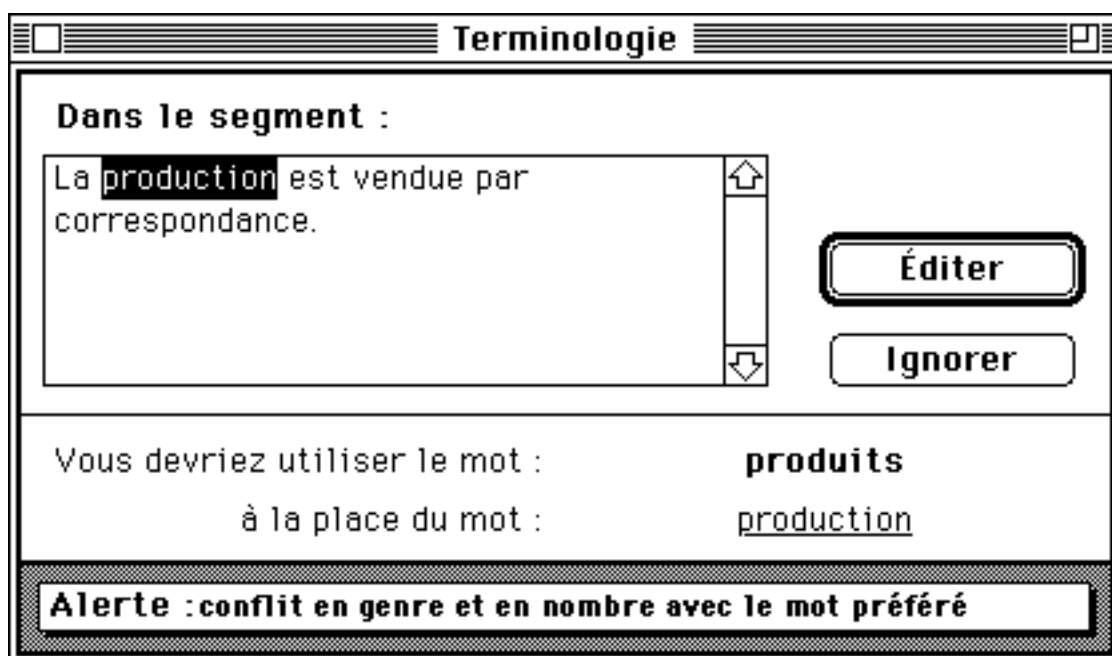


Figure 5.13 : Détection d'un terme non standard en conflit en genre et en nombre avec le terme standard

L'utilisateur peut choisir d'ignorer la proposition ou d'éditer le texte afin d'effectuer les modifications nécessaires.

S'il choisit d'éditer, l'occurrence non standard est remplacée, et le texte est déverrouillé afin que l'utilisateur puisse effectuer les modifications. Ces modifications effectuées, l'utilisateur demande à Continuer le traitement (figure 5.14).

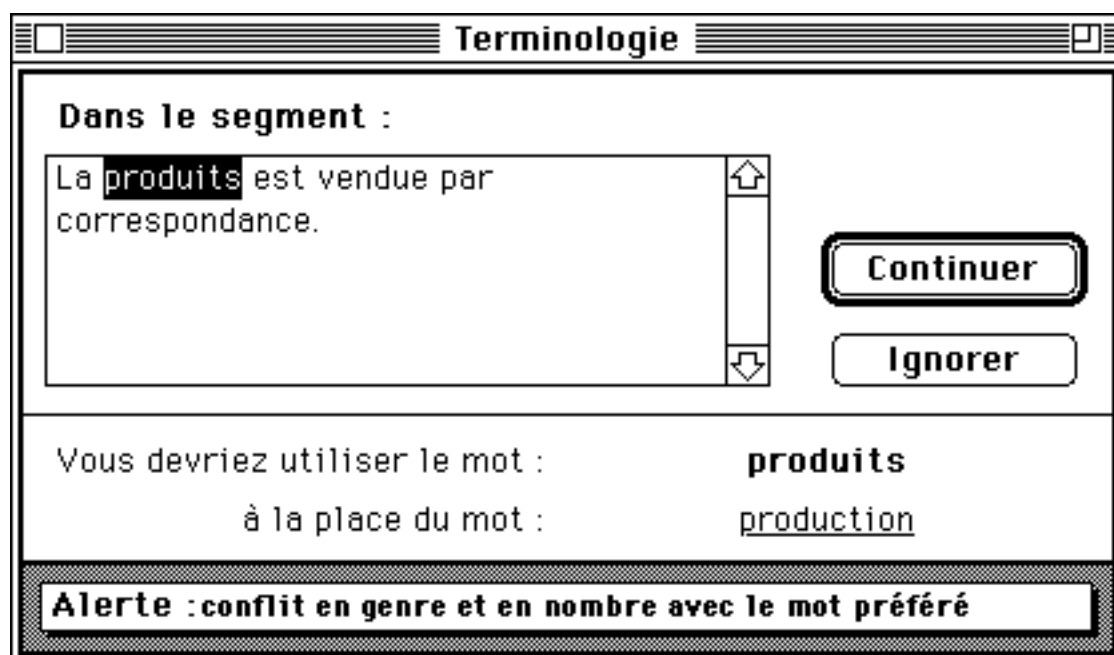


Figure 5.13 : Remplacement d'un terme avec conflit en genre et en nombre

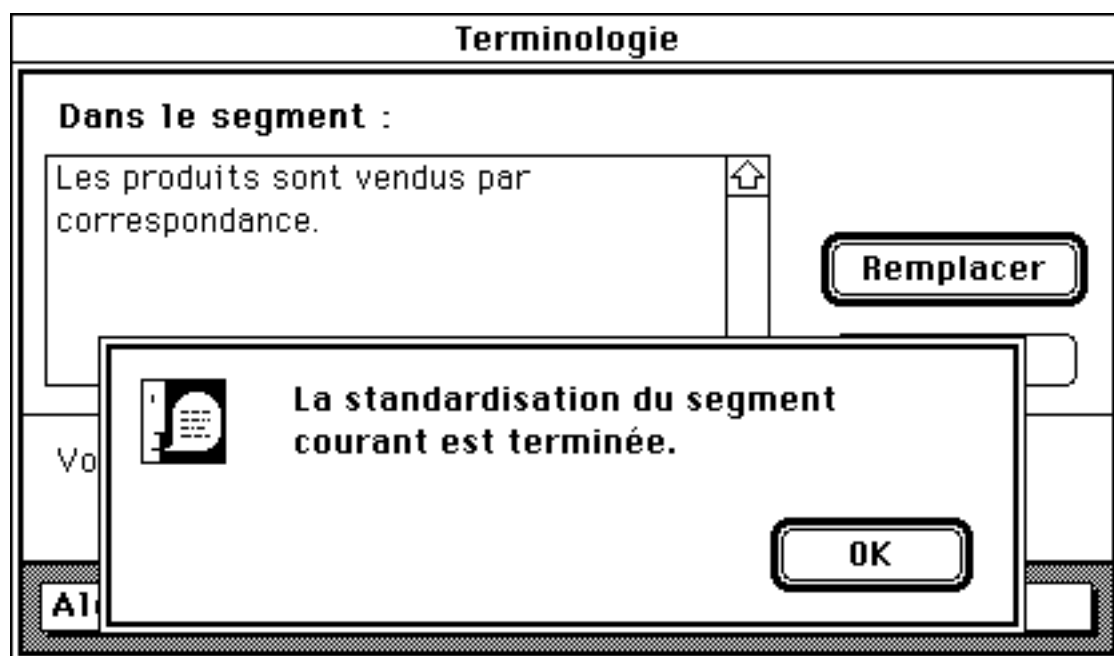


Figure 5.14 : Poursuite de la standardisation terminologique après modification morphologique du texte origine

5.3. Tournures et syntagmes figés spéciaux

Le module de traitement des tournures et syntagmes figés spéciaux⁴³ n'a pas été étudié en profondeur. Le corpus comporte volontairement peu de syntagmes figés. En effet, pour faire un véritable module opérationnel et général, il faut utiliser d'autres outils linguistiques et au moins un analyseur morphologique permettant de lemmatiser les tournures connexes fléchissables.

Nous proposons le scénario suivant pour ce module. Le texte de l'objet à traiter est d'abord lemmatisé. On recherche les syntagmes figés dans la forme lemmatisée. Si on en trouve, on demande alors à l'utilisateur d'intervenir (Menu Messages, bouton **Form!!**). Pour chaque tournure que l'on a trouvée, on demande à l'utilisateur de dire si oui ou non l'ensemble des occurrences doit être lu comme une tournure ou bien littéralement. S'il s'agit d'une tournure, les occurrences qui la composent seront marquées afin que l'analyseur tire parti de cette information.

Une tournure est encadrée par les chaînes de caractères "&<" et ">&".

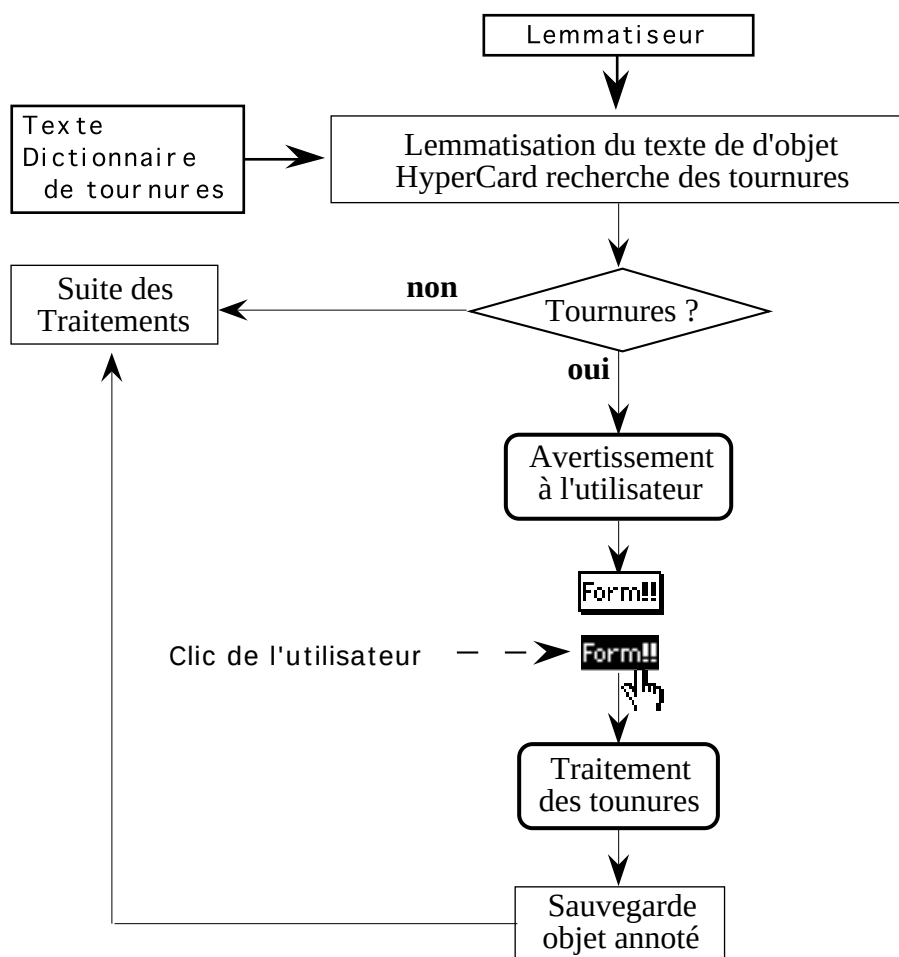


Figure 5.15 : Scénario envisagé pour le traitement des tournures

⁴³ Ici, le mot tournure doit être compris au sens habituel. Un syntagme figé spécial est un terme propre au type de document, du genre "Cacher les bulles d'aide".

Lorsque l’auteur accepte d’effectuer le repérage des tournures sur un objet, un dialogue apparaît, dans lequel figure notamment le texte à standardiser.

Le traitement commence dès que le dialogue apparaît.

Lorsque le système découvre un groupe d’occurrences contiguës qui constituent peut-être une tournure, il surligne le groupe. On montre à l’utilisateur le sens que prend cet ensemble de mots s’il est lu comme une tournure. L’utilisateur doit indiquer au système si il a écrit une tournure ou non :

- si c’est le cas (Oui), le système pose les marques de tournure et examine le reste du texte à la recherche d’autres tournures potentielles,
- si ce n’est pas le cas (Non), le système examine le reste du texte à la recherche d’autres tournures potentielles.

Voici les déroulements possibles de la session de traitement des tournures pour la phrase “Il convient avec le gardien de l’ordre du jour.”

Le système propose “gardien de l’ordre” comme tournure possible.



Figure 5.16 : Détection de la tournure potentielle “gardien de l’ordre”

Si l’utilisateur valide “gardien de l’ordre” comme tournure, le système pose les marques autour de “gardien de l’ordre”, et l’examen du texte se termine.

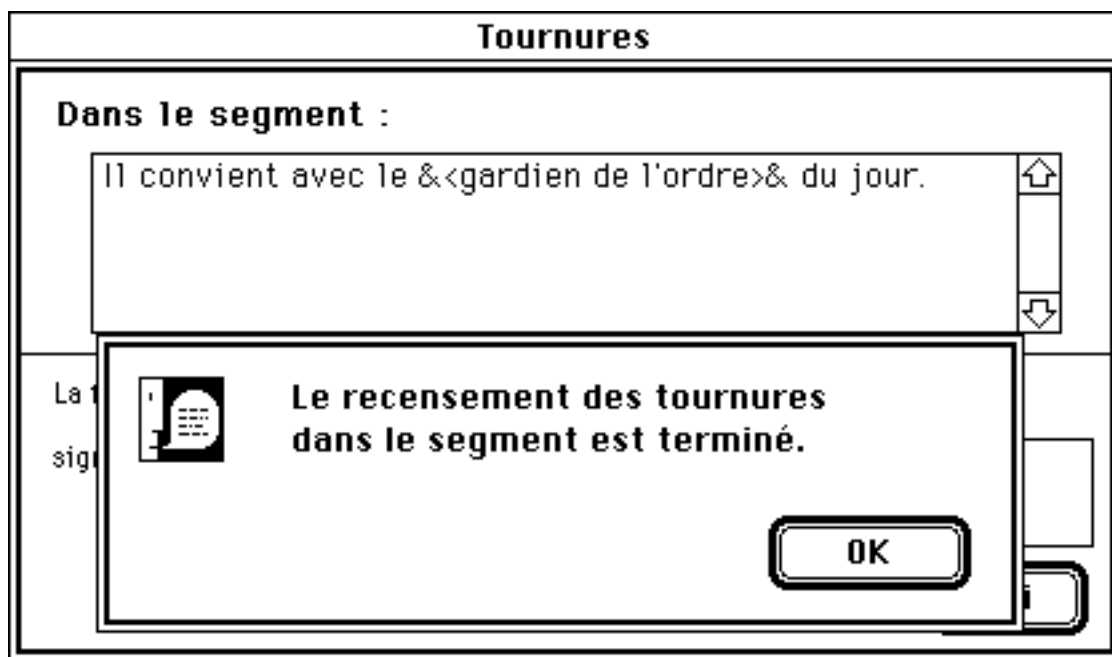


Figure 5.17 : Validation de la tournure “gardien de l’ordre”

Si l'utilisateur ne valide pas “gardien de l’ordre” comme tournure, le système propose “ordre du jour”. Si l'utilisateur valide la tournure, le système pose les marques et le processus se termine. Sinon, le processus se termine immédiatement.

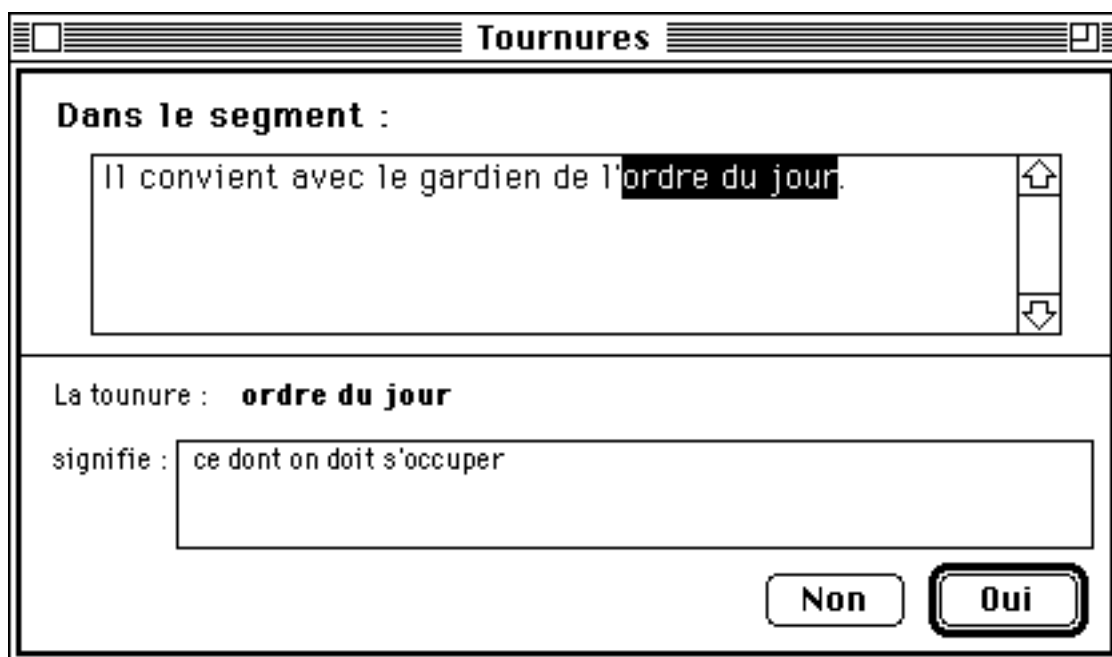


Figure 5.18 : Détection de la tournure potentielle “ordre du jour”

Si le même groupe d'occurrences apparaît à plusieurs reprises dans le segment en cours de traitement, le système demande à chaque fois à l'utilisateur si ces occurrences définissent une tournure, en donnant comme défaut la dernière réponse obtenue.

5.4. Typage

Le module de typage textuel permet à l'utilisateur d'affecter un genre de texte ou un style d'énoncé aux objets à traduire. Cette étiquette permet de sélectionner un sous-ensemble de l'ensemble des règles d'analyse du français, mais aussi un graphe de contrôle d'enchaînement des règles retenues.

Pour l'instant, l'analyseur du français ne permet pas vraiment de procéder de cette façon. Nous avons quand même trois types de texte : les éléments à ne pas traduire, les titres, et le reste.

Comme le module de traitement des formules figées, le module de typage textuel n'est pas un module qui opère des vérifications sur le texte à traiter puis alerte le rédacteur si nécessaire. Le module de typage alerte systématiquement l'utilisateur afin que celui-ci affecte un type à l'objet.

Une vérification doit être faite a posteriori. Si le type que l'utilisateur a affecté à un objet ne permet pas d'en faire l'analyse, c'est que l'utilisateur a mal typé. Dans ce cas, il faudrait prévoir un outil de "retypage" plus perfectionné que le premier afin de l'on puisse profiter du diagnostic de l'analyse. L'outil de vérification du type n'a pas été prototypé dans LIDIA-1.

La figure 5.19 illustre le scénario envisagé.

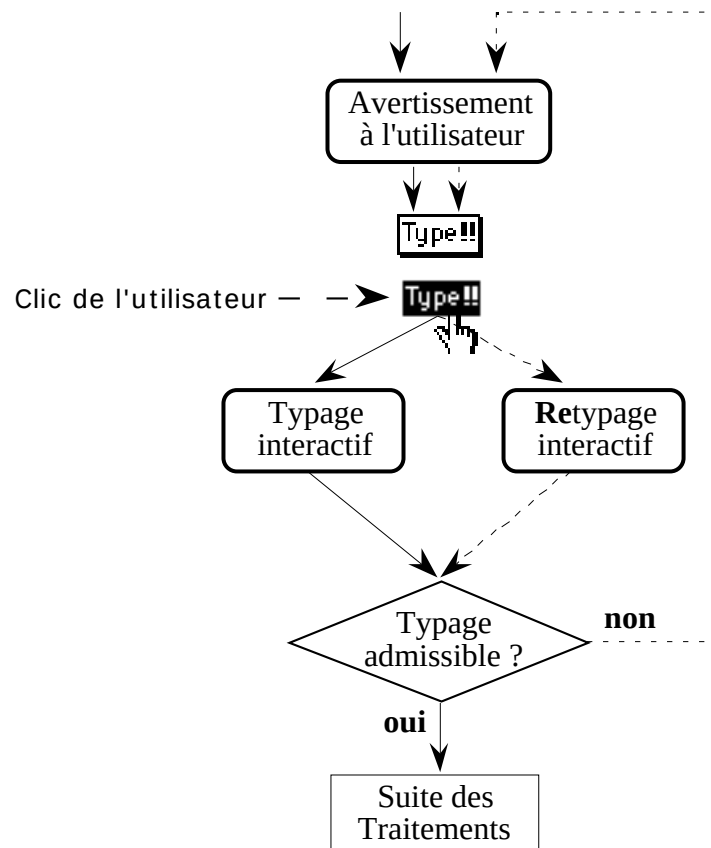


Figure 5.19 : Typage textuel, scénario imaginé

Pour qu'un utilisateur type un objet, nous proposons de lui montrer le texte à typer et la liste des types disponibles.

Trois types sont actifs pour le prototype :

- --Ne pas traduire-- : le texte doit être reproduit tel quel dans le document en langue cible,
- --Sans type-- : ce type doit être utilisé si l'utilisateur ne sait pas choisir,
- -Titre : ce type est utilisé pour des titres grammaticalement simples.

Les autres types que l'on voit dans la figure 5.20 sont une sélection des types proposés par Denos [Denos 1993] lors d'une expérience de typage à la main des piles d'aide de la version francisée d'HyperCard 2.0. Nous avons complété le travail de Denos en définissant et exemplifiant chacun des types retenus.

On peut :

- sélectionner un type et demander des exemples (si ceux-ci sont pertinents).
- sélectionner un type et demander sa définition.
- sélectionner un ou plusieurs types et valider son choix. Le choix peut être validé en cliquant sur le bouton OK ou en fermant le dialogue.

Si l'utilisateur sélectionne le style d'énoncé -Bouton d'action (figure 5.20), il peut en demander la définition, voir des exemples et valider le type choisi en appuyant sur OK.

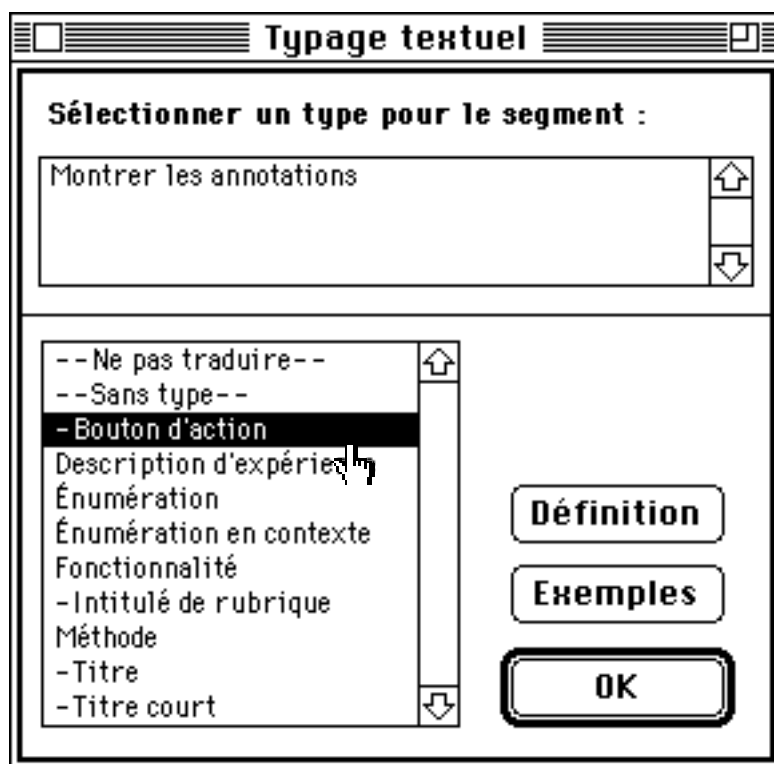


Figure 5.20 : Sélection du style d'énoncé -Bouton d'action défini et exemplifié

Si l'utilisateur demande des exemples, le panneau de typage du dialogue est remplacé par le panneau des exemples (figure 5.21). Depuis les exemples, on peut demander la définition du style ou revenir au panneau principal du typage en appuyant sur OK.

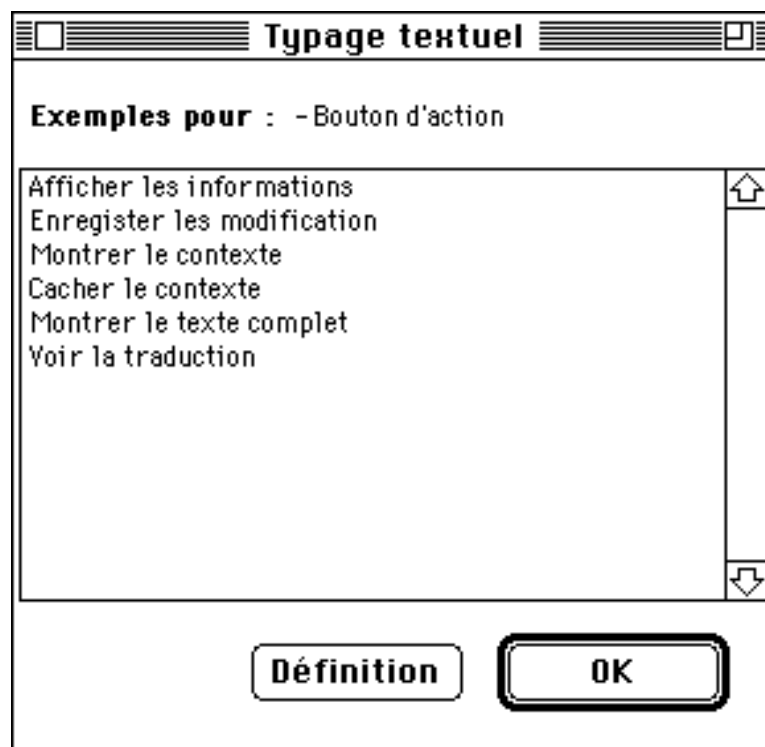


Figure 5.21 : Exemples associés au style d'énoncé -Bouton d'action

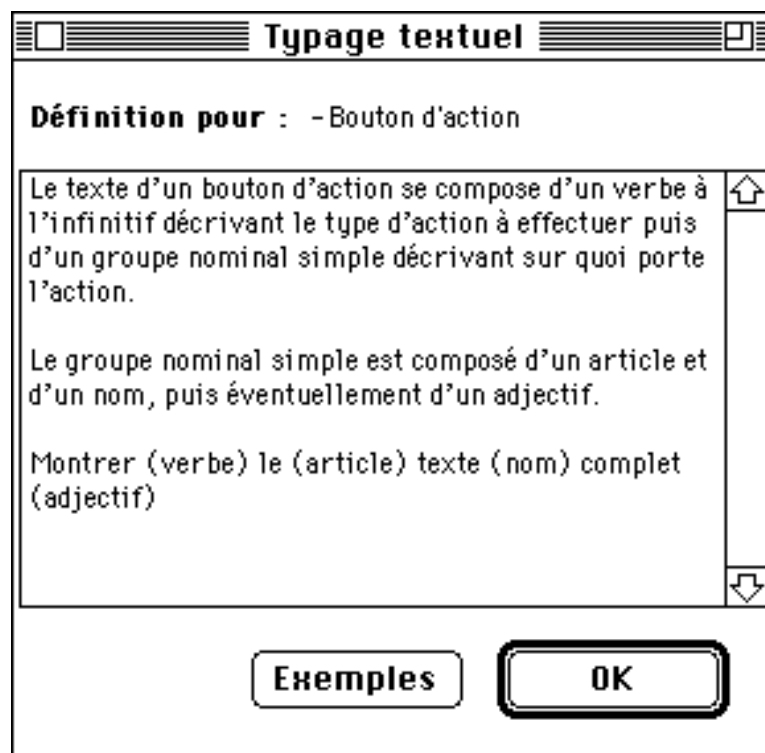


Figure 5.22 : Définition associée au style d'énoncé -Bouton d'action

Si l'utilisateur demande la définition, le panneau courant est remplacé par le panneau d'explication du type (figure 5.22). Depuis la définition, on peut demander des exemples ou revenir au panneau principal en appuyant sur le bouton OK.

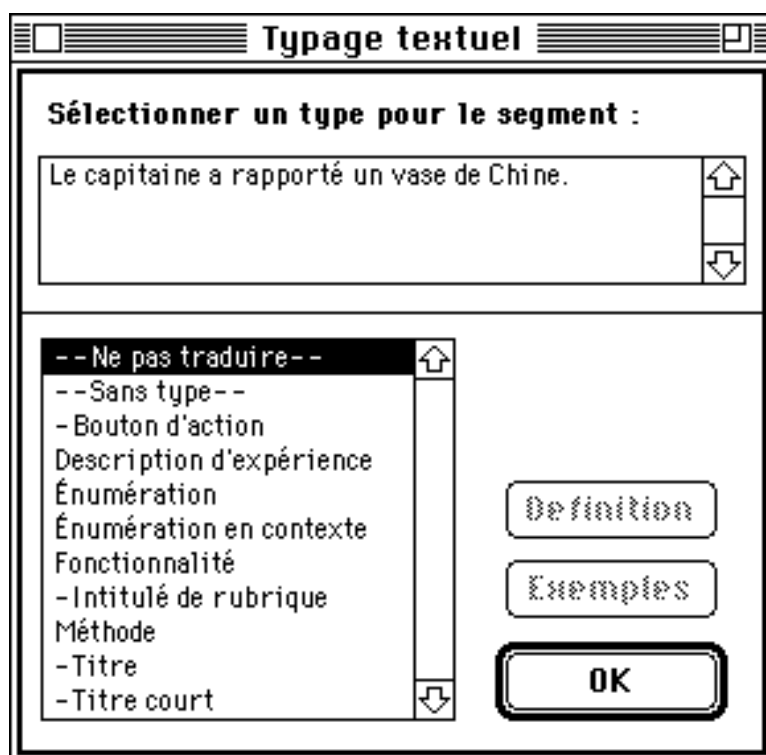


Figure 5.23 : Sélection d'un type textuel non exemplifié



Figure 5.24 : Choix multiple pour le type d'un énoncé

Pour les types --Ne pas traduire-- et --Sans type-- les exemples sont inutiles, le bouton d'accès aux exemples est donc inactif. L'exemple donné dans la figure 5.23 est tiré de notre pile exemple qui présente des types d'ambiguïté en français : il ne faut donc pas traduire les exemples tout en traduisant les autres fragments textuels..

Si l'utilisateur n'est pas certain du type qu'il doit choisir, il peut toujours sélectionner plusieurs types en effectuant des Commande clic. Dans ce cas, on ne peut plus voir de définition ni accéder aux exemples (figure 5.24).

Chapitre 6

Clarification

Lorsque prend fin la phase de standardisation, le système dispose d'informations utiles pour guider l'analyse : une étiquette de type et l'usage des tournures et syntagmes figés. Le serveur LIDIA demande alors l'analyse du texte. Le résultat produit par l'analyseur est examiné afin d'y déceler d'éventuelles ambiguïtés. Si le texte est ambigu, on procède à la phase de clarification.

L'analyseur que nous utilisons calcule toutes les analyses. Il n'effectue que les désambiguïssations certaines résultant des rections fortes et des règles d'accord. Étant donné que chaque unité de traduction est typée, il serait théoriquement possible de définir et d'utiliser des heuristiques de nature symbolique attachées à chaque sous-langage comme on le fait en TA classique pour des typologies de texte bien cernées. Outre le fait qu'il faudrait définir et valider un très grand nombre d'heuristiques, le concept même de TAFD demande de ne pas se limiter à produire la meilleure solution, mais toutes les solutions, si possible ordonnées par vraisemblance.

Dans un système réel, on devrait mettre en œuvre des “semi-heuristiques”, en général statistiques, comme dans [Chiang, *et al.* 1992 ; Cottrell 1989 ; Cowie, *et al.* 1992 ; Doi & Muraki 1992 ; Hirst 1987 ; Kitano 1992 ; Kitano, *et al.* 1989 ; Landau 1990 ; Roche 1992 ; Sano & Fukumoto 1990 ; Sumita, *et al.* 1990 ; Veronis & Ide 1990], produisant une telle liste ordonnée. Pour cette maquette, nous nous bornons à calculer toutes les analyses possibles, sans calculer de préférences.

Afin de définir la phase de clarification, nous avons été conduit à nous intéresser aux divers travaux de classification de l'ambiguïté que nous avons pu recenser. Cette première étape du travail permet de définir une classification qui synthétise diverses approches et a le mérite d'être simple. À partir de cette classification, nous avons choisi les ambiguïtés à traiter dans la maquette et construit une pile de démonstration.

Nous avons défini trois classes et quatre niveaux de dialogue pour résoudre les problèmes auxquels nous nous sommes intéressé. Un seul niveau a été implémenté dans la maquette.

6.1. Ambiguïté et traduction automatique : une classification

Nous proposons ici une classification des ambiguïtés que l'on peut rencontrer lorsque que l'on veut faire de l'analyse de l'écrit ou de l'oral. L'oral est peu détaillé, mais nous avons commencé à nous y intéresser dans le cadre du projet MIDDIM⁴⁴ de désambiguïsation interactive multimodale. Des exemples seront donnés en français, en anglais et en japonais.

Nos principales sources ont été : [Béjoint 1990 ; Carré, *et al.* 1991 ; Fuchs 1985 ; Fuchs 1987 ; Haverkort 1990 ; Hirst 1987 ; Hutchins & Somers 1992 ; Lepage 1985a ; Lepage 1988 ; Nijholt 1988 ; Sadler 1989 ; Schubert 1986 ; Vauquois & Nédobekine 1977].

6.1.1. Ambiguïtés spécifiques de l'oral

a. Homophonie

Les homophones sont des formes ayant le même son et des sens différents.

<u>Exemples :</u>	<u>Remarque :</u>
Les grands <u>mots</u> entraînent toujours de grands <u>maux</u> . (A. France)	mots/maux
[ilprɛ̃ leso]	il prend le seau/le sceau/le sot
[ilparl]	ils parlent/il parle
[japresiE]	j'appréciai/j'appréciais

Le français comporte des règles phonologiques qui distinguent “j'appréciai” de “j'appréciais”, mais, dans la pratique, on observe assez de variations dues à des usages ou des accents régionaux pour lesquels la différenciation n'est pas effectuée.

b. Jointure, segmentation des occurrences

Les ambiguïtés de segmentation sont aussi appelées ambiguïtés de jointure. Dans ce cas, la segmentation d'une chaîne articulatoire en mots, termes ou phrases ne peut être réalisée de manière unique. Pour une étude plus détaillée sur la jointure, on pourra consulter [Wioland 1987].

<u>Exemples :</u>	<u>Remarque :</u>
[tynep÷palavwar]	tu ne peux pas l'avoir/tu ne peux pas la voir.
[ilsɛ̃ trɔ̃zami]	Ils sont treize amis./Ils sont très amis
[lamitoz]	l'avitose/la mitose

⁴⁴ Projet en coopération avec ATR-Interpreting Telecommunication Laboratory Research Labs (Kyoto).

6.1.2. Ambiguïtés de forme

a. Polysémies

La polysémie, c'est le fait pour une forme de présenter plusieurs "sens". La polysémie d'une forme peut relever du fonctionnement normal d'une langue (polysémie interne) ou apparaître à l'occasion d'une étude contrastive, par exemple à des fins de construction d'un dictionnaire bilingue.

i. Polysémie interne (à l'intérieur d'une langue)

Cette classe contient les homographes qui présentent des polysémies pures à l'intérieur d'une même classe morphologique ou d'un même paradigme.

<u>Exemples :</u>	<u>Remarques :</u>
Il admire cette vedette.	vedette : star, bateau ou soldat
Est-ce que la pêche était bonne ?	pêche : fruit, pêcheur
L'aigle quitte son aire.	aire : nid, sa surface de chasse
C'était un vol extraordinairement risqué.	vol plané, vol à la tire
The box was in the pen.	

ii. Polysémie externe (induite par la comparaison avec une autre langue)

<u>Exemples :</u>	<u>Remarques :</u>
"river" en anglais	"rivière" ou "fleuve" en français
"mur" en français	"Wand" ou "Mauer" en allemand ; "muro" ou "parete" en italien.

b. Ambiguïtés de classe syntaxique

C'est le cas où, à une occurrence donnée, on ne sait affecter une classe syntaxique unique, ce qui en général donne un choix entre deux Unités Lexicales différentes. Fuchs nomme ce problème une ambiguïté de syntaxe superficielle.

<u>Exemples avec différentes UL :</u>	<u>Remarques :</u>
Le pilote ferme la porte.	"le pilote porte" ou "le pilote ferme"
La petite brise la glace.	"petit vent" ou "bris de glace"
Devant cette somme, il hésite.	"somme due" ou "somme importante"
Un savant compromis a été arrêté hier.	"savant arrêté" ou "compromis trouvé"

Exemples avec la même UL :

He records records.
La montée qu'il a montée.

c. Ambiguïtés morphologiques

Le problème se pose pour un mot dont le paradigme flexionnel ne permet pas de distinguer certains traits grammaticaux.

<u>Exemples :</u>	<u>Remarques :</u>
L'artiste lit le livre.	un artiste <i>ou</i> une artiste
Je finis subitement.	indicatif : présent <i>ou</i> passé simple
I read.	présent <i>ou</i> passé
He saw the sheep.	"one sheep" <i>ou</i> "several sheep"

6.1.3. Ambiguïtés de structure

Il s'agit des cas où l'on peut regrouper des groupes de mots de différentes façons, toutes cohérentes.

a. Mots composés, tournures

L'ambiguïté structurelle des mots composés et des tournures est surtout évidente dans les langues où l'on procède à leur découpage en éléments plus petits, que ce soit pour des raisons d'économie de la taille des dictionnaires (anglais), de leur flexion interne (partiellement en français) ou bien pour les deux raisons à la fois (allemand).

<u>Exemples :</u>	<u>Remarques :</u>
antichar	contre
antichambre	devant
antimoine	sans décomposition
Il convient avec le gardien de l'ordre du jour.	"gardien de l'ordre" <i>ou</i> "ordre du jour"
Vous sacrifiez l'artiste à la mode que vous pensez indigne.	"artiste à la mode" <i>ou</i> "mode indigne"
a thin can opener	(thin can) opener <i>ou</i> thin (can opener)

b. Structure de constituants grammaticaux

Ce type d'ambiguïté se rapproche de celle des mots composés. Elle n'entraîne pas de mauvaise interprétation de l'ensemble de la phrase, mais seulement des groupes en question (groupes nominaux, en particulier). Elle se divise en deux catégories selon qu'il s'agit de coordination ou de subordination.

i. Coordination

<u>Exemples :</u>	<u>Remarques :</u>
On étudie la structure du réseau et des investissements.	"structure du réseau et structure des investissements" <i>ou</i> "investissements et structure du réseau"
Il prends des crayons et des classeurs noirs.	"crayons noirs et classeurs noirs" <i>ou</i> "classeurs noirs et crayons"

Les neutrons et les protons rapides...

“protons rapides” ou “neutrons rapides et protons rapides”

ii. Subordination

Avec le verbe

Exemples :

Jean rend ce livre illisible.

Le magistrat juge les enfants coupables.

Le capitaine a rapporté un vase de Chine.

Le colonel X a été trouvé pendu par sa femme. (Le Monde, 1964)

Remarques :

Le livre était-il illisible avant ?

“enfants coupables jugés” ou “enfants jugés coupables”

“vase chinois” ou “vase quelconque rapporté de Chine”

“sa femme a pendu le capitaine” ou “sa femme a trouvé le capitaine pendu”

Sans le verbe

Exemples :

L'école de cuisine lyonnaise est ouverte.

Elle épouse un professeur de droit anglais.

Remarques :

“école lyonnaise” ou “cuisine lyonnaise”

“professeur anglais” ou “droit anglais”

6.1.4. Ambiguïtés de relations logiques et sémantiques

Les ambiguïtés de structure relatives aux constituants logiques proviennent le plus souvent des ambiguïtés morphologiques non résolues par la construction des groupes simples de la phrase. Elles sont surtout fréquentes dans les langues où l'ordre des mots n'est pas imposé pour chaque terme logique. Ce type d'ambiguïté est appelé ambiguïté fonctionnelle par Sadler [Sadler 1989].

Exemples :

Pierre fait manger les poulets.

Jean a acheté un livre à Pierre.

Pierre fait porter des chocolats à Lucie.

Je vote pour ma mère.

Je vous parle de la Tour Eiffel.

Quel auteur cite ce conférencier ?

J'ignore quels ennemis redoutaient les soldats.

Remarques :

les poulets mangent ou sont mangés ?

Pierre est vendeur ou un livre pour Pierre

Lucie porte ou reçoit les chocolats

en faveur ou à la place de ma mère

depuis ou à propos de la Tour Eiffel

un auteur cite ou un conférencier cite

“les soldats redoutent des ennemis” ou “les ennemis redoutent les soldats”

6.1.5. Ambiguïtés anaphoriques

Le terme “Anaphore” est utilisé ici pour les reprises pronominales et les ellipses.

a. Reprise pronominale ou adjectivale

Quel est l'élément repris ?

Exemples :

Les médecins ont soigné les malades, ils sont très contents.

La fermière vend sa vache parce qu'elle est malade

Remarques :

“les malades sont contents” *ou*
“les médecins sont contents”

“la vache est malade” *ou*
“la fermière est malade”

b. Ellipse

Il y a ellipse quand un terme de phrase ou de groupe n'est pas repris. Elle peut être simple, si un seul mot ou un groupe de mots a disparu, ou complexe, si plusieurs groupes ont disparu.

i. Simple

Exemples :

Elle aime son fils plus que son mari.

Il traite son prochain comme un ennemi.

Lorsqu'une belle lune met fin à une morne journée, la suivante sera plus belle.

Remarques :

“Elle aime plus son fils qu'elle n'aime son mari.” *ou*
“Son mari aime moins son fils qu'elle n'aime ce fils.”

“... comme si c'était un ennemi.” *ou*
“... comme un ennemi traite son prochain.”

“la lune suivante sera plus belle” *ou*
“la journée suivante sera plus belle”

ii. Sur un mot composé

Cette forme d'ellipse est un cas particulier de polysémie. En effet, lorsque l'on ne reprend pas complètement un terme composé pour alléger ou abréger le texte, le terme est souvent remplacé par son gouverneur. Le gouverneur est alors l'ultime déterminé et devient une occurrence polysémique.

Exemples

Entretien de la centrale inertielle. Démonter puis déposer la centrale. Vérifier le guide d'amorce de l'échangeur. Si nécessaire, changer le guide et remonter la centrale.

Un micro-ordinateur portable Æ un portable

Une phrase transitive Æ une transitive

Un roman de la série noire Æ une série noire

6.1.6. Autres ambiguïtés

a. Détermination, Généricité

Exemples :

Elle veut épouser un milliardaire.

Jean doit aller à l'hôpital.

Remarques :

“Jean qui est milliardaire” *ou* “n'importe qui pourvu qu'il soit milliardaire”

“cet hôpital précis” *ou* “un hôpital quelconque”

b. Portée des quantificateurs

Exemples :

Un fleuve traverse chaque pays européen. “des fleuves différents traversent chaque pays, un par pays”
ou “un même fleuve traverse tous les pays”.

Remarques :

On peut noter que ces deux derniers types d’ambiguïtés peuvent souvent être conservées en traduction. Mais il y a des cas où elles doivent être résolues.

Exemples :

la détermination pour le russe Æ français ou le japonais Æ français,
l’aspect pour le français Æ japonais.

6.2. Ambiguïtés retenues

Pour la maquette LIDIA-1, nous nous concentrons sur les problèmes qui sont mis en valeur par les calculs linguistiques effectués par notre analyseur. On peut ainsi proposer une méthode de résolution effective. Les ambiguïtés retenues forment un ensemble cohérent, et très représentatif des problèmes que l’on rencontre effectivement.

En vue de proposer une méthode de résolution pour chaque problème, nous avons réuni un corpus d’exemples. Ce corpus ne vise pas à être représentatif des phrases de la langue, mais des constructions admises. Les “phrases de linguistes” que nous avons retenues sont choisies parce qu’elles permettent de détecter les problèmes facilement. Nous avons choisi ces exemples dans un souci pédagogique en ne perdant jamais de vue que le processus de désambiguïsation devrait s’appliquer à des textes réels. La formalisation que nous effectuons au chapitre 9 aidera à s’en convaincre.

À des fins de démonstration de la maquette, nous avons intégré ce corpus dans un ensemble de documents HyperCard, dans lesquels les phrases exemples sont placées dans différents contextes, suggérant à un humain des désambiguïsations différentes, impossibles à obtenir par des techniques usuelles (symboliques ou statistiques).

6.2.1. Corpus

Parmi la classification précédente, nous avons retenu un certain nombre d’ambiguïtés et choisi un corpus.

a. Ambiguïté de forme

i. polysémie interne et contrastive

ii. ambiguïté de classe syntaxique

Corpus :

Remarques :

Le pilote ferme la porte.	cf. § 6.1.2.b.
Devant cette somme, il hésite.	cf. § 6.1.2.b.
Un savant compromis a été arrêté hier.	cf. § 6.1.2.b.
Il atteint la grange et la ferme.	“fermer la grange” ou “atteindre la grange”

Pierre prend la boule et la lance.	“il prend la boule” <i>ou</i> “il lance la boule”
L’ouvrier creuse la chaussée et l’asphalte.	“l’ouvrier creuse l’asphalte” <i>ou</i> “l’ouvrier asphalte la chaussée”
Il observe la photo et la classe.	“il observe la classe” <i>ou</i> “il classe la photo”
L’artiste considère le volume qui contient la pièce et le lit.	“l’artiste considère le lit” <i>ou</i> “l’artiste lit le volume”

b. Ambiguïté de structure

i. Mots composés, expressions

<u>Corpus :</u>	<u>Remarques :</u>
Il parle avec le gardien de l’ordre du jour.	“gardien de l’ordre” <i>ou</i> “ordre du jour”
Vous sacrifiez l’artiste à la mode que vous pensez indigne.	cf. § 6.1.3.a.

ii. Structure de constituants grammaticaux

Coordination

<u>Corpus :</u>	<u>Remarques :</u>
On étudie l’évolution de la structure du réseau et des investissements.	3 interprétations
L’évolution de la structure du réseau et des investissements est étudiée.	2 interprétations, cf. figure 9.4
Il prend des cahiers et des classeurs noirs.	cf. § 6.1.3.b.i.
La directrice et les capitaines ruinés sont coupables.	“capitaines ruinés” <i>ou</i> “directrices ruinées et capitaines ruinés”
La grange et les ponts défraîchis sont noirs.	“pont défraîchis” <i>ou</i> “grange défraîchie et ponts défraîchis”
La directrice et le magistrat lyonnais lisent un livre de droit anglais.	“magistrat lyonnais” <i>ou</i> “directrice lyonnaise et magistrat lyonnais”

Subordination, avec le verbe, sans le verbe

<u>Corpus :</u>	<u>Remarques :</u>
Elle épouse un professeur de droit anglais.	cf. § 6.1.3.b.ii.
L’école de cuisine lyonnaise ouvre ses portes.	cf. § 6.1.3.b.ii.
Marie voit l’homme dans le parc avec un télescope.	5 interprétations
Pierre a vu la grange dans les bois près du ruisseau.	même modèle que la phrase précédente

La directrice de la société est ruinée.	“directrice ruinée” <i>ou</i> “société ruinée”
Le magistrat juge les enfants coupables.	cf. § 6.1.3.b.ii.
Le capitaine a rapporté un vase de Chine.	cf. § 6.1.3.b.ii.
Il parle de l'école de cuisine.	“il parle depuis l'école de cuisine” <i>ou</i> “il parle à propos de l'école de cuisine” <i>ou</i> “il parle depuis l'école à propos de cuisine” cf. figure 9.3

c. Relation logique et relation sémantique

<u>Corpus :</u>	<u>Remarque :</u>
Jean a acheté un livre à Pierre.	cf. § 6.1.4.
Pierre fait porter des chocolats à Lucie.	cf. § 6.1.4. et figure 9.7
Je vote pour ma mère.	cf. § 6.1.4.
Je vous parle de la Tour Eiffel.	cf.. § 6.1.4. et figure 9.8
Quel auteur cite ce conférencier ?	cf. § 6.1.4. et figure 9.10
Quel hôtel gère ce bureau ?	“un hôtel gère un bureau” <i>ou</i> “un bureau gère un hôtel”

d. Croisements

<u>Corpus :</u>	<u>Remarques :</u>
Il écrit sur le pont du navire.	5 interprétations
Jean parle de l'école de cuisine lyonnaise.	5 interprétations

6.2.2. Pile de démonstration - vue externe

Nous avons construit une pile de démonstration dans laquelle nous avons inséré les phrases du corpus dans de petites histoires. Pour que ces histoires conservent un sens après le processus de désambiguïsation, il est nécessaire que l'utilisateur accepte d'effectuer la phase de clarification. La pile contient au moins deux histoires par phrase du corpus.

La pile obtenue possède des cartes sur lesquelles sont rassemblées les histoires qui intègrent une même phrase du corpus (figure 6.1). Depuis ce type de carte, on peut atteindre grâce aux boutons *Traitement* une carte sur laquelle on voit la phrase partagée avec ou sans son contexte (figures 6.2 et 6.3. C'est sur ce type de carte que l'on effectue les traitements LIDIA.

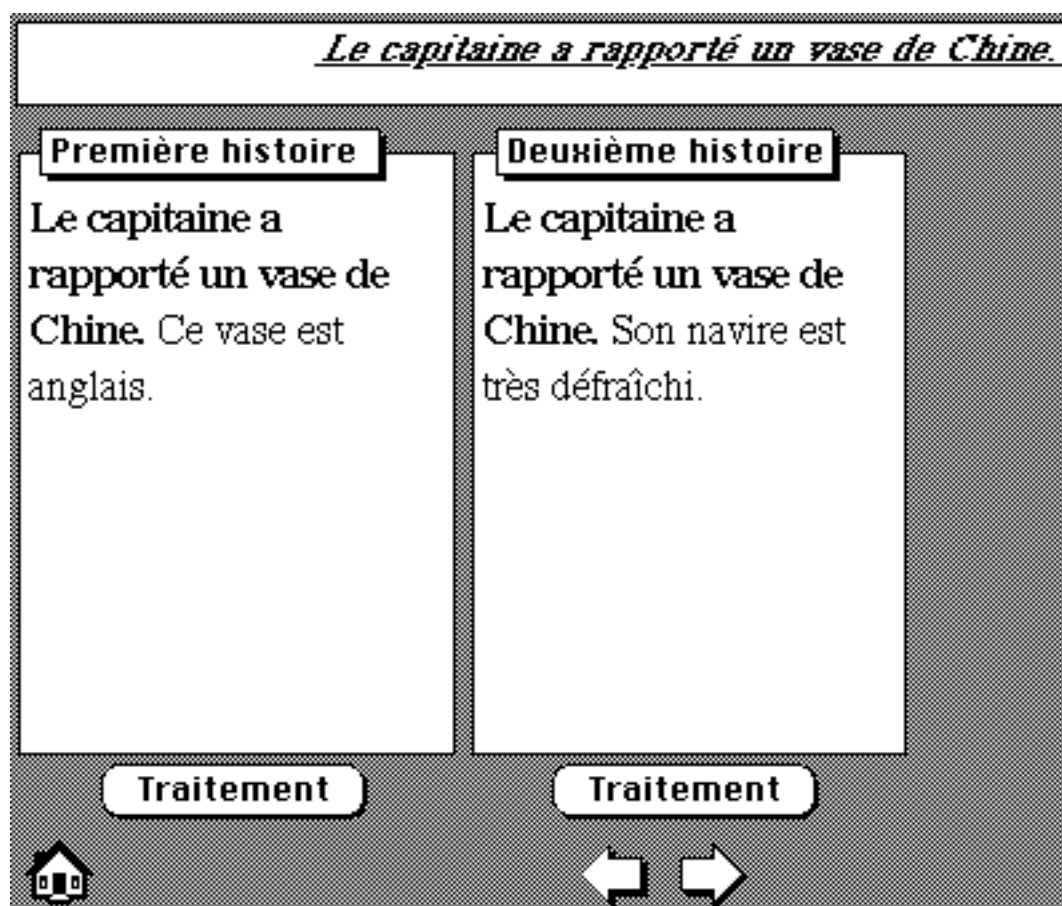


Figure 6.1 : Carte d'histoires relatives à une phrase du corpus LIDIA-1



Figure 6.2 : Carte de traitement d'une histoire - sans contexte

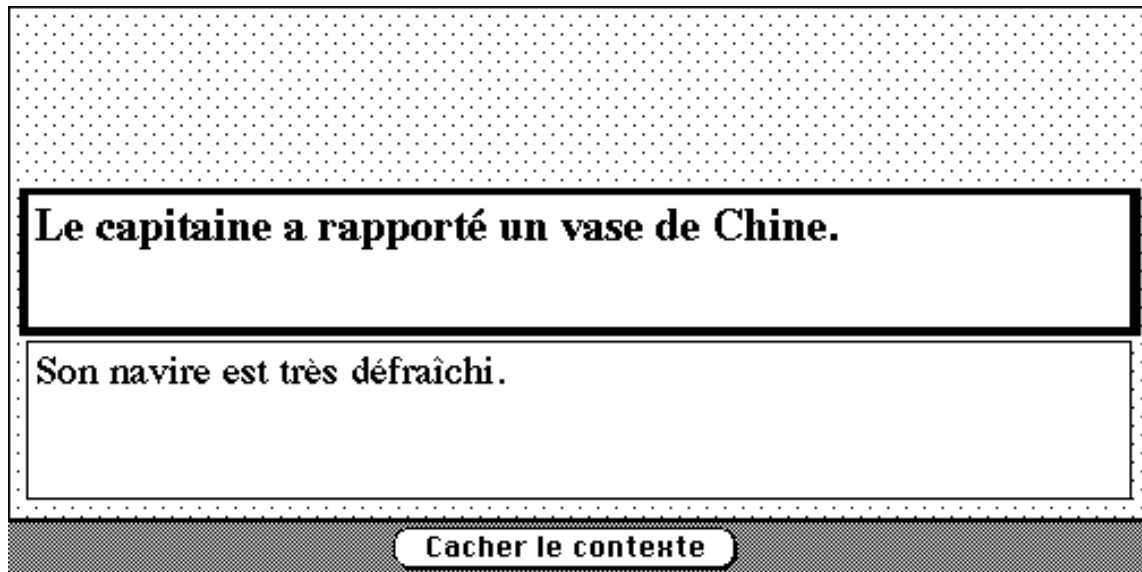


Figure 6.3 : Carte de traitement d'une histoire - avec contexte

6.2.3. Pile de démonstration - vue interne

Cette pile de démonstration a été construite en respectant les contraintes de traductibilité que nous avons évoquées au § 4.1.3.a. Voici comment nous avons procédé :

Les champs de texte des cartes d'histoires (figure 9.3) sont remplis par des scripts à partir des champs contenus dans les cartes de traitements (figures 9.4 & 9.5). Pendant la phase de traduction, les champs de contexte sont visibles.

Le bouton de gestion du contexte change de nom. Ce nom est donc instancié en prenant sa valeur dans un champs textuel. Voici le script de ce bouton :

```
on mouseUp
  if the short name of me = cd fld "voir-context" of cd "receptacle"
  then
    set the name of me to cd fld "cach-context" of cd "receptacle"
    [...]
  else
    set the name of me to cd fld "voir-context" of cd "receptacle"
    [...]
  end if
end mouseUp
```

6.3. Scénario et dialogues

La phase de clarification se déroule en deux étapes. Dans un premier temps on cherche à retenir une seule des solutions produites par l'analyseur. Pour cela, on doit trouver la classe syntaxique de chaque occurrence, la géométrie de l'analyse et les décorations des nœuds de l'arbre. Dans un deuxième temps, on résout les ambiguïtés dues aux polysémies.

Cet ordre nous semble plus pertinent que l'ordre inverse, car le degré de polysémie des occurrences diminue en fonction de la quantité d'information dont on dispose. Les deux exemples qui suivent illustrent notre point de vue.

i. Un savant compromis a été arrêté hier.

Dans cette phrase, on ne sait pas décider a priori à quelle classe syntaxique appartiennent les occurrences "savant" et "compromis". Si "savant" est un nom, "compromis" est un adjectif, et inversement. Si on posait une question avant de connaître la classe de l'occurrence, il faudrait retenir les sens du nom et les sens de l'adjectif, cela ne nous semble pas être une bonne solution.

ii. Quel hôtel gère ce bureau ?

Dans cette phrase, l'occurrence "bureau" est polysémique. Pour poser une question sur le sens de "bureau" avant l'analyse, on ne pourrait pas exclure le sens "meuble" de "bureau". Par contre, lorsqu'on pose la question après la phase d'analyse, le sens "meuble" n'apparaît pas dans la question.

6.3.1. Scénario pour la clarification

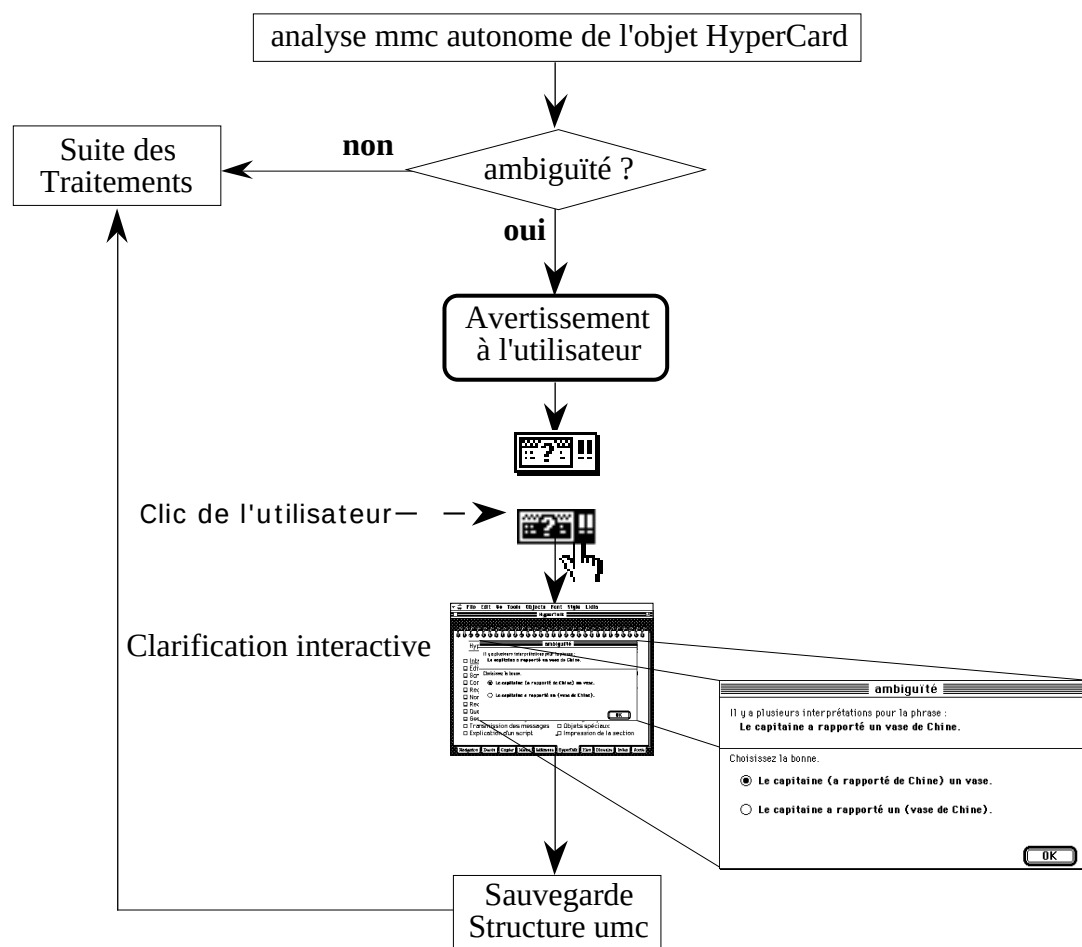


Figure 6.4 : Scénario pour la clarification

La structure mmc est examinée par le module de désambiguïsation qui construit, si nécessaire, un arbre de questions. Le système demande alors à l'utilisateur d'intervenir en mettant une nouvelle entrée dans le menu Messages et en posant si nécessaire un bouton sur l'objet concerné.

L'utilisateur accepte le dialogue en cliquant sur le bouton  et passe ainsi en désambiguïsation interactive. La première étape de cette phase permet de sélectionner une seule structure, contenant éventuellement des polysémies codées dans certains nœuds par l'attribut sens. La seconde étape consiste à réduire ces polysémies. À la fin, on obtient la structure désirée. Ce processus est schématisé sur la figure 6.4, page précédente.

6.3.2. Classes de dialogue

Les dialogues de clarification procèdent par complétion (ajout d'information), paraphrasage, ou remplacement.

a. Complétion

Dans un dialogue par complétion, on ajoute de l'information dans la phrase qui pose problème. Ce type de dialogue est utilisé pour résoudre les ambiguïtés de classes syntaxiques et les ambiguïtés de polysémie. Des tels dialogues sont donnés figures 9.23 et 9.24.

b. Paraphrasage

Lorsqu'il faut reconstituer une éliision, nous procédons par paraphrasage en reconstituant le ou les éléments élidés. Dans ce cas, nous pouvons être conduits à mettre en œuvre un processus de pseudo-génération syntaxique pour produire une autre forme fléchiée d'un adjectif du texte.

Dans d'autres cas, une simple permutation d'occurrences suffit pour montrer les diverses interprétations d'un énoncé. Nous utiliserons alors des parenthèses car la forme de surface peut ne pas être transformée par l'interprétation. Un tel dialogue est donné figure 9.22.

c. Remplacement

Lorsque la complétion et le paraphrasage ne permettent pas de produire le dialogue de désambiguïsation, nous procédons par remplacement sémantique en utilisant les informations fournies par Parax, la base lexicale multilingue par acception. Dans la maquette, le remplacement concerne les prépositions. Un tel dialogue est donné figure 9.25.

6.3.3. Types de dialogue

En nous inspirant des différents niveaux d'assistance qui sont prévus dans le système CRITIQUE [Heidorn, *et al.* 1982 ; Jensen, *et al.* 1993 ; Lepage 1985b ; Richardson 1985], nous imaginons quatre niveaux de dialogue pour un système réel. Dans la maquette, nous n'avons implémenté que le second (cf. le paragraphe b. suivant). Pour la polysémie, nous avons envisagé un mode de traitement de la répétition lexicale dont l'implémentation ne présente pas de difficulté.

a. Avec explications

Le dialogue sans explication s'adresse à un utilisateur qui souhaite apprendre à reconnaître les cas de figure qui forment un problème pour le système lors de l'analyse.

ambiguïté

Il y a plusieurs interprétations pour la phrase :
Un savant compromis a été arrêté hier.

Choisissez la bonne.

☒ **Un savant (nom) compromis (adjectif) a été arrêté hier.**

☐ **Un savant (adjectif) compromis (nom) a été arrêté hier.**

Explications **OK**

Figure 6.5 : Dialogue avec complétion et accès à des explications

ambiguïté

La phrase : Un savant compromis a été arrêté hier.

Comporte une ambiguïté de classe syntaxique

Une forme peut correspondre à des mots ayant des classes syntaxiques différentes (savant : nom commun ou adjectif).
Lorsque des formes qui se succèdent dans la phrase ont cette propriété, il est parfois possible que plusieurs combinaisons donnent une analyse admissible par l'analyseur.
C'est le cas ici où l'on obtient les successions :
- article (un) nom (savant) adjectif (compromis) / une maison rouge
- article (un) adjectif (savant) nom (compromis) / un beau vêtement

Dialogue **OK**

Figure 6.6 : Explications pour l'ambiguïté de classe syntaxique

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Il prend les cahiers et les classeurs noirs.	
Choisissez la bonne.	
☒	Il prend les cahiers et (les classeurs noirs).
☐	Il prend les cahiers noirs et les classeurs noirs.
Explications OK	

Figure 6.7 : Dialogue avec paraphrasage et accès à des explications

ambiguïté	
La phrase :	<u>Il prend les cahiers et les classeurs noirs.</u>
Comporte une ambiguïté de coordination adjectivale	
À cause des règles d'accord de l'adjectif qualificatif, l'attachement de celui-ci peut être ambigu. Il est des cas où l'attachement n'est pas ambigu : (1) "le(s) volet(s) et la <u>fenêtre fermée</u>" (2) "le(s) volet(s) et les <u>fenêtres fermées</u>" (3) "la (les) <u>fenêtre(s)</u> et le <u>volet fermé</u>" (4) "le <u>volet</u> et la <u>fenêtre fermés</u>" Ce n'est le cas que si l'accord de l'adjectif n'est compatible qu'avec le dernier nom (1, 2, 3) ou si le dernier nom est féminin singulier et l'adjectif masculin pluriel (4). Dans tous les autres cas, l'attachement est ambigu.	
Dialogue OK	

Figure 6.8 : Explications pour l'ambiguïté de subordination adjectivale

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Je vous parle de la Tour Eiffel.	
Choisissez la bonne.	
☒	Je vous parle depuis la Tour Eiffel.
☐	Je vous parle à propos la Tour Eiffel.
Explications OK	

Figure 6.9 : Dialogue avec remplacement sémantique et accès à des explications

ambiguïté	
La phrase :	<u>Je vous parle de la Tour Eiffel.</u>
Comporte une ambiguïté de construction argumentaire du verbe	
Une préposition peut souvent introduire différents types de compléments. Par exemple : "parler de x" : parler depuis x, parler à propos de x. parler de la tribune à la foule rassemblée, parler de la peinture sur soie, parler de la cuisine, depuis la cuisine, à propos de la cuisine ? "voter pour" : à la place de, en faveur de voter pour son frère qui est en voyage, voter pour un bon candidat, voter pour sa mère ? "acheter x à y" : x qui appartient à y, x pour y acheter un vieux meuble à l'antiquaire, acheter une bague à sa fiancée, acheter un livre à Paul ?	
Dialogue OK	

Figure 6.10 : Explications pour l'ambiguïté de construction argumentaire du verbe

b. Sans explication

Le dialogue sans explication présente les items produits par le serveur de désambiguïsation. C'est le type de dialogue que nous avons implémenté actuellement.

c. Pour expert non linguiste

Le dialogue pour expert non linguiste permet à l'utilisateur de se familiariser avec les informations calculées par l'analyseur. On met en correspondance la reformulation du problème avec la représentation du système. Ce type de dialogue doit permettre à l'utilisateur d'acquérir une connaissance du système suffisante pour qu'il puisse insérer des annotations dans les textes qu'il rédige.

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Un savant compromis a été arrêté hier.	
Choisissez la bonne.	
☒	Un savant (nom) compromis (adjectif) a été arrêté hier. Un savant.&n compromis.&adj a été arrêté hier.
☐	Un savant (adjectif) compromis (nom) a été arrêté hier. Un savant.&adj compromis.&n a été arrêté hier.
OK	

Figure 6.11 : Dialogue pour expert non linguiste – 1

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Il prend les cahiers et les classeurs noirs.	
Choisissez la bonne.	
☒	Il prend les cahiers et (les classeurs noirs). Il prend les cahiers et {les classeurs noirs}.
☐	Il prend les cahiers noirs et les classeurs noirs. Il prend {les cahiers et les classeurs} noirs.
OK	

Figure 6.12 : Dialogue pour expert non linguiste – 2

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Je vous parle de la Tour Eiffel.	
Choisissez la bonne.	
☒	Je vous parle depuis la Tour Eiffel. Je vous parle {de la Tour Eiffel}.&circ.
☐	Je vous parle à propos de la Tour Eiffel. Je vous parle {de la Tour Eiffel}.&obj 1.
OK	

Figure 6.13 : Dialogue pour expert non linguiste – 3

Bilan

Dans un système tel que LIDIA, l'interface est une composante essentielle. C'est pourquoi nous avons consulté un grand nombre de références bibliographiques sur le sujet. Les références les plus utiles pour notre travail ont été [Apple Computer 1989 ; Apple Computer 1987 ; Carrol, *et al.* 1991 ; Chignell & Hancock 1991 ; Coutaz 1988 ; Egan 1991 ; Gould 1991 ; Heckel 1991 ; Hix & Schulman 1991 ; Karat, *et al.* 1986 ; Landauer 1991 ; Lewis 1990 ; Paap 1991 ; Perlman 1991 ; Roberts 1991 ; Streeter 1991 ; Sullivan & Tyler 1991 ; Wærn 1990].

Pour cette maquette, nous nous sommes volontairement limités à des dialogues par menus, ne mettant en œuvre aucune véritable génération de texte, aucune manipulation graphique comme on en trouve dans KBMT-89 ou dans JETS, et aucune synthèse ou reconnaissance de la parole. Toute une étude reste à faire sur la désambiguïsation interactive multimodale dans le cadre de la traduction et de l'interprétation.

Nous sommes aussi très conscients de certaines imperfections de notre interface. Ainsi, le point le plus faible nous semble être l'accès à la correction orthographique. Si nous faisons une seconde version de la maquette, il faudra absolument maîtriser l'appel du moteur de correction. Pour un prototype, les modules de traitement de la terminologie et des tournures devront intégrer au moins un lemmatiseur.

L'interface des modules concernant la terminologie, les tournures et le typage n'est qu'une première ébauche. Nous n'avons, en particulier, pas encore trouvé d'icônes satisfaisantes pour tous les boutons. Par contre, les icônes des curseurs LIDIA sont homogènes avec les boutons d'information, et pourraient être réutilisées telles quelles.

Dans une version ultérieure, il faudra enfin implémenter tous les niveaux de dialogues présentés dans ce mémoire de façon à ce que l'auteur puisse réellement "découvrir" le système et, à partir d'un certain niveau de

compétence, mettre lui-même les marques. Peut-être pourra-t-on aussi introduire de nouvelles formes utilisant la voix et les graphiques.

Troisième partie

La maquette LIDIA-1 – vue interne

Introduction

La partie précédente concernait surtout les aspects ergonomiques (interface de LIDIA et de ses services) et linguistiques (ambiguïté) de notre maquette. L'aspect informatique, et plus particulièrement les problèmes de communication et le traitement algorithmique par objets, n'est pas moins intéressant.

L'architecture de la maquette s'organise autour de trois serveurs : le serveur LIDIA, le serveur de communication, et le serveur de traduction. Le serveur LIDIA supervise l'ordonnancement du processus de traduction. Le serveur de communication règle les échanges entre le serveur LIDIA et le serveur de traduction. Le serveur de traduction assure la réalisation des traitements linguistiques. Ces trois serveurs sont en principe hébergés par des machines physiques différentes, bien que les deux premiers puissent tourner sur le même Macintosh.

Comme dans tout système distribué, les messages jouent un grand rôle. Dans LIDIA, toutes les composantes communiquent avec les autres et avec elles-mêmes au moyen de messages. Les appels directs de procédure sont évités autant que possible. Les données relatives au processus de traduction sont stockées dans des fichiers que seul le serveur LIDIA peut modifier. Cette organisation donne une grande flexibilité au niveau de l'implémentation, et une composante peut facilement être déportée.

Le désambigüiseur est lui aussi implémenté comme un serveur. Le noyau de ce serveur est un processeur de filtrage. En effet, une grande partie du processus de clarification est fondée sur un mécanisme de reconnaissance de formes utilisant des "patrons". Un patron décrit une classe d'ambiguïtés. À chaque patron est associée une méthode de production du dialogue réalisée au moyen d'un certain nombre d'opérateurs de base. L'ordre dans lequel on cherche à reconnaître les patrons est fixé par une stratégie globale.

Chapitre 7

Architecture

Le serveur LIDIA étant intégré à la station de rédaction, et plus particulièrement à HyperCard, nous présentons ces deux aspects dans la première partie. Les deux suivantes sont consacrées au serveur de communication et au serveur de traduction.

7.1. Station de rédaction

La station de travail du rédacteur héberge trois applications, HyperCard, Parax et le serveur LIDIA.

HyperCard est l'environnement de rédaction et sert de frontal au serveur LIDIA. Parax est la maquette d'une base lexicale par acception [Sérasset & Blanc 1993], réalisée en HyperCard. Le serveur LIDIA constitue le noyau du système complet.

7.1.1. HyperCard

Nous avons donné une vue externe de HyperCard dans le paragraphe 4.1.2.a. Pour comprendre l'implémentation du système, nous avons besoin de décrire certains mécanismes internes.

a. HyperCard vue interne

Les objets HyperCard interagissent entre eux, avec l'utilisateur, avec HyperCard et avec l'environnement Macintosh en envoyant des messages.

Certains messages décrivent des phénomènes ou des actions qui se produisent dans l'environnement. Par exemple, le bouton de la souris a été enfoncé, une carte a été ouverte, etc. Ces messages sont les messages système. Par exemple, si l'on enfonce le bouton de la souris, HyperCard envoie le message `mouseDown`. Lorsqu'on le relâche, HyperCard envoie le message `mouseUp`.

Les messages sont transmis aux différents objets dans l'ordre défini figure 7.1.

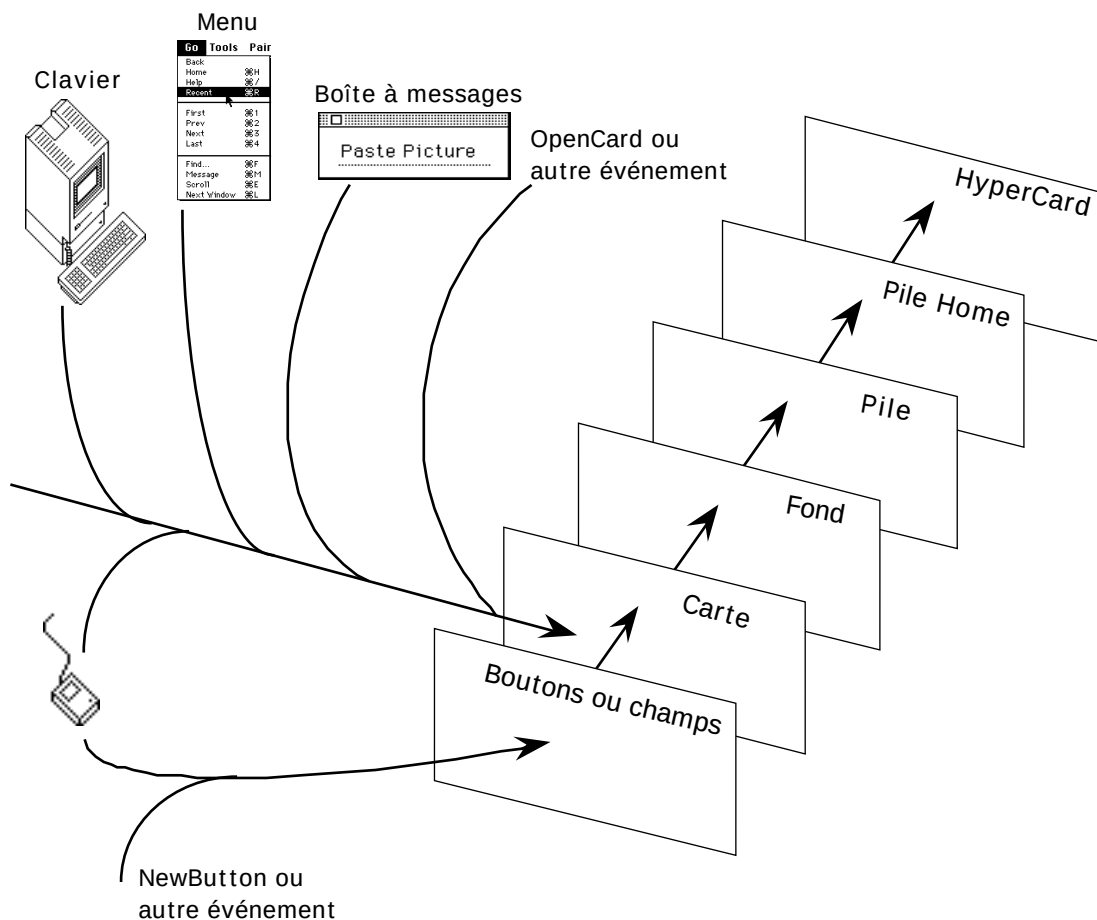


Figure 7.1 : Hiérarchie par défaut de transmission des messages dans HyperCard

Les commandes HyperTalk⁴⁵ sont aussi des messages.

Chaque objet HyperCard possède un *script*. Un script est une collection de *handlers*. Un handler est une collection d'instructions HyperTalk. Un handler peut être une procédure destinée à répondre à un message, ou une fonction rendant un résultat. Dans la suite, un handler du premier type est appelé un processeur de message. Un processeur de message est invoqué lorsque l'objet dans le script duquel il se trouve reçoit le message que le processeur traite. Le processeur de message est destiné à traiter le message `mouseUp`.

```
on mouseUp
go to next card
end mouseUp
```

Un message parcourt la hiérarchie des objets jusqu'à ce qu'un processeur de message soit en mesure d'y répondre. S'il ne rencontre aucun processeur de message qui puisse le traiter, il est perdu et n'a aucun effet.

L'utilisateur peut changer cette hiérarchie en insérant une ou plusieurs piles entre la pile courante et la pile Home. Ces piles insérées servent de bibliothèque de processeurs de message.

⁴⁵ HyperTalk est le langage de script de Hypercard [Claris 1991].

b. Commandes et fonctions externes

Les commandes et fonctions externes sont des extensions aux commandes et fonctions de base de HyperTalk. HyperCard fournit les procédures d'interface qui permettent une extension aisée et pratique aux programmeurs.

Les commandes externes (XCMDs) et les fonctions externes (XFCNs) sont des ressources codes exécutables écrites dans certains des langages de programmation disponibles pour le Macintosh (C, Pascal, Assembleur, mais pas Lisp). Ces ressources peuvent être attachées à l'application HyperCard ou à n'importe quelle pile.

Une XCFN ou XCMD est appelée avec HyperTalk de la même manière que les commandes natives HyperTalk ou les fonctions définies par l'utilisateur.

c. HyperCard pour LIDIA

Pour l'utilisateur, le comportement externe de HyperCard n'est pas changé. Par contre, nous avons dû modifier la pile Home pour LIDIA. Nous avons ajouté un bouton qui permet d'avoir accès aux traitements LIDIA, nous avons inséré deux piles bibliothèques dans la hiérarchie standard, et nous définissons et initialisons des variables globales spécifiques à LIDIA.

Les deux piles bibliothèques freeware de XFCN et XCMD sont :

- la pile X-Archive 3.1 qui inclut 127 fonctions ou commandes externes de Frederic Rinaldi,
- la pile Dartmouth XCMD's 4.0⁴⁶.

Nous utilisons aussi une pile de bibliothèque personnelle (Services LIDIA) qui sert de réceptacle à divers processeurs de message et commandes externes. Nous avons choisi cette organisation afin de ne pas transformer le logiciel HyperCard. La gestion des menus Lidia et Messages est aussi confiée à la pile Services LIDIA.

7.1.2. Parax⁴⁷

Pour une TAO fortement multilingue telle que visée par la TAFD, il nous semble nécessaire d'ajouter aux représentations multiniveaux un niveau lexical, celui des acceptions interlingues (§ 2.2.1). C'est pourquoi notre laboratoire travaille depuis longtemps sur les dictionnaires multilingues [Boitet & Nédobekine 1986 ; Sérasset 1992a ; Sérasset 1992b ; Sérasset 1992c].

Lancé en 1990, le projet NADIA vise ainsi à la définition et à l'implémentation d'un outil de construction et l'utilisation de bases de données lexicales multilingues, indépendantes d'une théorie linguistique et indépendantes d'une application. Ce système est basé sur l'utilisation d'un dictionnaire central d'acceptions interlingues (sens généralement reconnus d'un mot).

⁴⁶ © 1988-1993 by the trustees of Dartmouth college, Hanover, NH. Ces deux piles, ou leurs versions suivantes, sont disponibles sur [sumex-aim.stanford.edu /info-mac/card](http://sumex-aim.stanford.edu/info-mac/card).

⁴⁷ Pour une description plus précise de Parax, consulter [Sérasset & Blanc 1993],

Parax est une maquette implémentée sous HyperCard par Étienne Blanc pour étudier la faisabilité d'une telle approche. Cette maquette comporte pour l'instant 5 langues : français, anglais, allemand, russe et chinois.

La structure linguistique des dictionnaires monolingues est inspirée des dictionnaires du générateur de systèmes de traduction automatique ARIANE du GETA [Boitet, *et al.* 1982 ; Boitet, *et al.* 1985] Cette structure linguistique est fixe.

Cette maquette illustre le choix de l'interlangue par acceptions. L'utilisateur peut sélectionner un mot d'une des langues de la base. Parax propose alors une liste d'acceptions correspondant au mot choisi. Chaque acception est accompagnée d'une définition en français. Dans une première version, les définitions sont stockées avec les acceptions dans la base interlingue (et non dans chacune des bases monolingues). Elle n'apparaissent donc que dans la langue de référence de la base : le français. La figure 7.2 montre l'entrée française du verbe "fermer" dans la base Parax. On a trois acceptions.

<u>fermer 1,a</u> #fermer_fermeture\$ SENS:1. CAT: vb. AUX: avoir.	#fermer_fermeture\$ appliquer une partie mobile pour boucher un passage, une ouverture (<i>' fermer la porte, les rideaux.'</i>)
<u>fermer 1,b</u> #fermer_lieu SENS:2. CAT: vb. AUX: avoir.	#fermer_lieu priver de communication avec l'extérieur, par la mise en place d'un élément mobile; interdire le passage (<i>' fermer sa chambre, une valise, une route.'</i>)
<u>fermer 1,c</u> #fermer_replier\$ SENS:3. CAT: vb. AUX: avoir.	#fermer_replier\$ rapprocher, réunir (parties d'un organe, éléments d'un objet) de manière à ne pas laisser d'intervalle ou à replier vers l'intérieur, et par ext. mettre cet objet ou cet organe dans cette nouvelle position (<i>'fermer le poing, un canif.'</i>)

Figure 7.2 : Entrée "fermer" dans la base Parax

L'utilisateur peut ensuite choisir l'une des acceptions.

S'il choisit la troisième acception, il découvre un problème contrastif entre le français et l'allemand. Il existe deux acceptions allemandes différentes correspondant à l'acception française sélectionnée : "fermer_replier". Si l'objet du verbe "fermer" est un poing, la traduction allemande est "ballen", sinon, c'est "zuklappen". La figure 7.3 illustre cette situation.

*fermer_replier\$ °AN °FR °RU	
rapprocher, réunir (parties d'un organe, éléments d'un objet) de manière à ne pas laisser d'intervalle ou à replier vers l'intérieur, et par ext. mettre cet objet ou cet organe dans cette nouvelle position ('fermer le poing, un canif) ** SEMPR: activoper. SEM1: produit.	
*fermer_replier\$poing °AL	<u>ballen</u> #fermer_replier\$poing
comme fermer_replier\$, mais	SENS:1.
s'applique particulièrement au poing (AL) **SEMPR: activoper. SEMZ:	CAT: vb. VAL1: acc. AUX: h.
humain. SEM1: humain.	
*fermer_replier\$objet °AL	<u>zuklappen</u> #fermer_replier\$objet
complément de fermer_replier\$1 à fermer_replier\$ ** SEMPR: activoper.	SENS:1.
SEM1: produit.	CAT: vb. VAL1: acc. AUX: h. (al)

Figure 7.3 : Problème contrastif entre le français et l'allemand

Si l'utilisateur veut un équivalent russe, il n'a qu'à choisir le russe comme langue cible. Pour ce faire, il clique sur °RU. Il obtient alors l'écran de la figure 7.4.

*fermer_replier\$ °AN °FR °RU	закрыть #fermer_replier\$
rapprocher, réunir (parties d'un organe, éléments d'un objet) de manière à ne pas laisser d'intervalle ou à replier vers l'intérieur, et par ext. mettre cet objet ou cet organe dans cette nouvelle position ('fermer le poing, un canif) ** SEMPR: activoper. SEM1: produit.	МНЕН:4. КАТ: глгг. ВИД: сов. ДОП1: ВИН.
*fermer_replier\$poing °AL	
comme fermer_replier\$, mais	
s'applique particulièrement au poing (AL) **SEMPR: activoper. SEMZ:	
humain. SEM1: humain.	
*fermer_replier\$objet °AL	
complément de fermer_replier\$1 à fermer_replier\$ ** SEMPR: activoper.	
SEM1: produit.	

Figure 7.4 : Équivalent russe de #fermer_replier

L'écran de Parax, montré ici tronqué, présente dans une colonne de gauche, le terme dans la langue source (comme sur la figure 7.2).

Pour l'instant, Parax comporte 134 lemmes français. Ces lemmes français produisent 526 acceptions ou sous-acceptions qui sont toutes documentées. Pour les langues cibles que nous visons au GETA, les chiffres s'établissent comme suit : 304 lemmes pour l'anglais, 370 lemmes pour l'allemand et 394 lemmes pour le russe. Les chiffres plus importants obtenus pour les langues cibles correspondent au fait qu'un lemme français

peut avoir plusieurs traductions. Cela n'est, en général, pas le cas pour les lemmes des langues cibles car, pour le français, le vocabulaire de la base est réduit à celui du corpus.

7.1.3. Serveur LIDIA

Le serveur LIDIA est le noyau du système. Il détermine et supervise l'ordonnancement des traitements. Ce noyau peut être considéré comme un serveur au vrai sens du terme. Il peut être déporté sur une autre station du réseau et prend lui-même en charge la gestion des fichiers à manipuler.

Ce serveur est réalisé en Macintosh Common Lisp qui est une implémentation du standard Common Lisp et du standard CLOS (Common Lisp Object System). C'est un environnement dynamique et interactif parfaitement intégré au Macintosh.

Les préférences LIDIA associées à un objet définissent un graphe de contrôle qui permet au serveur LIDIA de savoir quels traitements doivent être appliqués à chaque objet. La succession des traitements se fait de façon autonome tant que l'un de ceux-ci ne requiert pas l'intervention de l'utilisateur.

L'étape de correction orthographique est prise en charge par le correcteur lui-même (101 en l'occurrence). Les étapes de typage textuel, de vérification terminologique et de marquage les expressions sont prises en charge par HyperCard (supervisées par le serveur LIDIA). Le développement de ces étapes étant au stade de maquette, la puissance d'expression d'HyperTalk suffit pour monter ce que l'on veut faire. Pour un prototype, il faudrait que ces modules soient autonomes.

L'interaction est décidée par le noyau du système et est matérialisée par HyperCard. Pour les traitements évoqués ci-dessus, MCL demande à HyperCard d'alerter l'utilisateur. Cela passe par la modification du menu message, l'envoi d'un avertissement à l'utilisateur et, si nécessaire par la pose d'un bouton sur l'objet concerné.

Le processus de désambiguïsation est complètement implémenté en MCL, depuis l'étude du résultat produit par l'analyseur jusqu'à la production des dialogues de désambiguïsation.

Pour mener à bien le processus de traduction, le serveur LIDIA communique avec HyperCard, mais aussi avec le serveur de communication auquel il envoie des requêtes. La communication est assurée au moyen d'*Apple Events* et d'*Apple Scripts*, nous y reviendrons dans le chapitre 8.

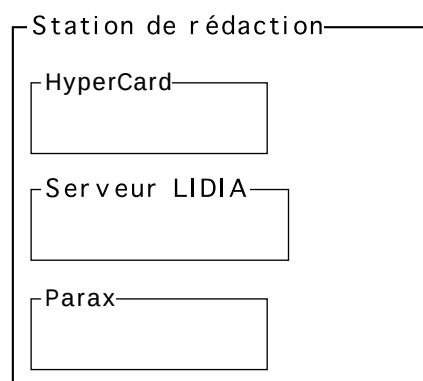


Figure 7.5 : Composants logiciels de la station de rédaction

7.2. Serveur de communication

Pour échanger des fichiers de requête, nous utilisons MacMainFrame, un émulateur de terminal IBM. Avec cette application, on peut échanger des fichiers entre le Macintosh et l'IBM en mode immédiat ou en mode différé. MacMainFrame n'étant pas pilotable par une autre application, nous sommes amenés à utiliser deux fonctions externes pour HyperCard. La première, IBM3270, offre les primitives de base qui permettent d'émuler un terminal, et la seconde IBMXFER, permet de faire des transferts de fichier en mode autonome.

Nous utilisons une version individualisée d'HyperCard (que nous appelons par la suite RemoteMacMainFrame) qui appelle elle-même les deux commandes externes IBM3270 et IBMXFER. RemoteMacMainFrame exécute certaines actions régulièrement grâce à des messages qui lui sont adressés par des scripts.

7.2.1. RemoteMacMainFrame

RemoteMacMainFrame est un HyperCard déguisé. Si l'on souhaite définir une architecture qui reste cohérente, que le serveur de communication soit hébergé par la station de rédaction ou une autre machine, il faut en effet deux programmes physiquement différents.

7.2.2. Scripts

AppleScript est un langage de commandes qui permet d'écrire et d'utiliser des scripts pour contrôler des applications. Ces scripts sont des acteurs indépendants qui peuvent interagir entre eux et avec les applications qui les environnent. AppleScript permet de piloter toute application qui supporte les AppleEvents. Ce langage intègre la notion d'objet et permet de manipuler des objets complexes.

C'est avec des scripts que nous allons piloter RemoteMacMainFrame pour qu'il effectue diverses tâches de manière répétitive.

7.2.3. Serveur

Le serveur de communication se compose de plusieurs entités logicielles, RemoteMacMainFrame et ses scripts.

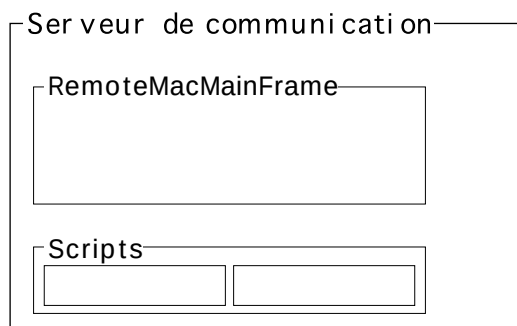


Figure 7.6 : Composants logiciels sur le serveur de communication

Lorsque depuis la station de rédaction, le serveur LIDIA a besoin des services du serveur de traduction, il envoie une requête au serveur de communication. Le serveur de communication transfère le fichier miroir qui est associé à l'objet qui réclame un

traitement. Il construit un fichier requête et transmet ce fichier ainsi que la demande de traitement au serveur de traduction. Le serveur de communication attend ensuite le fichier résultat associé à la requête. Lorsque ce fichier est prêt, il construit un nouveau fichier miroir qui est transféré sur la station de rédaction. Le serveur de communication avertit ensuite le contrôleur LIDIA que le travail a été effectué et que le résultat est disponible.

7.3. Serveur de traduction

On accède aux traitements linguistiques via le réseau LIDIA/IBM qui rend transparent l'usage de Ariane-G5.

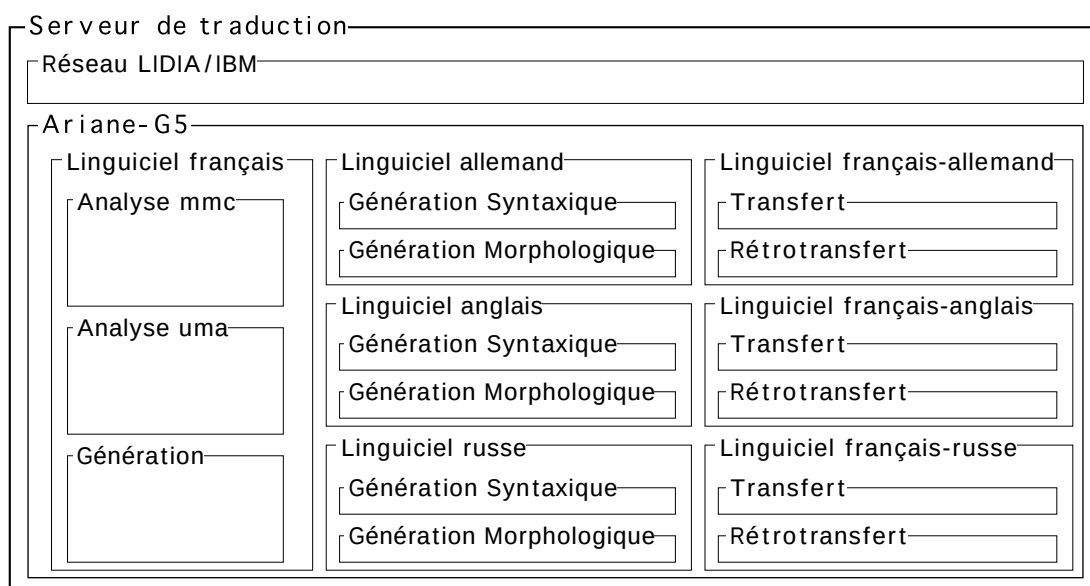


Figure 7.7 : Composants logiciels sur le serveur de traduction

Les travaux sur le serveur de traduction ont été effectués principalement par P. Guillaume [Guillaume 1993] et M. Quézel-Ambrunaz [Quézel-Ambrunaz 1993]. J'ai travaillé avec eux à la définition du fichier requête à échanger entre le serveur de communication et le serveur de traduction, ainsi qu'aux spécifications des fonctionnalités de ce que nous appelons le réseau LIDIA/IBM.

Le linguiciel en Ariane-G5 a été écrit par É. Blanc, J.-P. Guilbaud et N. Nédobéjkine. Avec J.-P. Guilbaud, nous avons travaillé à la définition de la forme de la structure produite, à la définition du système d'annotations que reconnaît l'analyseur et au choix des phrases du corpus. Ch. Boitet, nous a donné des idées de présentation de la pile de démonstration qui a été construite avec M. Beltran et J. Gaschler.

7.3.1. Réseau LIDIA/IBM

Le serveur de traduction est un IBM 4361 exploité sous VM/ESA (370), donc offrant le contexte de machines virtuelles. Ces machines virtuelles sont exploitées sous CMS.

On appelle réseau LIDIA la partie logicielle prenant en charge les communications entre les différentes machines virtuelles utilisées pour les différentes phases écrites en Ariane-G5. Le serveur est lui-même organisé comme un réseau de machines virtuelles fonctionnant en temps partagé sous VM/ESA. Ce réseau développé au GETA est appelé réseau LIDIA/IBM bien qu'il ne soit fonctionnellement pas lié à l'organisation LIDIA.

a. Machines virtuelles

Le réseau LIDIA/IBM intègre trois types de machines virtuelles.

i. Les machines interfaces

Les machines interface permettent les connexions entre l'IBM et un Macintosh. En principe, chaque machine interface est liée à un accès déterminé, donc à un Mac équipé d'une carte adaptateur, et essentiellement utilisée pour les échanges entre ce Mac et l'IBM. C'est donc un point d'entrée et de sortie pour les traitements sur le réseau IBM.

Les échanges dans le sens IBM $\rightarrow$ Macintosh sont gérés par la prise en compte de la file d'attente constituée sur le lecteur virtuel d'une machine interface et la constitution d'un fichier dit "d'interface". Les échanges dans le sens Macintosh $\rightarrow$ IBM, sont gérés par réception d'un fichier d'interface conforme aux conventions établies et transfert vers une machine nœud du réseau. Ces échanges nécessitent l'échange d'un fichier interface dans le sens Macintosh $\rightarrow$ IBM (upload) ou IBM $\rightarrow$ Macintosh (download).

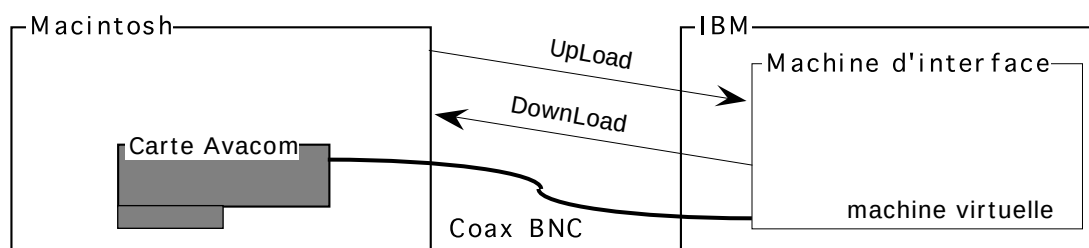


Figure 7.8 : Le serveur de communication et une machine interface

ii. Les machines serveurs

Les machines virtuelles "serveurs" exécutent des parties prédéterminées du processus de traduction que nous appelons "sous-chaînes de traduction".

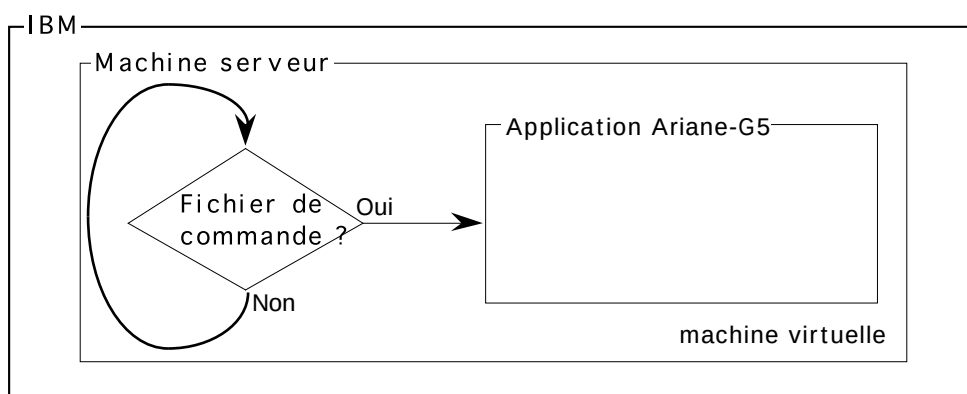


Figure 7.9 : Une machine serveur

Une machine serveur consulte périodiquement son lecteur virtuel. S'il n'est pas vide, elle exécute sa sous-chaîne de traduction sur la première unité de traduction contenue dans ce lecteur et transmet les résultats (via le perforateur virtuel) sous forme d'un fichier de

traitement mis à jour vers une machine virtuelle noeud de réseau. Si le lecteur est vide, elle se met en attente d'un événement sur le lecteur ou de l'arrivée à échéance d'une période de temps.

Il est possible de provoquer l'arrêt immédiat ou différé (après le traitement en cours) d'une machine serveur en lui envoyant des messages adéquats.

iii. Les machines nœuds de réseau

Les machines "nœuds" du réseau centralisent les échanges sur le réseau, et servent d'intermédiaires entre les machines des deux autres types qui ne communiquent normalement pas directement entre elles. Elles permettent aussi d'assurer un suivi des flux de traitements. Leur rôle est essentiellement d'organiser l'échange des fichiers via le "Spool".

Dans une implémentation effective du réseau, on peut se contenter d'une unique machine nœud. On peut aussi en prévoir plusieurs en les organisant de façon symétrique, pour augmenter les performances, ou en sous-réseaux, pour augmenter la modularité et la sécurité.

En plus de ces machines virtuelles, on peut utiliser des machines annexes dédiées à des tâches diverses. On a ainsi, dans l'implémentation effectivement réalisée, particularisé une machine "moniteur réseau" de nom "LIDIA00" destinée à assurer le démarrage du réseau, son arrêt, le suivi de certains traitements, la consultation de l'état du réseau et diverses autres interventions sur ce même réseau.

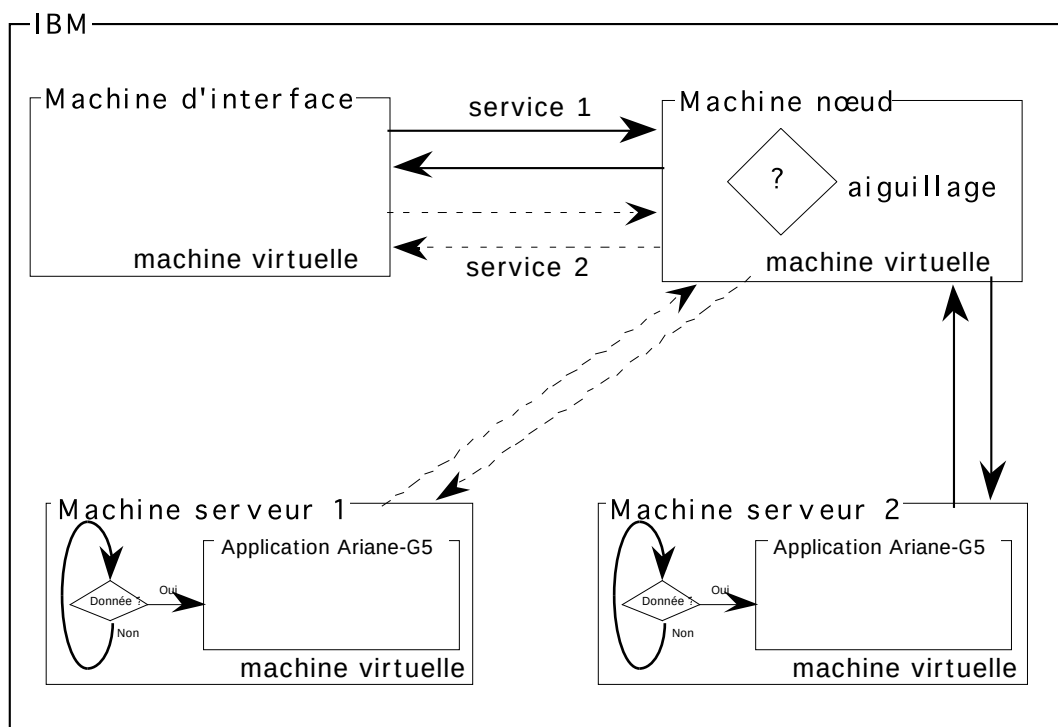


Figure 7.10 : Les machines virtuelles du réseau LIDIA/IBM

b. Environnement de traitement LIDIA

Sur une machine serveur, les sous-chaînes de traduction sont accessibles via l'environnement de traitement LIDIA qui permet l'appel au moniteur Ariane-G5, au moyen de trois commandes notées dans un fichier de commandes LIDIA. Ce fichier comporte :

- le nom de la sous-chaîne de traduction⁴⁸ ;
- le nom de la phase de reprise du traitement ;
- une liste de la ou des phases dont on veut produire les résultats.

Si l'objet HyperCard est à traduire en 3 langues, il y a 3 fichiers de commandes LIDIA pour le passage de la structure mmc aux trois traductions (as2, transfert, génération). Voici l'exemple concret d'une traduction vers l'allemand pour lequel nous avons besoin de trois commandes.

1. Analyse à partir du texte d'entrée

Fichier de commandes LIDIA:

Sous-chaîne de traduction	:	LDANFRA
Phase de reprise	:	INIT
Résultat(s) à produire	:	AS

La sous-chaîne de traduction LDANFRA demande en particulier que l'analyse structurale utilise la grammaire 1. Le résultat de l'analyse syntaxique (structure mmc sous forme lisp) est transféré vers le Macintosh pour désambiguïsation.

Notons au passage une curiosité ! La sous-chaîne de traduction LDANFRA n'effectue bien que l'analyse vers la forme mmc. Mais elle contient en outre des phases de transfert sans effet et des phases de génération qui produisent une pseudotraduction, qui est en fait la forme lisp de la structure mmc. Il a en effet paru plus judicieux d'utiliser Ariane-G5 — linguistiquement fait pour cela — plutôt que d'y ajouter encore d'autres formes de sorties d'arbres.

2. Suite du traitement à partir de la sortie d'analyse désambiguïsée

Fichier de commandes LIDIA:

Sous-chaîne de traduction	:	LD2ALD
Phase de reprise	:	AS
Résultat(s) à produire	:	GS GM

L'analyse structurale de LD2ALD utilise la grammaire 2. Le résultat du traitement est constitué par les sorties des processus de génération structurale (GS) et de traduction vers l'allemand (GM). Notons que la structure umc arrive au serveur de traduction sous une forme externe lisible, et est alors compilée vers la forme Ariane-G5 avant d'être envoyée vers le serveur.

⁴⁸ Une sous-chaîne de traduction est un objet nommé qui mémorise une chaîne de traduction et ses divers paramètres de trace, de sorties de résultats intermédiaires, et de contrôle [Quézel-Ambrunaz 1989]

3. Rétrotraduction de l'allemand vers le français

Fichier de commandes LIDIA:

Sous-chaîne de traduction : LD2FRA
 Phase de reprise : ASTL
 Résultat(s) à produire : GM

Résultat du traitement : rétrotraduction en français

Le résultat de la génération structurale (GS) produit par la seconde commande est utilisé comme entrée de la phase ASTL.

Le schéma suivant résume le processus.

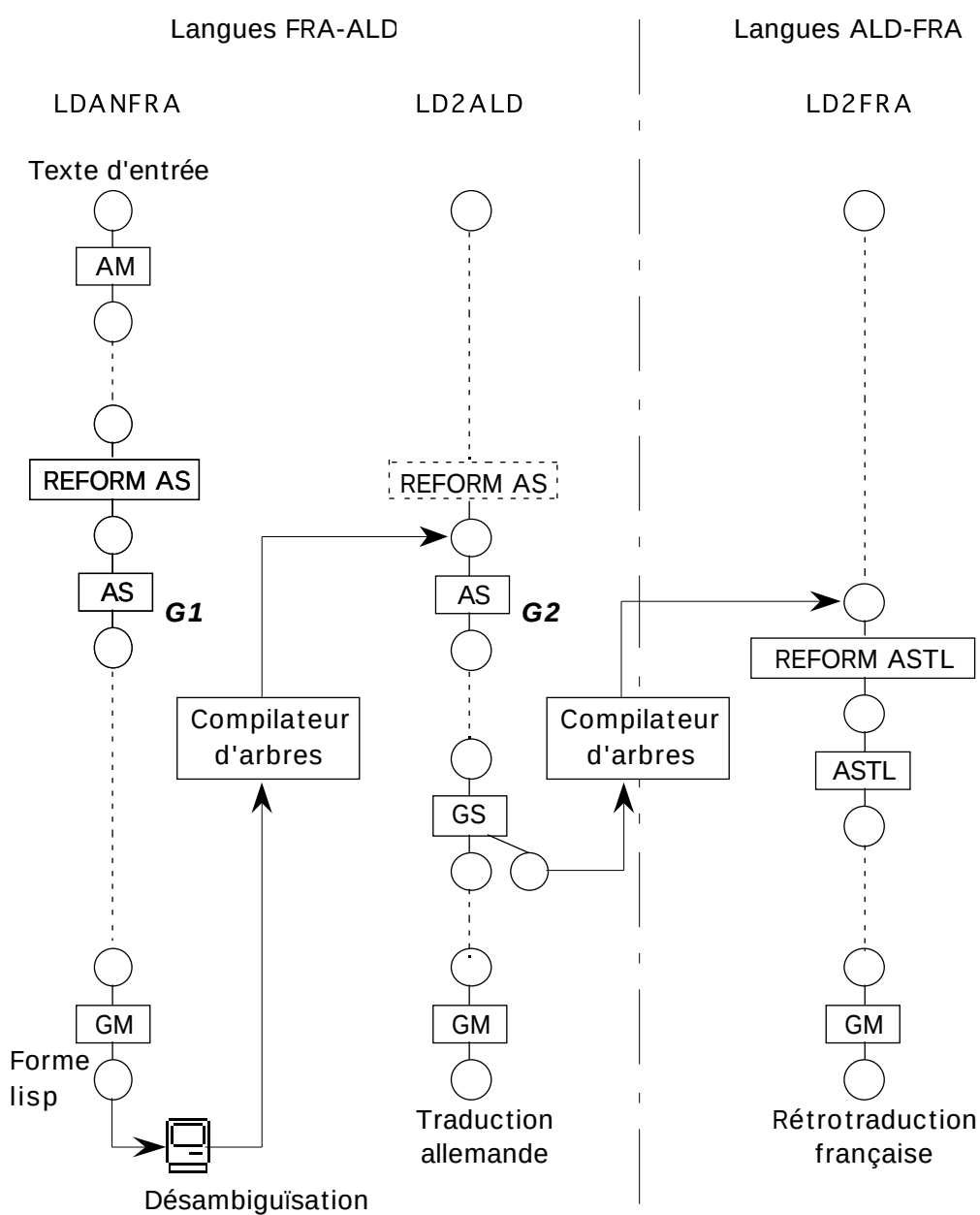


Figure 7.11 : Traitements linguistiques effectués sur le serveur de traduction

c. Interface

L'interface entre le réseau LIDIA/IBM et le serveur de communication est assurée par l'échange d'un fichier dit "fichier requête" et d'une commande.

i. Fichier requête

Il est constitué par une suite de blocs séparés par des délimiteurs.

<u>Contenu du fichier :</u>	<u>Commentaire :</u>
/ *-----*/	Délimiteur Bloc 1 Ø
Stack_name : démo_LIDIA	Nom de la pile
Card_ID : 128	ID de la carte
Object_ID : 752	ID de l'objet
Objetc_type : field	Type de l'objet
/*-----*/	Délimiteur Bloc 2 Ø
LGS = fvx	Langue source
TLS = --sans type--	Type de texte
LLC = ()	Langues cibles visées
TYTR = I	Type de traitement
/*-----*/	Délimiteur Bloc 3 Ø
*JEAN A ACHETE!1 UN LIVRE A!2 *PIERRE.	Donnée à traiter
/*-----*/	Délimiteur Bloc 4 Ø
((nil ""ULTXT"" nil) ((nil ""ULFRA"" nil) ((nil ""ULSOL"" nil) ((nil ""PHVB"" (K phvb CAT v ENONCP decl VOIX act PHASE acc MT ipr SUBV vf PERS 3 LINKS oui R echts oui NBR sing VALET q ARGS (a1 a0))) ((nil ""DGN"" (K gn FS suj RL arg0 CAT (d n) SUBN nc GNR mas PERS 3 NBR sing VALET n)) ((Le ""LE"" (FS des CAT d GNR mas NBR sing TYPOG pcap))) ((capitaine""CAPITAINE"" (FS gov CAT n SUBN nc GNR mas PERS 3 NBR sing SENS (3 2 1)))) ((nil ""NV"" (K phvb CAT v ENONCP decl VOIX act PHASE acc MT ipr SUBV vf PERS 3 LINKS oui R echts oui NBR sing VALET q ARGS (a1 a0))) ("a"" ""AVOIR"" (FS aux CAT v MT ipr SUBV vf PERS 3 NBR sing SENS 1))) ("rapporté""RAPPORTER"" (FS gov CAT v SUBV ppa GNR mas PERS 3 NBR sing SENS (3 2 1)))) ((nil ""DGN"" (K gn FS obj RL arg1 CAT (d n) SUBN nc GNR mas PERS 3 RECHTS oui NBR sing VALET n)) ("un"" ""UN"" (FS des CAT d GNR mas NBR sing))) ("vase"" ""VASE/LE"" (FS gov CAT n SUBN nc GNR mas NBR sing SENS 1))) ((nil ""GP"" (K gn FS comp RS unde CAT (s n) SUBN np GNR fem PERS 3 NBR sing VALET den)) ("de"" ""DE"" (FS reg CAT s))) ("Chine"" ""CHINE"" (FS gov CAT n SUBN np GNR fem NBR sing SENS 1)))) ("." ""."" (CAT p)))) ((nil ""ULSOL"" nil) ((nil ""PHVB"" (...)) ((nil ""DGN"" (...)) ((Le ""LE"" (...)) ((capitaine""CAPITAINE"" (...)) ((nil ""NV"" (...)) ("a"" ""AVOIR"" (...)) ("rapporté""RAPPORTER"" (...)) ((nil ""DGN"" (...)) ("un"" ""UN"" (...)) ("vase"" ""VASE/LE"" (...)) ((nil ""GP"" (K gn FS circ RS unde CAT (s n) SUBN np GNR fem PERS 3 NBR sing VALET den)) ("de"" ""DE"" (FS reg CAT s))) ("Chine"" ""CHINE"" (FS gov CAT n SUBN np GNR fem NBR sing SENS 1)))) ("." ""."" (CAT p))))))))	Résultat, mmc sous forme lisp

Le bloc 4 peut être suivi d'autres blocs si le traitement requis produit plusieurs résultats.

ii. Les paramètres de traitement

Les traitements s'organisent en trois étapes : analyse, transfert, et génération. L'analyse part d'un texte standardisé pour produire une structure mmc. Le transfert part d'une structure uma langue source pour produire une structure uma langue cible. La génération part d'une structure uma et produit une structure umc et un texte traduit.

On peut effectuer séparément ces différentes étapes: analyse (TYTR = A), transfert (TYTR = C) et génération (TYTR = G), ou les enchaîner par deux: analyse + transfert (TYTR = B), transfert + génération (TYTR = D), ou toutes les trois (TYTR = E).

Lors du retour vers la station, TYTR comporte une des trois valeurs: I, J ou Z selon que ce fichier provient d'une phase d'analyse (I), de transfert (J) ou de génération (Z). TYTR figure parmi les paramètres du bloc numéro deux.

On y trouve également:

– LGS = <code_langue_source>

– TLS = <type_de_langage>

et optionnellement, une liste de langues cibles qui peut être vide.

– LLC = (<liste_lg_cible>) ! LLC = ()

<liste_lg_cible> == <lg_cible> ! <lg_cible> , <liste_lg_cible>

Pour les phases autres que "A", la valeur de TYTR est suivie obligatoirement du code de la langue cible active, qui doit figurer dans la liste LLC. Les valeurs de LGS, TLS et la langue cible déterminent les applications Ariane-G5 intervenant, et TYTR le type de traitement que l'on veut y associer.

iii. Les procédures d'échange

Les procédures d'échange sont décrites au paragraphe 8.1.3.

7.3.2. Linguiciel en Ariane-G5

Plutôt qu'une description académique de Ariane-G5, nous donnons le résultat produit par chacune des phases Ariane pour la phrase "Le capitaine a rapporté un vase de Chine." dans le cadre d'une traduction vers l'allemand.

Cette description et les annexes qui s'y rapportent sont issues de [Guilbaud 1993]. En annexe 15 nous donnons des chiffres sur le linguiciel français-allemand.

a. Phase d'Analyse Morphologique

Le texte est d'abord lemmatisé. Le module AM (cf. annexe 1) produit un arbre plat dont la décoration de chacune des feuilles contient l'information morphologique pour chacun des mots analysés. La variable prédéfinie UL donne le lemme de référence.

À ce niveau d'analyse, on observe deux cas d'ambiguïtés:

- “le” est soit déicteur, soit pronom [cat (d, r)];
- “vase” est masculin singulier ou bien féminin singulier [gnbn (ms, fs)].

b. Phase dictionnaire AX

Après exécution de la phase AX (cf. annexe 1), la variable UL sert à indiquer la valeur d'unité lexicale de chacun des lemmes (l'unité lexicale ou UL est une famille dérivationnelle de lemmes. On a remplacé les lemmes “le” et “vase” par des sous-arbres dont chacun des dépendants renvoie à une valeur d'UL différente :

- le = [UL('le'),cat(d)] ou bien UL('jetuil'),cat(r)];
- vase = [UL('vase/la'),gnbn(fs)] ou bien [UL('vase/le'),gnbn(ms)]

c. Phase dictionnaire AY

AY est le dictionnaire d'analyse (cf. annexe 1). Un fois consulté, on a toute l'information lexicale, syntaxique et sémantique pour entreprendre l'analyse structurale (AS). À chaque noeud lexical de l'arbre d'entrée, on a associé un sous-arbre appelé ensemble de “cerises”. Une cerise exprime, pour les prédicats, une construction syntaxo-sémantique et l'ensemble des sens possibles associés à cette construction, et, pour les entités non prédictives, une description sémantique avec les sens concernés. Par exemple, l'UL “rapporter” a 5 constructions et 4 sens (ou acceptions) possibles dans la base lexicale PARAX (cf. § 7.3.3. et 7.1.2.) et décrits à l'issue d'AY comme suit :

```
12 'RAPPORTE!1': UL('RAPPORTER'), SUBV(PPA), CAT(V), GNR(MAS), NBR(SING).
13 'RAPPORTE!1': UL('C3A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V),
    ARFAC(A2), GNR(MAS), NBR(SING), SEM1(CONCRET), SEM2(ANIME), SEMZ(ANIME), SENS(1),
    SSANIM2(HUM).
14 'RAPPORTE!1': UL('C3B'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V),
    ARFAC(A2), GNR(MAS), NBR(SING), SEM1(ABSTRAIT), SEM2(ANIME), SEMZ(ANIME), SENS(4),
    SSANIM2(HUM).
15 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(Q), VAL2(AN), CAT(V),
    ARFAC(A2), GNR(MAS), NBR(SING), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
16 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V),
    ARFAC(A1,A2), GNR(MAS), NBR(SING), SEM1(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS),
    SEM2(ANIME), SEMZ(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SENS(3), SSANIM2(HUM).
17 'RAPPORTE!1': UL('C2A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), CAT(V), GNR(MAS),
    NBR(SING), SEM1(CONCRET), SEMZ(ANIME), SENS(2), SSANIMZ(HUM).
```

Les valences syntaxiques, les valences sémantiques et la facultativité éventuelle des arguments du prédicat “rapporter” sont décrits, respectivement, à l'aide des variables

- [val0, val1, val2];
- [semz, sem1, sem2];
- [arfac].

L'entité “capitaine”, quant à elle, a trois acceptions possibles [sens (1, 2, 3)].

d. Phase d'Analyse Syntaxique

La phase AS d'analyse structurale syntaxo-sémantique est réalisée en trois étapes principales :

- production de la structure mmc en 5 sous-étapes, avec AS1, AS2, AS3, AS5 et AS6;
- transformation de l'arbre Ariane de cette structure mmc en arbre LISP grâce à des modules Ariane de transfert (langues FVX-LSP) et de génération (langue LSP);

- transformation de la structure umc récupérée de la phase de désambiguïsation interactive avec l'auteur en structure uma avec AS7.

La grammaire n° 1 d'AS (cf. annexe 2) procède essentiellement à des désambiguïsations par exploration du contexte immédiat des mots ambigus. Cela permet d'alléger les traitements ultérieurs. Ainsi, dans la phrase qui nous occupe, le mot “le” est reconnu comme un article [UL('le')] et non pas comme un pronom, et le mot “vase” est reconnu comme étant du masculin [UL('vase/le'), gnr(mas)].

La grammaire n° 2 d'AS (cf. annexe 3) effectue les premières constructions syntaxiques. Elle construit les groupes simples et sûrs. Ainsi, “le capitaine a rapporté” est identifié comme un groupe verbal [k(phvb)] muni de son sujet [fs(suj), rl(arg0)], “le capitaine”, “un vase” et “de Chine”, comme des groupes nominaux [k(gn)].

La grammaire n° 3 d'AS (cf. annexe 4) constitue la grammaire d'analyse principale. Elle travaille selon un mode non-déterministe et n-aire, propre au moteur ROBRA. Cela veut dire que l'on essaie toutes les analyses possibles. On obtient de cette façon deux interprétations pour “Le capitaine a rapporté un vase de Chine”, chacune avec sa géométrie et ses décorations particulières. Ces deux interprétations différentes sont en clair :

< le capitaine a rapporté un vase chinois > et < le capitaine a rapporté de Chine un vase (qui n'est pas nécessairement chinois !) > :

- 21 '*CHINE': UL('GP'), FS(COMP), K(GN), RS(UNDE), SUBN(NP), VALET(DEN), CAT(N,S), GNR(FEM), NBR(SING), PERS(3).
- 46 '*CHINE': UL('GP'), FS(CIRC), K(GN), RS(UNDE), SUBN(NP), VALET(DEN), CAT(N,S), GNR(FEM), NBR(SING), PERS(3).

La grammaire n° 5 d'AS (cf. annexe 5) est une grammaire qui filtre les résultats produits par grammaire n° 3 et effectue la recherche de l'antécédent des pronoms.

La grammaire n° 6 d'AS (cf. annexe 6) effectue quelques aménagements dans l'arbre d'analyse, comme la suppression des cerises et la recopie de l'information qu'elles contiennent sur le gouverneur [fs(gov)] des groupes, ainsi que la création d'un noyau verbal [ul(nv)] regroupant les éléments verbaux des prédicats. Elle produit la version Ariane de la structure mmc source qui sera traduite en forme LISP (cf. annexe 7) lors de la deuxième étape principale annoncée plus haut.

La troisième et dernière étape de l'analyse est effectuée par la grammaire n° 7 (cf. annexe 8) qui prend en entrée une umc qu'elle transforme en uma. Cette dernière est débarrassée des mots outils tels que les auxiliaires de conjugaison et les prépositions régies des groupes nominaux, d'une part, et contient sous chacun des gouverneurs lexicaux autant de cerises qu'il y a de valeurs de sens encore possibles pour eux, d'autre part. Si la désambiguïsation est complète, il ne doit y avoir qu'une seule cerise sous chaque gouverneur. On a finalement retenu pour “Le capitaine a rapporté un vase de Chine” l'interprétation n°1 < le capitaine a rapporté un vase chinois >, et pour “capitaine” et “rapporter”, les acceptions n° 2, [sens(2)].

e. Phase de Transfert Lexical

TL (cf. annexe 9) est une phase de consultation de dictionnaire bilingue, pendant laquelle chaque valeur d'UL source est remplacée par la valeur qui lui correspond en cible. En outre, chaque UL cible est accompagnée des propriétés syntaxiques distributionnelles qui la concernent afin de pouvoir effectuer une génération syntaxique correcte du texte cible en GS et GM. Dans notre exemple, les UL à traduire sont : “le”, “capitaine”, “rapporter”, “de” et “*chine”. Elles sont remplacées respectivement par “*kapitaen”, “mitbringen”, “aus” et “*china”. Les décorations associées sont transformées comme suit :

- UL('LE'), FS(DES), CAT(D), GNR(MAS), NBR(SING), TYPOG(PCAP).
UL('DER'), FS(DES), GNR(MAS), CAT(D), NBR(SING), TYPOG(PCAP).
- UL('CAPITAINE'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), PERS(3), SENS(2).
UL('*KAPITAEN'), SUBN(NC), GENUS(MAS), GNR(MAS), CAT(N), NBR(SING), PERS(3), SENS(1).
- UL('RAPPORTER'), SUBV(PPA), CAT(V), GNR(MAS), NBR(SING), PERS(3), SENS(2).
UL('MITBRINGEN-V'), SUBV(PPA), MT(PPA), AUXV(HABEN), CAS1(ACC), GNR(MAS), CAT(V), NBR(SING), PERS(3), SENS(2), VAL0(GN), VAL1(GN).
- UL('UN'), FS(DES), CAT(D), GNR(MAS), NBR(SING).
UL('EIN'), FS(DES), GNR(MAS), CAT(D), NBR(SING).
- UL('*VASE'), SUBN(NC), GENUS(FEM), GNR(MAS), CAT(N), NBR(SING), SENS(1).
UL('*VASE'), SUBN(NC), GENUS(FEM), GNR(MAS), CAT(N), NBR(SING), SENS(1).
- UL('DE'), FS(REG), CAT(S).
UL('AUS'), FS(REG), CAS1(DAT), CAT(S), VAL1(GN).
- UL('*CHINE'), FS(GOV), SUBN(NP), CAT(N), GNR(FEM), NBR(SING), SENS(1).
UL('*CHINA'), SUBN(NP), GENUS(NEU), GNR(FEM), CAT(N), NBR(SING), SEMN(LIEU), SENS(1),
POTDRV(NOMADJ).

f. Phase dictionnaire TX

Les dictionnaires de la phase TX (cf. annexe 10) permettent de remplacer les UL cibles qui sont en fait des locutions, par un sous-arbre. C'est le cas, par exemple de [UL("mitbringen")]:

- 9: 'GOV' (10: 'MITBRINGEN-V')
9: 'GOV' (10: 'BRINGEN-V' (11: MIT)

g. Phase de Transfert Syntaxique

TS (cf. annexe 11) est une grammaire comparative bilingue. On y traite tous les aspects grammaticaux pour lesquels on n'a pas de modèle de représentation suffisamment universel, comme le mode, le temps, l'aspect, et la détermination.

h. Phase de Génération Syntaxique

La phase GS de génération syntaxique est réalisée en deux étapes. La première étape GS1 (cf. annexe 12) est censée produire un résultat qui correspondrait à une structure uma (éventuellement mma) obtenue par un analyseur de la langue cible. La deuxième étape GS2 (cf. annexe 13) transforme cette uma en structure de surface en vue de la génération morphologique du texte cible. Cette organisation du processus de GS a été décidée à des fins de rétrotraduction du texte source : le rédacteur désirant avoir une idée de la traduction obtenue en langue cible, alors qu'il ne connaît pas cette langue, peut lire son texte retraduit en français à partir d'une uma cible.

i. Phase dictionnaire GY

La phase GY (cf. annexe 13) est une phase EXPANS, comme AX, AY, TL, ou TX. À ce titre elle permet d'interfacer les jeux de variables de la phase précédente avec la suivante, qui use d'un jeu différent. La partie dictionnaire permet également d'effectuer des ajustements lexicaux.

j. Phase de Génération Morphologique

En GM (cf. annexe 14), chaque feuille de l'arbre est parcourue afin de générer, morphe après morphe, le mot qui lui correspond. Le transducteur arbre à chaîne de SYGMOR permet d'obtenir l'image textuelle cible du texte source: c'est à nouveau une chaîne de caractères séparés par des blancs.

TEXTE SOURCE EN ENTRÉE D'AM : Le capitaine a rapporté un vase de Chine.

TEXTE CIBLE EN SORTIE DE GM : Der Kapitaen hat eine chinesische Vase mitgebracht.

7.3.3. Génie linguiciel entre Parax et les dictionnaires Ariane-G5

Parax permet d'indexer complètement chaque terme, aux niveaux morphologique, syntaxique et sémantique. Les dictionnaires utilisés par les linguiciels Ariane-G5 sont construits avec les informations contenues dans Parax.

Nous donnons ici l'exemple du verbe atteindre.

<u>atteindre 1,a</u> #atteindre_blesser SENS:1. CAT: vb. VAL1: nom. AUX: avoir.	#atteindre_blesser blesser par une arme (<i>il l'a atteint au front d'une balle.</i>) ** SEMPR: activité. SEMZ: humain, personnif, outil, véhicule. SEM1: humain, animal.
<u>atteindre 1,b</u> #atteindre_moralt SENS:2. CAT: vb. VAL1: nom. AUX: avoir.	#atteindre_moralt toucher moralement (<i>le reproche l'atteint au vif.</i>) ** SEMPR: procpysch. SEMZ: humain, entabst. SEM1: humain.
<u>atteindre 3,a</u> #parvenir_àlieu SENS:3. CAT: vb. VAL1: nom. AUX: avoir.	#parvenir_àlieu parvenir à un lieu (<i>parvenir à Paris avant la nuit.</i>) ** SEMPR: déplact. SEMZ: animé, physique. SEM1: lieu.
<u>atteindre 3,b</u> #atteindre_but SENS:4. CAT: vb. AUX: avoir. VAL1: nom.	#atteindre_but parvenir à rattraper quelqu'un ou qqch (<i>atteindre un but, une cible.</i>) **SEMPR: activité. SEMZ: animé, physique. SEM1: physique.

Figure 7.12 : Entrée française “atteindre” de la base Parax

La figure 7.12 montre l'entrée “atteindre” pour le français. Cette entrée comporte quatre sens. Les informations morphologiques apparaissent dans la colonne de gauche, les informations sémantiques dans la colonne de droite.

La figure 7.13 page suivante montre l'entrée “atteindre” du dictionnaire d'analyse Ariane. On observe que l'entrée du dictionnaire Ariane est structurée par constructions. Les quatre constructions “C2A” à “C2D” correspondent ici respectivement à chacun des sens définis dans Parax “\$S1” à “\$S4”.

Les informations qui apparaissent sont extraites de Parax et peuvent être codées différemment. C'est ainsi que les traits sémantiques de l'argument zéro du verbe décrits dans la rubrique "SEMZ" de Parax sont codés, en Ariane, sous la forme de "procédures d'affectations" (\$OHUM, \$OPERS...).

```
'ATTEINDRE'      ==/0(1,2,3,4)/
1 : 'C2A', $AVOIR,$S1,
      $ON,$OHUM,$OPERS,$OMACH,$OOUT,$OVEH,
      $1N,$1HUM,$1ANIMAL;
2 : 'C2B', $AVOIR,$S2,
      $ON,$OHUM,$OABS,
      $1N,$1HUM;
3 : 'C2C', $AVOIR,$S3,
      $ON,$OANIM,$OCONC,
      $1N,$1LOC;
4 : 'C2D', $AVOIR,$S4,
      $ON,$OANIM,$OCONC,
      $1N,$1CONC.
```

Figure 7.13 : Entrée du dictionnaire d'analyse du français en Ariane-G5

Dans le dictionnaire de transfert entre le français et l'allemand, on a les correspondances suivantes (figure 7.14):

- atteindre sens 1 Æ treffen sens 1 (cf. figure 7.15)
- atteindre sens 2 Æ treffen sens 2
- atteindre sens 3 Æ erreichen sens 1 (cf. figure 7.16)
- atteindre sens 4 Æ erreichen sens 2

```
'ATTEINDRE'      ==$GOV/'GOV'/
      $S1 /'TREFFEN-V', $B1, $HAB, $OGN, $1ACC/
      $S2 /'TREFFEN-V', $B2, $HAB, $OGN, $1ACC/
      $S3 /'ERREICHEN-V', $B1, $HAB, $OGN, $1ACC/
      $S4 /'ERREICHEN-V', $B2, $HAB, $OGN, $1ACC/
      /' ', *VIDE.
```

Figure 7.14 : Entrée "atteindre" du dictionnaire de transfert français-allemand

treffen #atteindre_moralt SENS:1. CAT: vb. VAL1: acc. AUX: h.	*atteindre_moralt toucher moralement (<i>'le reproche l'atteint au vif'</i>) ** SEMPR: procp psych. SEMZ: humain, entabst. SEM1: humain.
treffen #atteindre_blesser SENS:2. CAT: vb. VAL1: acc. AUX: h.	*atteindre_blesser blesser par une arme (<i>'il l'a atteint au front d'une balle'</i>) ** SEMPR: activité. SEMZ: humain, personnif, outil, véhicule. SEM1: humain, animal.

Figure 7.15 : Entrée allemande "treffen" de la base Parax

erreichen #atteindre_but SENS:1. CAT: vb. VAL1: acc. AUX: h.	*atteindre_but parvenir à rattraper quelqu'un ou qqch (<i>'atteindre un but, une cible.'</i>) **SEMPR: activité. SEMZ: animé, physique. SEM1: physique.
erreichen #parvenir_àlieu SENS:2.	*parvenir_àlieu parvenir à un lieu (<i>'parvenir à Paris avant la nuit.'</i>) **SEMPR: déplact. SEMZ: animé, physique. SEM1: lieu.

Figure 7.16 : Entrée “erreichen” du dictionnaire de génération de l’allemand

Chapitre 8

Implémentation

Dans la maquette LIDIA-1, trois machines physiques coopèrent pour former un système de traduction automatisée fondée sur le dialogue.

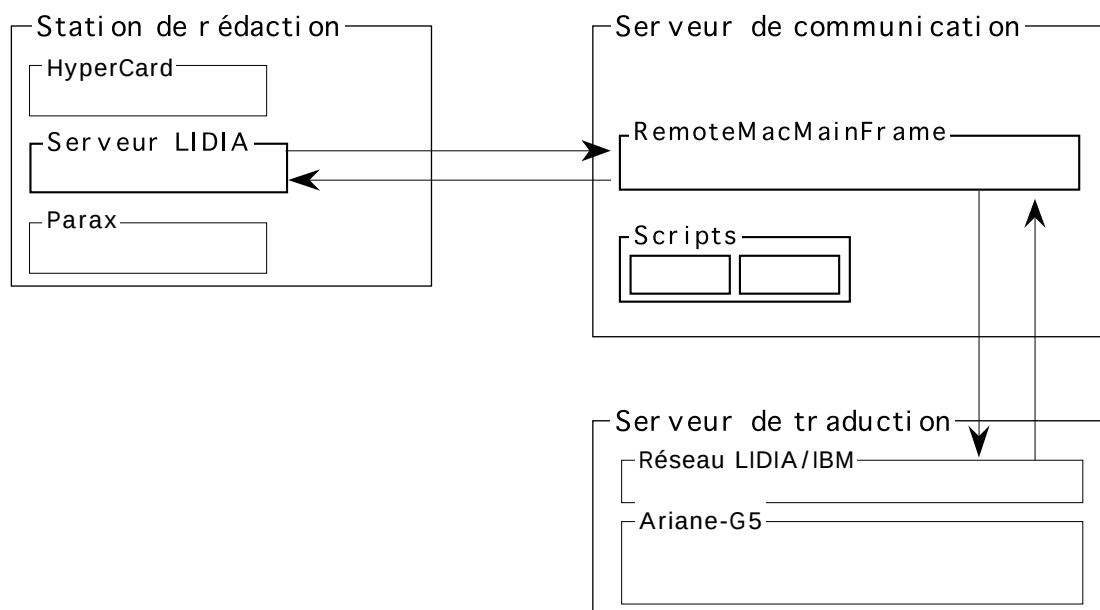


Figure 8.1 : Vue générale de l'architecture de la maquette LIDIA-1

8.1. Serveurs

Le serveur LIDIA est l'un des noyaux du système, il a en charge l'ordonnancement des traitements pour les objets qui doivent être traduits. Il intègre aussi un certain nombre de services indispensables, tels que le module de désambiguïsation et le module de transcription. Le serveur est implémenté en CLOS.

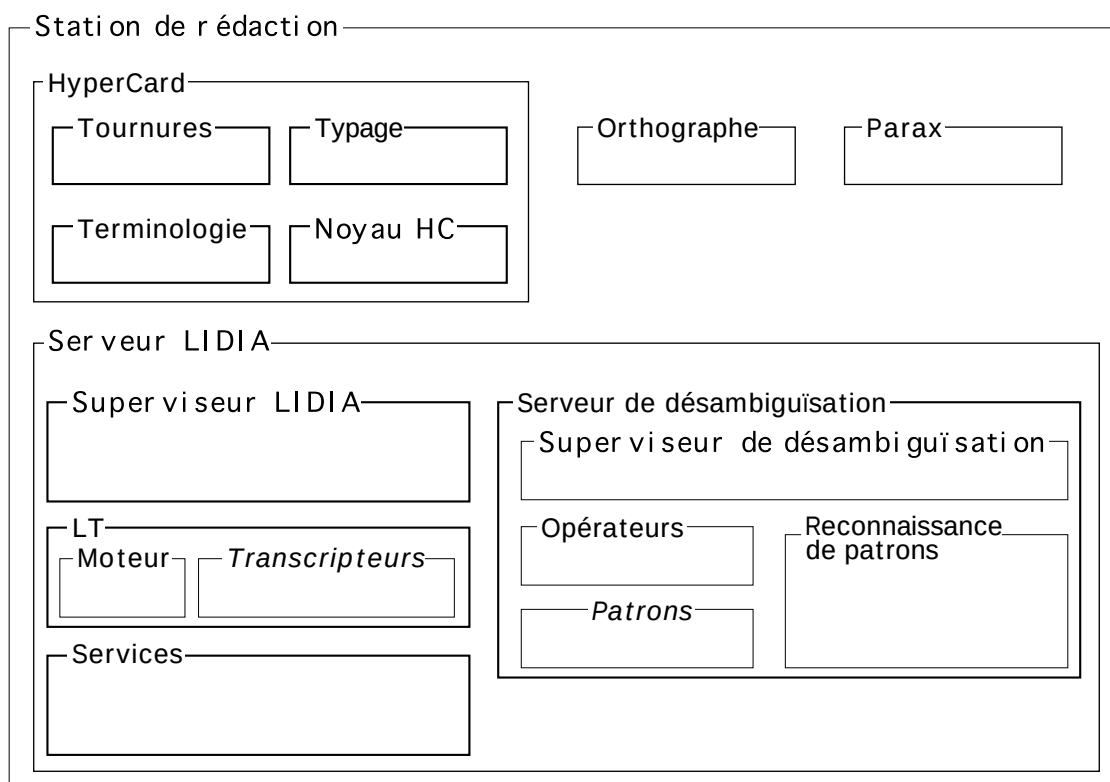


Figure 8.2 : Services LIDIA sur la station de rédaction

HyperCard héberge les modules de typage, traitement des tournures et traitement de la terminologie. Parax et le correcteur orthographique sont deux logiciels indépendants.

La caractéristique principale du système est que ses différents composants logiciels interagissent en échangeant des messages qui contiennent des directives. Les données partagées sont modifiées par le Superviseur LIDIA et le serveur de communication.

8.1.1. Superviseur LIDIA

Au niveau de l'implémentation, la boîte noire Superviseur LIDIA correspond au noyau de Macintosh Common Lisp. C'est en effet ce noyau qui reçoit les messages en provenance de HyperCard et du serveur de communication. Le serveur dispose d'un handler par type de message.

Le superviseur demande des services à HyperCard, au serveur de communication et au serveur LIDIA.

Fonctionnement pas à pas :

Lorsque l'utilisateur demande de traiter la sélection sur un objet de sa pile, un message est envoyé au superviseur pour lui indiquer qu'un nouvel objet entre dans la chaîne LIDIA.

En réponse à ce message, le superviseur crée le fichier miroir qui sera associé à l'objet pendant toute la durée des traitements LIDIA. Ce fichier est décrit en détail dans le § 8.3.3.

Lorsque le fichier miroir est créé, la procédure d'ordonnancement est appelée par un message. Cette procédure examine l'état courant de l'objet et demande le traitement suivant.

Si le traitement est autonome,

- le traitement est déclenché
- le superviseur avertit HyperCard que le traitement s'est terminé normalement. Ce type de messages permet de mettre à jour les boutons informatifs qui concernent l'objet :
 - état du traitement ,
 - disponibilité de la forme annotée ,
 - disponibilité de la rétrotraduction .

Si le traitement est interactif,

- le superviseur demande à HyperCard de modifier le menu Messages ;
- le superviseur demande à HyperCard de poser le bouton adéquat sur la surface de l'objet concerné par le dialogue :
 - demande de correction orthographique ,
 - demande de typage textuel ,
 - demande de vérification terminologique ,
 - demande d'emploi d'une tournure ou d'une formule ,
 - demande de clarification .
- le superviseur demande à HyperCard d'avertir l'utilisateur que son intervention est requise par l'envoi d'une notification,
- si c'est une demande de clarification qui est en suspens, l'arbre de questions concernant l'objet est mis dans une file d'attente.

Le superviseur est alors prêt à traiter un autre message.

8.1.2. Serveur de désambiguïsation

Le point d'entrée du serveur de désambiguïsation est le Superviseur de désambiguïsation, une fonction qui implémente une stratégie de désambiguïsation.

Ce superviseur produit un arbre de questions qui seront ensuite présentées à l'auteur. Le superviseur utilise un module de reconnaissance de patron et un certain nombre d'opérateurs qui constituent les primitives d'un langage de production de dialogues.

Le superviseur et les opérateurs sont décrits en détail dans le chapitre 9.

8.1.3. Serveur de communication

Le serveur de communication a un fonctionnement autonome. Il reçoit des ordres sous forme d'Apple events en provenance du serveur LIDIA.

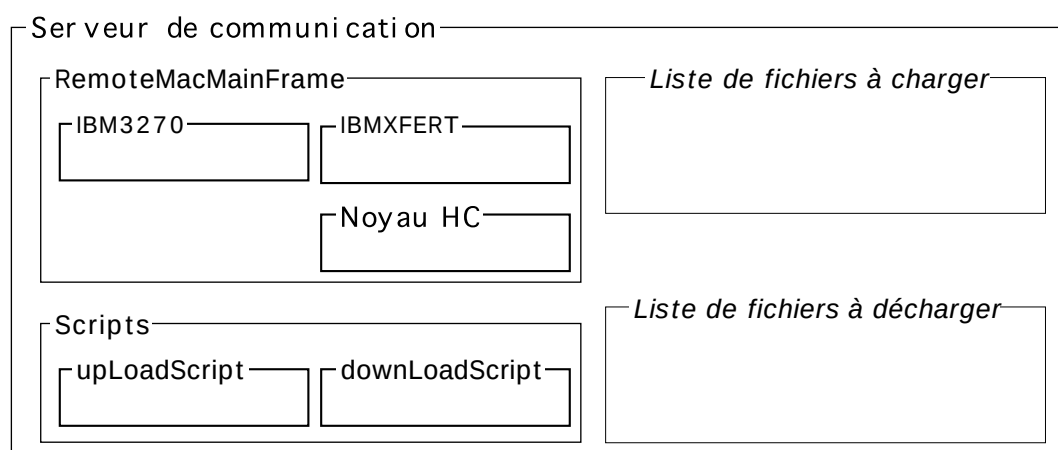


Figure 8.3 : Implémentation du serveur de communication

Le serveur de communication gère deux listes de fichiers, la liste des fichiers requête à transmettre au serveur de traduction et la liste des requêtes pour lesquelles le serveur attend une réponse.

Fonctionnement pas à pas :

Lorsque le serveur LIDIA a besoin de faire appel aux services du serveur de traduction, il envoie un message au serveur de communication qui identifie le fichier requête à transférer, c'est un message de upLoad.

Quant RemoteMacMainFrame reçoit un message de upLoad, il prend en charge le transfert du fichier requête depuis la station de rédaction. Il ajoute le nom de ce dernier fichier dans la liste des fichiers à transférer vers le serveur et dans la liste des fichiers à recevoir.

RemoteMacMainFrame reçoit aussi des ordres de chargement et de déchargement.

Lorsque RemoteMacMainFrame reçoit un ordre de chargement, le processeur de message upLoad exécute les traitements suivants :

- construction du fichier requête à partir d'un fichier contenu dans le dossier Poste de la station de rédaction,
 - construction du fichier batch qui va permettre d'envoyer le fichier requête,
 - exécution du fichier batch qui transfère effectivement le fichier requête,
- ```

IBMXFER("SETBATCH", 0, BatchFileName, session_number)
IBMXFER("DOBATCH", BatchFileName)

```
- envoi d'une commande à la machine interface afin que le fichier soit pris en compte sur le réseau LIDIA/IBM,
- ```

IBM3270("WRITE", "@&1,1, "APUP fName lidia", session_number)
IBM3270("SENDKEY", "Enter", session_number).
  
```

Lorsque RemoteMacMainFrame reçoit un ordre de déchargement, le processeur de message downLoad exécute les traitements suivants :

- envoi d'une commande à la machine interface afin que soit créé un fichier résultat,

```
IBM3270("WRITE", "@&1,1, "APDOWN", session_number)
IBM3270("SENDKEY", "Enter", session_number)
```

- construction du fichier batch qui va permettre de récupérer le fichier résultat,
- exécution du fichier batch qui transfère effectivement le fichier résultat,
- IBMXFER("SETBATCH", 0, BatchFileName, session_number)
- IBMXFER("DOBATCH", BatchFileName)
- envoi du résultat de la requête sur la station de rédaction,
- envoi d'un message au serveur signalant l'arrivée de ce fichier,
- send "(fileArrived file_Name)" to serveur_LIDIA without reply
- le serveur LIDIA modifie ensuite le fichier miroir et supervise la suite des traitements.

Deux scripts envoient régulièrement un message au Noyau HC afin que celui-ci effectue les transferts.

8.2. Communications

Le mécanisme de la communication inter-applications repose sur les *Apple events*. Dans la maquette, ces Apple events sont mis en œuvre de deux façons différentes, soit directement, soit via des scripts.

8.2.1. Apple events

La transmission des Appleevents ainsi que la gestion des files d'attente sont assurées par le système d'exploitation. Les applications doivent seulement prévoir d'y répondre. Pour ce faire, on doit écrire dans chaque application HyperCard et MCL une procédure qui réceptionne les messages et effectue les traitements requis.

Un Apple event est identifié par une classe (event class) qui regroupe un ensemble d'événements. Pour LIDIA, nous avons défini la classe "LIDI". Un Apple event est aussi défini par un identificateur (event ID) qui identifie un événement particulier au sein d'une classe.

a. Envoi

i. HyperCard

Des commandes intégrées à HyperTalk permettent d'envoyer des événements répertoriés dans la classe "misc". Il s'agit des événements "dosc" (do script) et "eval". Les commandes HyperTalk correspondant à ces deux événements sont :

```
send <expr> to program <programExpr> (without reply)
request <expr> of program <programExpr>
```

ii. MCL

Pour MCL, un ensemble de fonctions existent, qui permettent d'envoyer les événements "dosc" et "eval" à HyperCard.

```
(send-eval-to-hypercard "expression")
```

```
(send-dosc-to-hypercard "script")
```

Les messages LIDIA sont créés grâce à un ensemble de fonctions qui forment une boîte à outils pour les Apple events. Lorsqu'il est créé, un événement est envoyé à son destinataire par la commande `send-appleevent` décrite ci-dessous.

```
; send-appleevent
;
; Send an AppleEvent and possibly wait for a reply.
;
; the-appleevent    An AEDesc record for an Apple Event.
; the-reply         An AEDesc record. Reply is put here.
; reply-mode        One of :no-reply :queue-reply, :wait-reply.
;                   Default is :no-reply.
; interact-mode     One of NIL, :never-interact, :can-interact, :always-interact.
; can-switch-layer  boolean. True if user interaction is allowed to
;                   automatically switch layers. Default: NIL
; dont-reconnect    boolean. True if #_AESend should not attempt to reconnect if it
;                   gets a sessClosedErr from the PPC toolbox. Default: NIL.
; want-receipt      boolean. True if you want a receipt sent when
;                   the receiver gets the AppleEvent. Default: NIL
; priority          #$KAENormalPriority puts the apple event at the end of the
;                   event queue.
;                   #$KAEPriority puts it at the beginning of the event queue.
;                   Default: #$KAENormalPriority.
; timeout           Number of ticks to wait before a timeout error when reply-mode
;                   is :wait-reply or want-receipt is true.
;                   #$KAEDefaultTimeout "tells the Apple Event manager to provide
;                   an appropriate timeout." #kNoTimeOut means wait forever.
;                   Default: #$KAEDefaultTimeout.
; idleproc
; filterproc        Pascal routines. Use the defaults unless you have read and
;                   understand the Inside Mac description.
;
; Return value:     the-reply if reply-mode is :wait-reply. NIL, otherwise.

(defun send-appleevent (the-appleevent the-reply &key
  (reply-mode :no-reply)
  (interact-mode nil)
  (can-switch-layer nil)
  (dont-reconnect nil)
  (want-receipt nil)
  (priority #$KAENormalPriority)
  (timeout #$KAEDefaultTimeout)
  (idleproc appleevent-idle)
  filterproc)
  [...])
```

b. Réception et traitement

i. HyperCard

Le traitement des messages “eval” et “dosc” est pris directement en charge par HyperCard. Par contre, pour les messages LIDIA, nous avons écrit une procédure qui les intercepte, les décode et exécute les traitements requis.

```
on appleEvent Class, ID, sender
  if class & id is "LIDITYPE" then    --On a un événement pour le type
    request ae data with keyword "Object" --Objet concerné
    put the result into theObject
    request ae data with keyword "Interact" --Paramètres d'interaction
    put the result into theInteraction
    avertirType theObject theInteraction --On prépare la rétroaction
  else if class & ID is yy then
    [...]
  else
    pass appleEvent --On ne reconnaît pas cet Apple event, on le passe
  end if
end appleEvent

on avertirType objet interaction
[...]
end avertirType
```

ii. MCL

MCL peut répondre à tout événement pour lequel on a défini une méthode de traitement. Voici par exemple comment les événements “dosc” et “eval” sont interceptés et traités.

La définition des événements et de la procédure à exécuter lorsque l'événement est intercepté sont donnés ci-dessous. Ici la procédure est : eval-handler

```
(install-appleevent-handler :|misc| :|dosc| #'eval-handler)
(install-appleevent-handler :|misc| :|eval| #'eval-handler)
```

La méthode eval-handler rend le résultat de l'évaluation du paramètre.

```
(defmethod eval-handler ((a application)
                          theAppleEvent
                          reply
                          handlerRefcon)
  (declare (ignore handlerRefcon))
  (let ((what (ae-get-parameter-char theAppleEvent #$keyDirectObject t)))
    (check-required-params "unexpected parameters in scpt" theAppleEvent)
    (ae-put-parameter-char reply
                            #$keyDirectObject
                            (write-to-string (eval (read-from-string what)))
    )))
```

8.2.2. Scripts

Le texte du script de chargement dans son environnement d'édition est donné dans la figure 8.4. Le texte du script de déchargement est le même sauf que la ligne do script "upLoadFile" est remplacée par do script "downLoadFile".

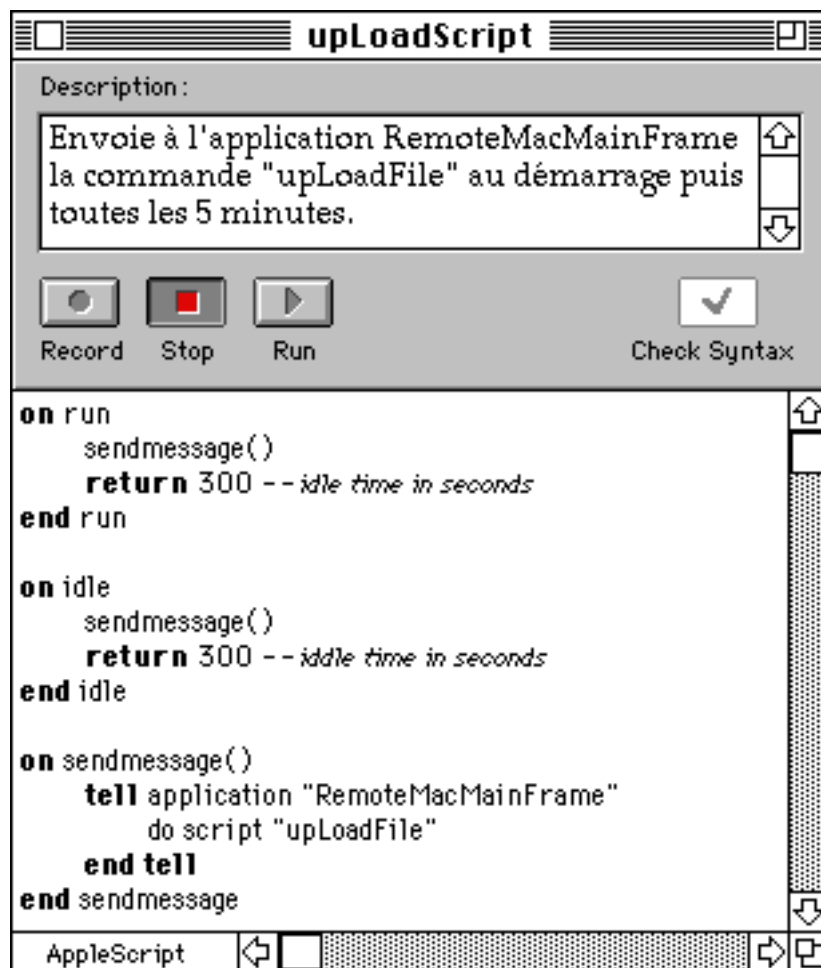


Figure 8.4 : Script de upLoad

Pour que RemoteMacMainFrame intercepte et traite les messages upLoadFile et downLoadFile. Le script de la pile Home de cet HyperCard déguisé comporte les processeurs de message suivantes.

```
on upLoadFile
    [...]      --déchargement du premier fichier qui est prêt
    [...]      --construction et envoi d'un fichier miroir à jour
    [...]      --avertissement que le résultat est arrivé
end upLoadFile
```

et

```
on downLoadFile
    [...]      --chargement d'un fichier requête
    [...]      --intégration du fichier sur le réseau LIDIA/IBM
end downLoadFile
```

8.2.3. Messages

Les schémas qui suivent résument les échanges de messages ou de fichiers qui ont lieu dans le système.

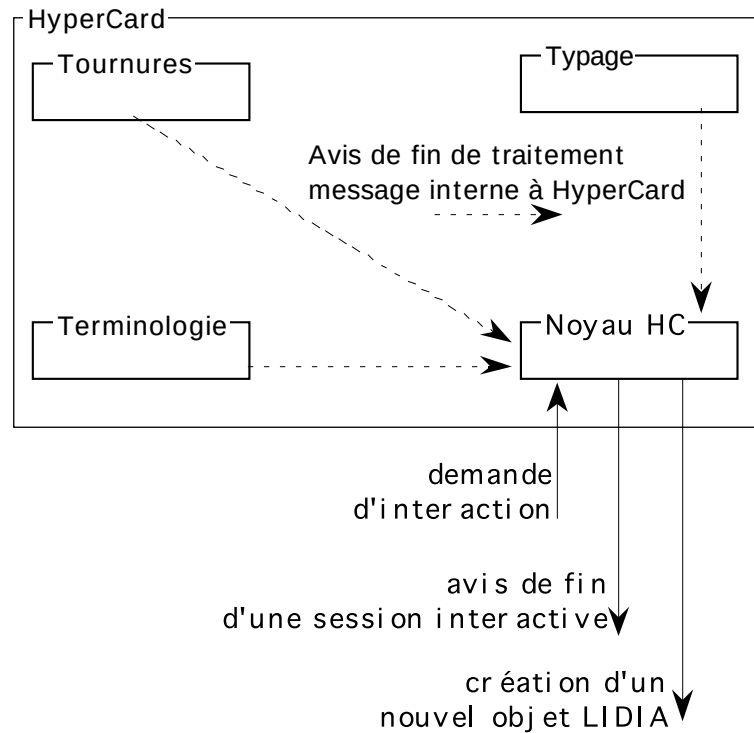


Figure 8.5 : Communication et HyperCard

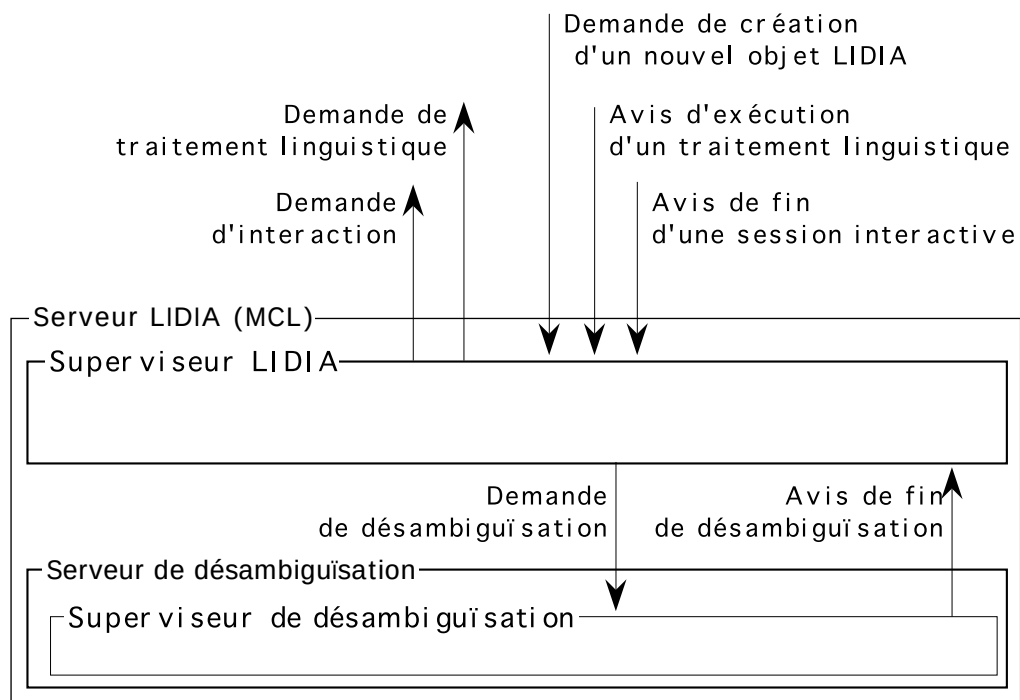


Figure 8.6 : Communication et serveur LIDIA

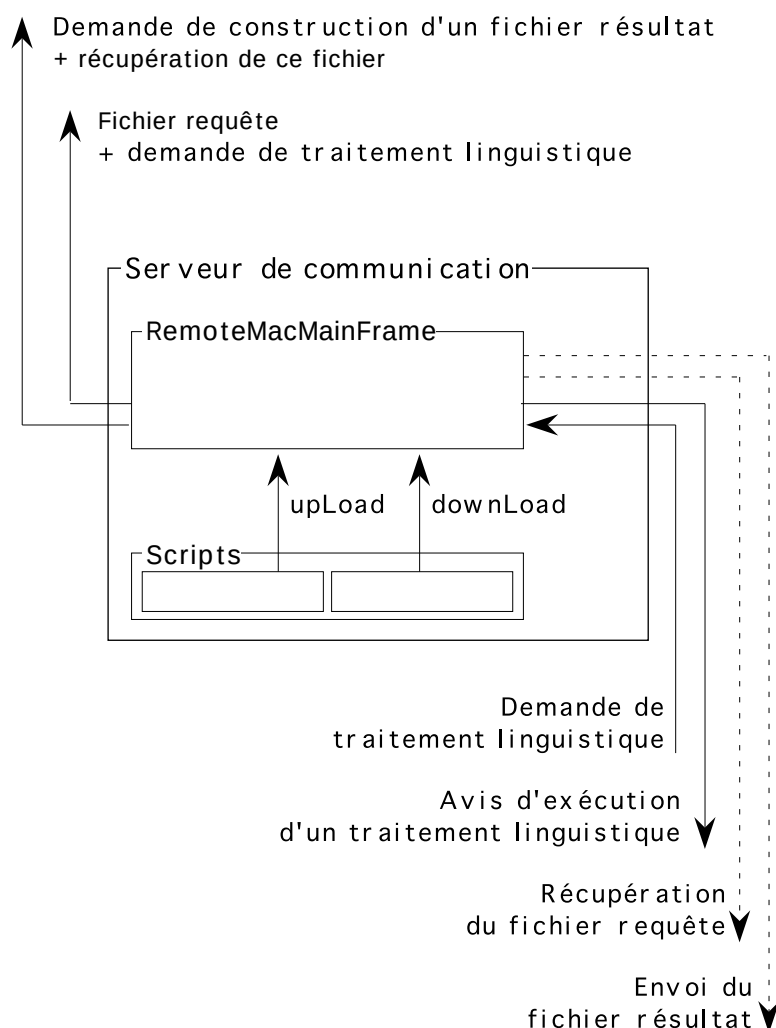


Figure 8.7 : Communication et serveur de communication

Sur ce dernier schéma figurent en pointillés les échanges de fichiers entre le serveur de communication et la station de rédaction. Ces échanges sont supervisés par le serveur de communication.

8.3. Informations manipulées et codage

8.3.1. Besoin de transcriptions⁴⁹

Nous sommes confrontés au problème 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 des systèmes d'exploitation "localisés".

⁴⁹Ce paragraphe est aussi extrait de [Boitet 1993] et [Boitet & Blanchon 1993b]

Mac.OS-7.1, fondé sur Unicode (un sous-ensemble de la norme multiscrit ISO-10646) et disponible depuis fin 1992, est le premier système d'exploitation réellement multiscrit⁵⁰ : avec n'importe quel texteur utilisant le "Script Manager" standard, il est possible de composer un document contenant des parties dans presque toutes les langues d'Europe, ainsi qu'en arabe, japonais, chinois, etc.

Cependant, Mac.OS-7.1 est encore un cas unique⁵¹, 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. Par exemple, l'ASCII français n'est pas le même sur un PC et sur un Macintosh. Dans notre cas, le serveur de TA n'utilise même pas de l'ASCII, mais de l'EBCDIC.

Notre solution est d'utiliser des transcriptions romaines pour les représentations 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, le GETA utilise depuis longtemps un jeu de caractères de transcription presque identique à celui de PL/I (ni minuscules ni diacritiques, mais seulement les majuscules simples, les signes de ponctuation usuels et quelques signes spéciaux). Cela donne une portabilité totale⁵², 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 partie micro de LIDIA-1, la lisibilité est plus importante, et nous utilisons un sous-ensemble plus grand de l'ISO-646, contenant les majuscules et les minuscules, mais pas les diacritiques. Les diacritiques sont représentés par des "séquences spéciales" introduites par "!" parce que, dans la documentation technique, les pièces sont souvent référencées par des identificateurs où lettres et chiffres sont mélangés (ex.: XA1). L'information relative à la structure ou à la mise en page du texte est représentée par des marques, ou "balises", dans l'esprit de SGML et de la TEI (<parag>, <section>, <greek>, etc.). On indique de façon analogue un changement de langue et/ou de système d'écriture (certaines langues en utilisent plus d'un).

⁵⁰ Mac.OS-7.1 n'est pas encore lui-même multilingue : bien que tous ses utilitaires soient indépendants des langues, chaque version distribuée n'a les messages et autres ressources spécifiques des langues que dans une langue.

⁵¹ 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.

⁵² Dans beaucoup de pays, comme la Thaïlande, les minuscules romaines sont remplacées par les caractères locaux dans les terminaux bilingues.

8.3.2. Transcriptions dans la maquette

a. Besoins

Nous définissons les jeux de caractères suivants :

- français_Macintosh : caractères du français affichables,
- français_Ariane : caractères du français pour Ariane,
- allemand_Macintosh : caractères de l'allemand affichables,
- allemand_Ariane : caractères de l'allemand pour Ariane,
- russe_Macintosh : caractères du russe affichables,
- russe_Ariane : caractères du russe pour Ariane.

Nous avons besoin des transpositeurs suivants :

- | | | |
|----------------------|---|--------------------|
| – français_Macintosh | ↔ | français_Ariane |
| – allemand_Ariane | ↔ | allemand_Macintosh |
| – russe_Ariane | ↔ | russe_Macintosh |

Soit six transpositeurs.

b. Outil

Ces six transpositeurs sont écrits en LT. LT est un langage permettant de décrire des transducteurs qui sont caractérisés par :

- un comportement déterministe,
- des états décrits par des étiquettes et des variables,
- deux têtes de lecture,
- une seule bande d'entrée,
- une seule bande de sortie,
- la lecture d'une chaîne non vide en entrée,
- l'écriture d'une chaîne éventuellement vide en sortie.

Le moteur LT réalisé à l'origine par Y. Lepage [Lepage 1986] et encore opérationnel sous Pascal/VS a été réimplémenté et étendu sur Macintosh sous MCL par M. Lafourcade [Lafourcade 1993a]. C'est cette dernière version que nous utilisons.

Pour l'écriture des transpositeurs, on dispose d'un éditeur syntaxique.

c. Exemple

Voici un extrait du transpositeur français_Macintosh ↔ français_Ariane dans son environnement d'édition.

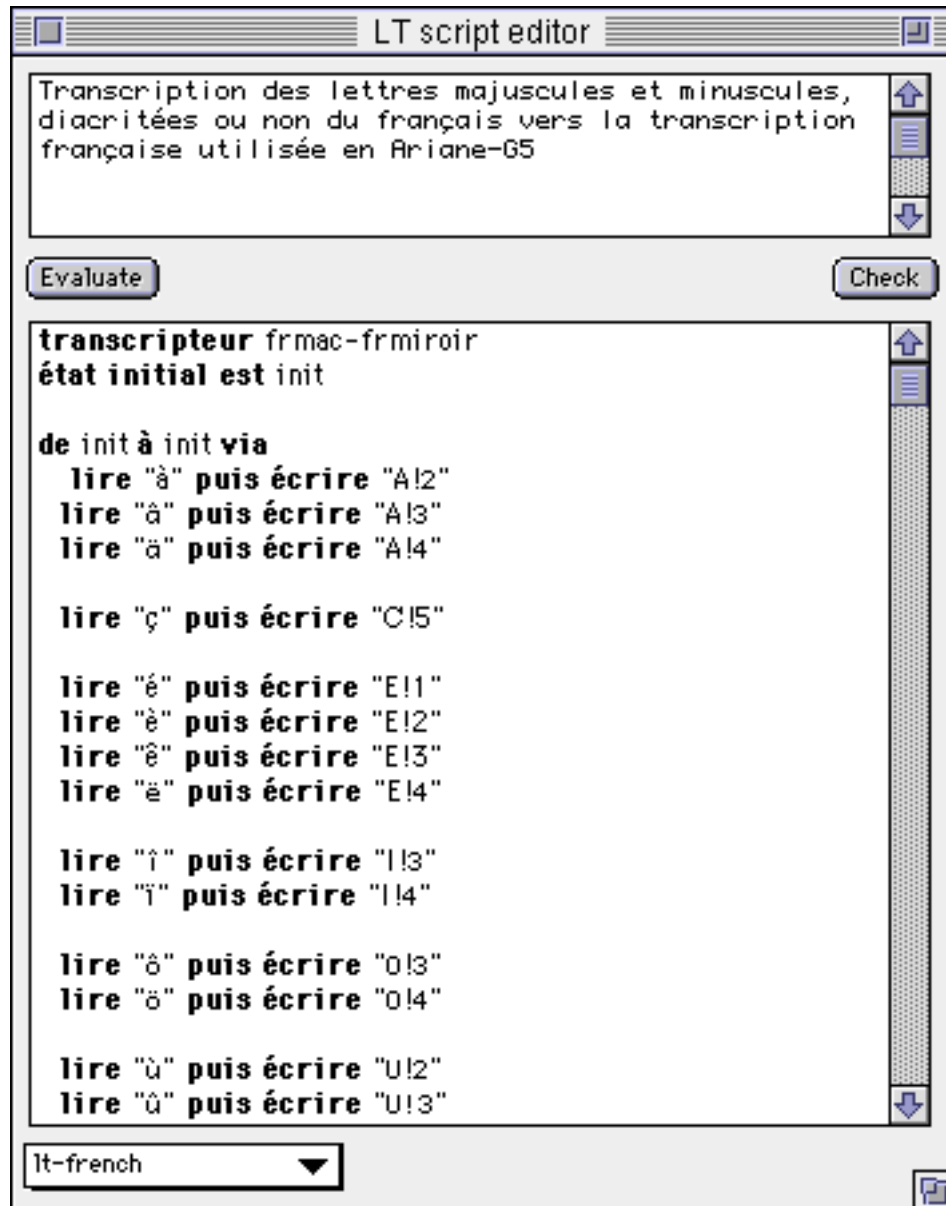


Figure 8.8 : Extrait du transcripteur
français_Macintosh → français_Ariane

8.3.3. Fichier miroir

Les fichiers miroirs contiennent toute l'information nécessaire au processus de traduction et à la construction des piles cibles.

Nous distinguons les *informations statiques* et les *informations dynamiques*.

Les informations statiques sont celles attachées par HyperCard à chaque objet. Les informations dynamiques sont celles utilisées par LIDIA pour traduire le contenu d'un objet.

Informations statiques :

- nom de la pile,

- ID⁵³ de la carte où se trouve l’objet,
- ID d’objet,
- type d’objet (champ ou bouton),
- visibilité de l’objet.

Informations dynamiques :

- style d’énoncé/genre de texte,
- préférences d’interaction associée à l’objet,
- traitements à effectuer,
- langue(s) cible(s),
- étape courante du traitement,
- information textuelle sous la forme directement manipulée par HyperCard,
- transcription du texte, avec les annotations éventuelles,
- structure mmc sous forme lisp,
- structure umc source sous forme lisp,
- résultat(s) de traduction,
- uma-structure(s) cible(s) et rétrotraduction(s) éventuelle(s).

Voici un exemple d’un tel fichier :

<u>Contenu du fichier :</u>	<u>Commentaire :</u>
	<i>Informations statiques Ø</i>
Stack : demo_LIDIA	<i>Nom de la pile</i>
Card_ID : 125	<i>ID de la carte</i>
Object_ID : 385	<i>ID de l’objet</i>
Object_type : field	<i>Type de l’objet</i>
Visible : true	<i>Visibilité de l’objet</i>
	<i>Informations dynamiques Ø</i>
Langues_cible : allemand	<i>Langues cible</i>
Orthographe : oui	
Terminologie : oui	
Termes_spéciaux_figés : oui	
Type_de_texte : oui	
Traduction : oui	

⁵³ HyperCard affecte un numéro dit “ID d’objet” (object ID number) à chaque objet d’une pile. Ce numéro est unique pour chaque type d’objet à l’intérieur de l’objet qui le contient. Ce numéro ne change jamais et, si l’objet est supprimé, n’est pas réaffecté à un objet nouvellement créé. LIDIA utilise ces IDs d’objet de façon interne.

	<i>Interaction Ø</i>
Annotation_manuelle : non	
Annotation_système : non	
Rétrotraduction : oui	
Etat_traitement : non	
	<i>Informations système Ø</i>
Type_de_texte : --sans type--	
Etape_courante : traduction	
Texte_origine :	
LE CAPITAINE A RAPPORTE!1 UN VASE DE *CHINE.	
Texte_annoté :	
(le.&d capitaine.&n).&subj (a.&v rapporté.&v).&nv (un.&d vase.&n de.&s Chine.&n).&obj (..&ponc)	
Forme_lisp :	
cf. Annexe 7	
Forme_lisp_choisie :	
<pre>((nil ""ULTXT"" nil) ((nil ""ULFRA"" nil) ((nil ""ULSOL"" nil) ((nil ""PHVB"" (K phvb CAT v ENONCP decl VOIX act PHASE acc MT ipr SUBV vf PERS 3 LINKS oui R echts oui NBR sing VALET q ARGS (a1 a0))) ((nil ""DGN"" (K gn FS suj RL arg0 CAT (d n) SUBN nc GNR mas PERS 3 NBR sing VALET n)) ("Le" ""LE"" (FS des CAT d GNR mas NBR sing TYPOG pcap))) (("capitaine" ""CAPITAINE"" (FS gov CAT n SUBN nc GNR mas PERS 3 NBR sing SENS (3 2 1)))))) ((nil ""NV"" (K phvb CAT v ENONCP decl VOIX act PHASE acc MT ipr SUBV vf PERS 3 LINKS oui R echts oui NBR sing VALET q ARGS (a1 a0))) (("a" ""AVOIR"" (FS aux CAT v MT ipr SUBV vf PERS 3 NBR sing SENS 1))) ("rapporté" ""RAPPORTER"" (FS gov CAT v SUBV ppa GNR mas PERS 3 NBR sing SENS (3 2 1)))) ((nil ""DGN"" (K gn FS obj RL arg1 CAT (d n) SUBN nc GNR mas PERS 3 RECHTS oui NBR sing VALET n)) ("un" ""UN"" (FS des CAT d GNR mas NBR sing))) ("vase" ""VASE/LE"" (FS gov CAT n SUBN nc GNR mas NBR sing SENS 1))) ((nil ""GP"" (K gn FS comp RS unde CAT (s n) SUBN np GNR fem PERS 3 NBR sing VALET den)) ("de" ""DE"" (FS reg CAT s))) ("Chine" ""*CHINE"" (FS gov CAT n SUBN np GNR fem NBR sing SENS 1)))) ("." ""."" (CAT p))))))</pre>	
Traduction_allemand	
Der Kapitaen hat eine chinesische Vase mitgebracht.	
Rétrotraduction_allemand	
Le capitaine a rapporté un vase chimois.	

Chapitre 9

Désambiguïsation

Le processus de désambiguïsation s'organise autour d'un programme de filtrage. Nous avons défini des classes de problèmes dont certaines sont associées à des patrons. À chaque patron est associée une méthode qui permet de produire le dialogue à partir des éléments mémorisés au moment de la superposition d'un filtre sur une donnée. L'ordre de recherche des patrons dans la structure produite par l'analyseur est défini par une stratégie.

9.1. Méta-classes et classes de problèmes

Pour résoudre les ambiguïtés que nous avons choisi de traiter (cf. § 6.2. Ambiguïtés retenues et choix d'un corpus), nous proposons une classification des problèmes retenus. Cette classification est fondée sur des critères plus pragmatiques que linguistiques.

Nous définissons quatre méta-classes d'ambiguïtés : la polysémie, l'ambiguïté de classe syntaxique, l'ambiguïté de géométrie et l'ambiguïté de décoration. Les trois dernières proviennent de notre étude de la structure mmc produite par l'analyseur.

Nous avons étudié une collection d'arbres en cherchant à les classer en fonction des techniques de désambiguïsation que nous pouvions et voulions mettre en œuvre. Dans un premier temps, nous avons distingué deux catégories selon que les analyses d'une phrase ont la même géométrie ou non.

Le groupe des analyses de même géométrie est lui-même divisé en trois classes de problèmes. Si les géométries sont différentes, alors : soit l'analyseur n'a pas réussi à affecter une classe syntaxique unique à une même occurrence dans les diverses analyses — on n'a pas déterminé de façon unique les groupes simples de la phrase —, soit les groupes simples sont déterminés de façon unique, mais on ne sait déterminer de façon unique l'attachement de ces groupes simples.

Comme les solutions que nous mettons en œuvre pour trouver les bons groupes simples ou trouver les relations entre des groupes simples bien définis sont très différentes, nous avons choisi de séparer les deux types de problèmes.

C'est ainsi que :

- la polysémie regroupe les polysémies interne et externe définies au § 6.1.2.a.,
- l'ambiguïté de classe syntaxique recouvre l'ambiguïté de classe syntaxique définie au § 6.1.2.b.,
- l'ambiguïté de géométrie est une sous-classe de l'ambiguïté de structure de constituants grammaticaux définie au § 6.1.3.b.,
- l'ambiguïté de décoration recouvre l'ambiguïté de relation logique et de relation sémantique définie au § 6.1.4.

Une étude plus détaillée de ces quatre types est indispensable pour la compréhension de ce qui suit.

9.1.1. Polysémie

La définition de la polysémie en LIDIA est la définition linguistique de la polysémie linguistique.

9.1.2. Ambiguïté de classe syntaxique

Il y a ambiguïté de classe si une même occurrence est étiquetée avec deux classes morpho-syntaxiques différentes dans deux arbres d'analyse distincts.

Nous distinguons deux classes de problèmes :

a. Coordination verbale

Si l'on a une ambiguïté de classe syntaxique qui apparaît pour un groupe d'occurrences qui appartiennent à un groupe coordonné et si, dans l'une des solutions, le groupe coordonné est un groupe verbal, alors nous parlerons d'ambiguïté de coordination verbale.

Le schéma suivant (figure 9.1) montre les deux arbres produits par l'analyse de la phrase "Il observe la photo et la classe".

Dans la première interprétation (arbre du haut), "classe" est le verbe classer conjugué à la troisième personne du présent de l'indicatif et son sujet est le sujet de la proposition principale de la phrase ("il") et "la" est un pronom qui reprend l'objet de la principale ("la photo") — Il observe la photo et il classe la photo.

Dans la seconde interprétation (arbre du bas), "et la classe" est un groupe nominal coordonné au verbe "observer" — Il observe la photo et il observe la classe.

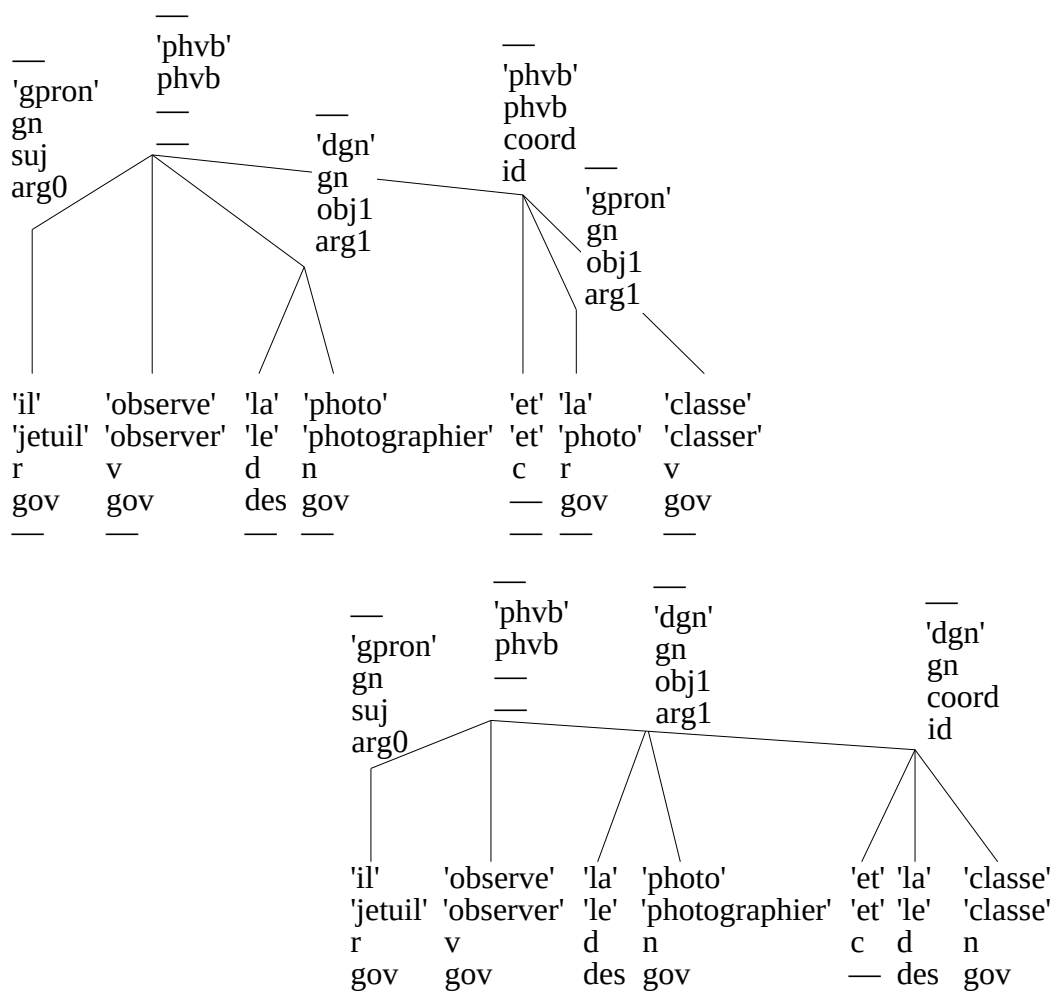


Figure 9.1 : Ambiguïté de coordination verbale

b. Classe simple

Toutes les ambiguïtés de classe qui ne sont pas des ambiguïtés de coordination verbale sont des ambiguïtés de classes simples.

Le schéma suivant (figure 9.2) montre les deux arbres produits par l'analyse de la phrase "Devant cette somme, il hésite".

"Devant" peut être interprété comme gérondif du verbe devoir (arbre du haut) — Il hésite parce qu'il doit cette somme — ou bien comme préposition (arbre du bas) — En face de cette somme, il hésite.

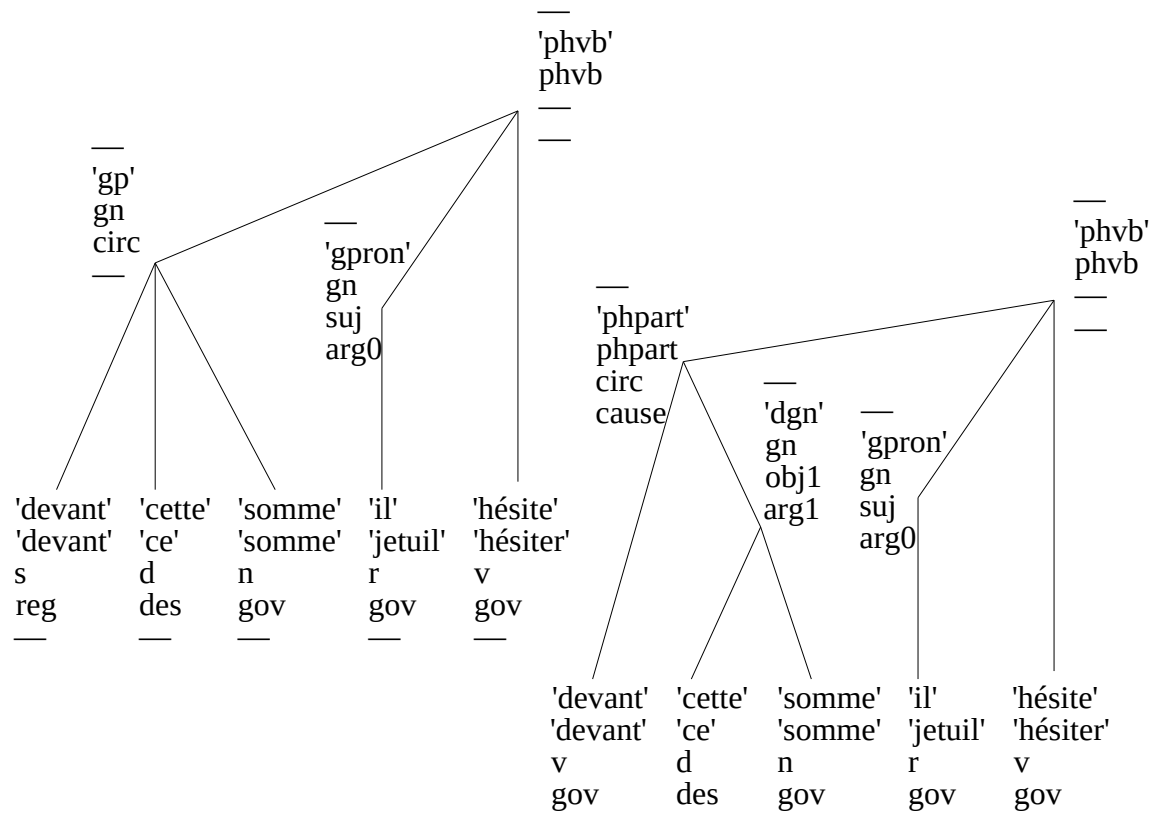


Figure 9.2 : Ambiguïté de classe simple

9.1.3. Ambiguïté de géométrie

Il y a ambiguïté de géométrie si deux arbres d'analyse différents ont des graphes distincts.

a. Structure argumentaire du verbe

Lorsque le verbe peut être complété de différentes façons et que l'un des groupes qui dépend du verbe peut être argument ou circonstant, nous parlons d'ambiguïté de structure argumentaire du verbe (on peut avoir ce même problème avec des géométries identiques, cf. infra). Par exemple, dans une analyse, le verbe peut être doté d'un sujet et d'un circonstant, dans une autre, d'un sujet et d'un objet, dans une troisième, d'un sujet, d'un objet, et d'un circonstant.

Le schéma suivant (figure 9.3) montre les deux arbres produits par l'analyse de la phrase "Il parle de l'école de cuisine lyonnaise". Le verbe parler peut avoir soit :

- un premier argument (arbre supérieur) — Il parle à propos de l'école de cuisine,
- un circonstant (arbre inférieur) — Il parle depuis l'école de cuisine,
- un circonstant et un argument (arbre médian) — Il parle depuis l'école à propos de cuisine.

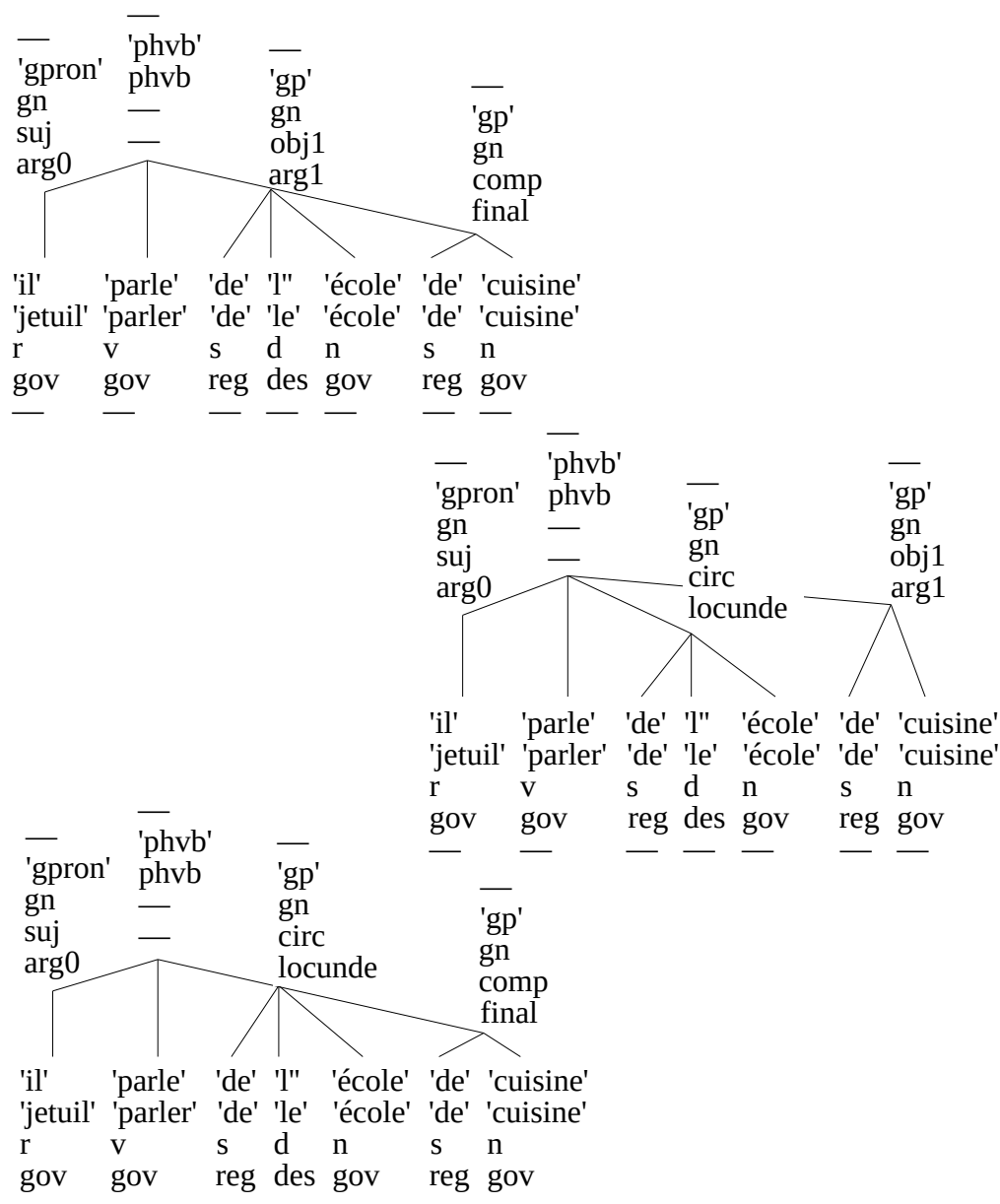


Figure 9.3 : Ambiguïté de structure argumentaire du verbe

b. Coordination non verbale

L'ambiguïté de coordination non verbale correspond à deux types de problèmes.

Coordination nominale

Dans le premier cas, un groupe nominal coordonné peut être subordonné de plusieurs façons aux groupes qui le précèdent dans la phrase.

Le schéma suivant (figure 9.4) montre ainsi les deux arbres produits par l'analyse de la phrase "L'évolution de la structure du réseau et des investissements est étudiée".

Le groupe nominal “et des investissements” peut être coordonné au groupe “de la structure” — l’évolution de la structure des investissements est étudiée (arbre du haut).

Dans l’autre cas, le groupe nominal “et des investissements” peut être coordonné au groupe “l’évolution” — l’évolution des investissements est étudiée (arbre du bas).

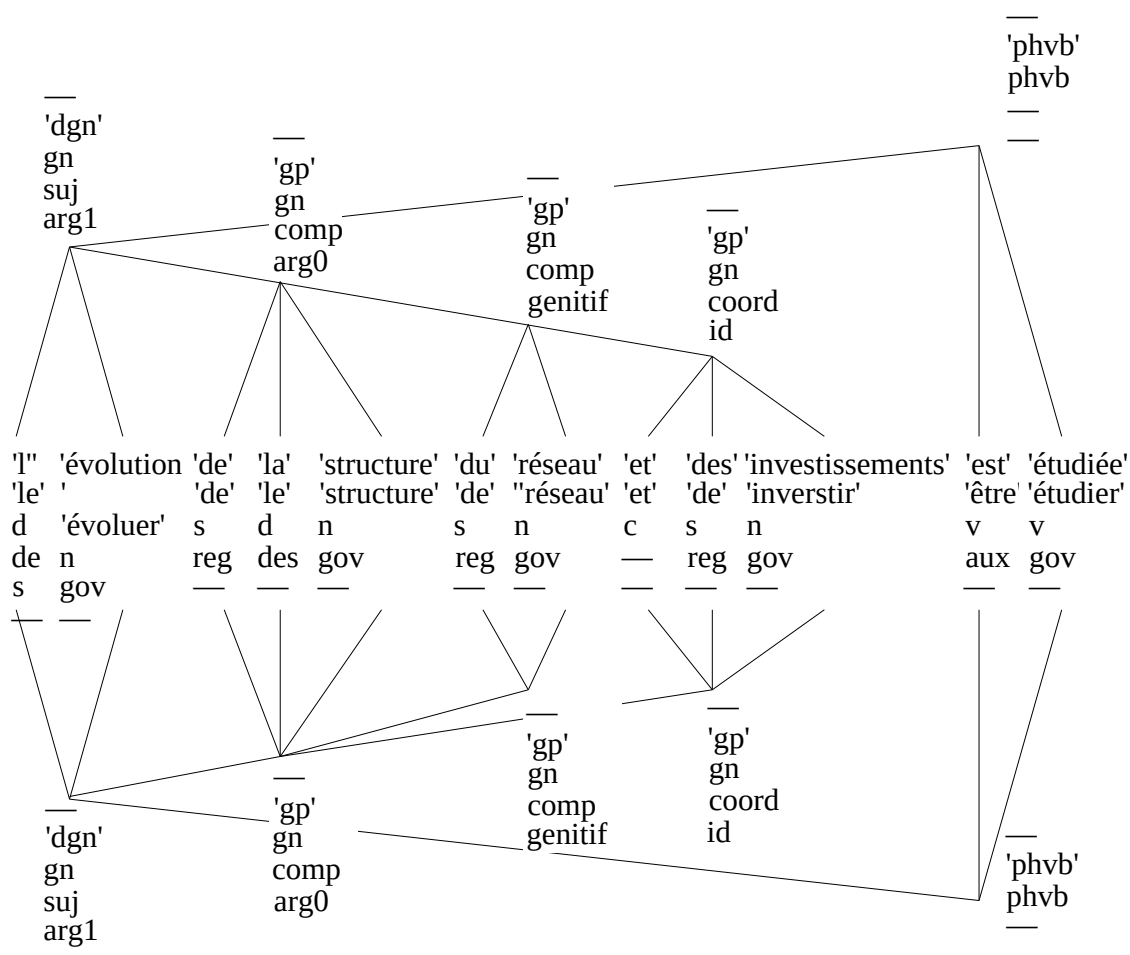


Figure 9.4 : Ambiguïté de coordination nominale

Coordination adjectivale

Dans le second cas, il faut connaître la distribution d’un groupe adjectival sur des groupes nominaux coordonnés.

Le schéma suivant (figure 9.5) montre ainsi les deux arbres produits par l’analyse de la phrase “Il prend les cahiers et les classeurs noirs”.

Dans le premier cas (arbre du haut), “noirs” s’applique seulement au groupe nominal “et les classeurs” — Il prend les classeurs noirs et les cahiers.

Dans le second (arbre du bas), “noirs” s’applique au deux groupes “les cahiers” et “et les classeurs” — Il prend les cahiers noirs et les classeurs noirs.

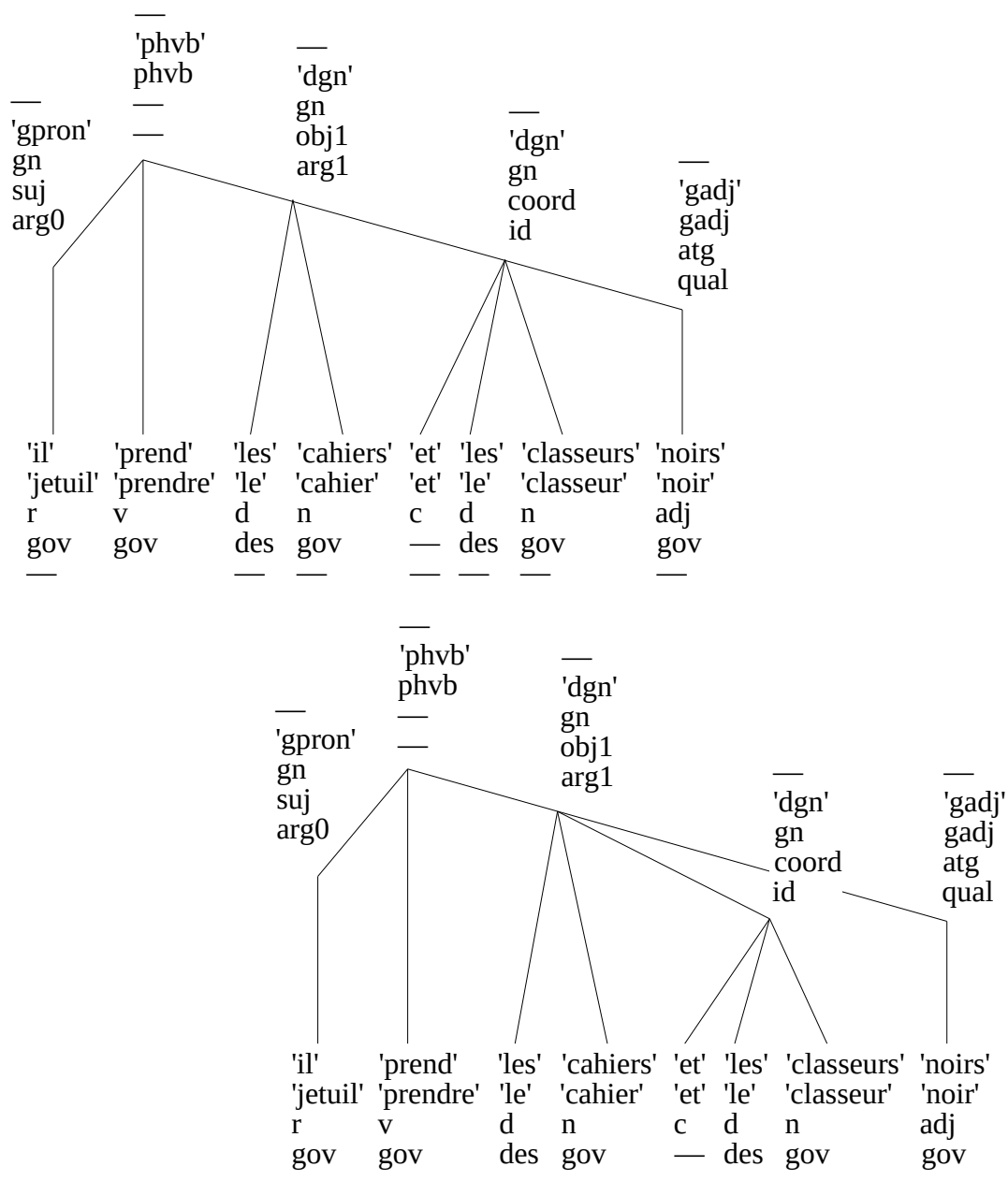


Figure 9.5 : Ambiguïté de coordination adjectivale

c. Subordination

Lorsqu'un groupe établit différents liens de subordination avec les groupes qui le précèdent⁵⁴ dans la phrase, nous parlons d'ambiguïté de subordination. Par exemple, il peut être subordonné au groupe qui lui est contigu immédiatement à gauche ou à un groupe non contigu qui le précède.

⁵⁴ Pour étudier la subordination nous regardons l'attachement de groupes simples en partant du dernier. C'est pourquoi nous parlons de lien de subordination d'un groupe simple avec ceux qui le précèdent.

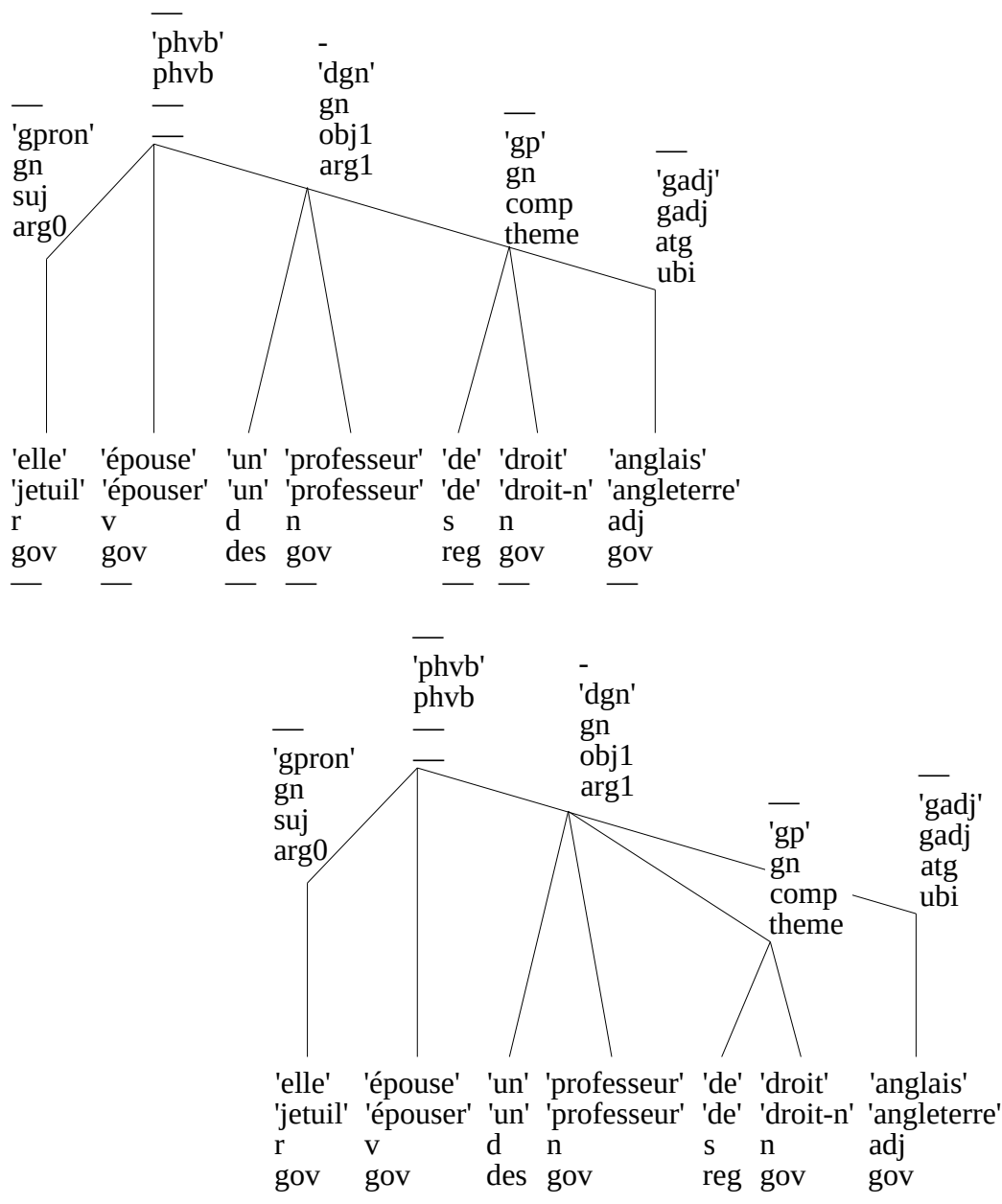


Figure 9.6 : Ambiguïté de subordination

Le schéma précédent (figure 9.6) montre ainsi les deux arbres produits par l'analyse de la phrase "Elle épouse un professeur de droit anglais".

Dans le premier cas (arbre du haut), "anglais" est rattaché au groupe nominal "de droit" — Elle épouse un professeur qui est spécialiste du droit anglais.

Dans le second (arbre du bas), "anglais" est rattaché au groupe nominal "un professeur" — Elle épouse un professeur anglais spécialiste de droit.

9.1.4. Ambiguïté de décoration

Il y a ambiguïté de décoration syntaxique, logique et sémantique si, pour des arbres ayant tous les mêmes groupes simples⁵⁵, l'étiquetage des nœuds non terminaux en fonctions syntaxiques et/ou en relations logiques et sémantiques est différent.

a. Étiquetage logico-sémantique

Lorsque les solutions d'analyse ont la même géométrie et que, pour un ou plusieurs groupe(s) simple(s), l'analyseur n'a pu calculer un unique étiquetage en relations logiques et/ou sémantiques, nous parlons d'ambiguïté d'étiquetage logico-sémantique.

Le schéma suivant (figure 9.7) montre ainsi les deux arbres produits par l'analyse de la phrase "Pierre fait porter des chocolats à Lucie".

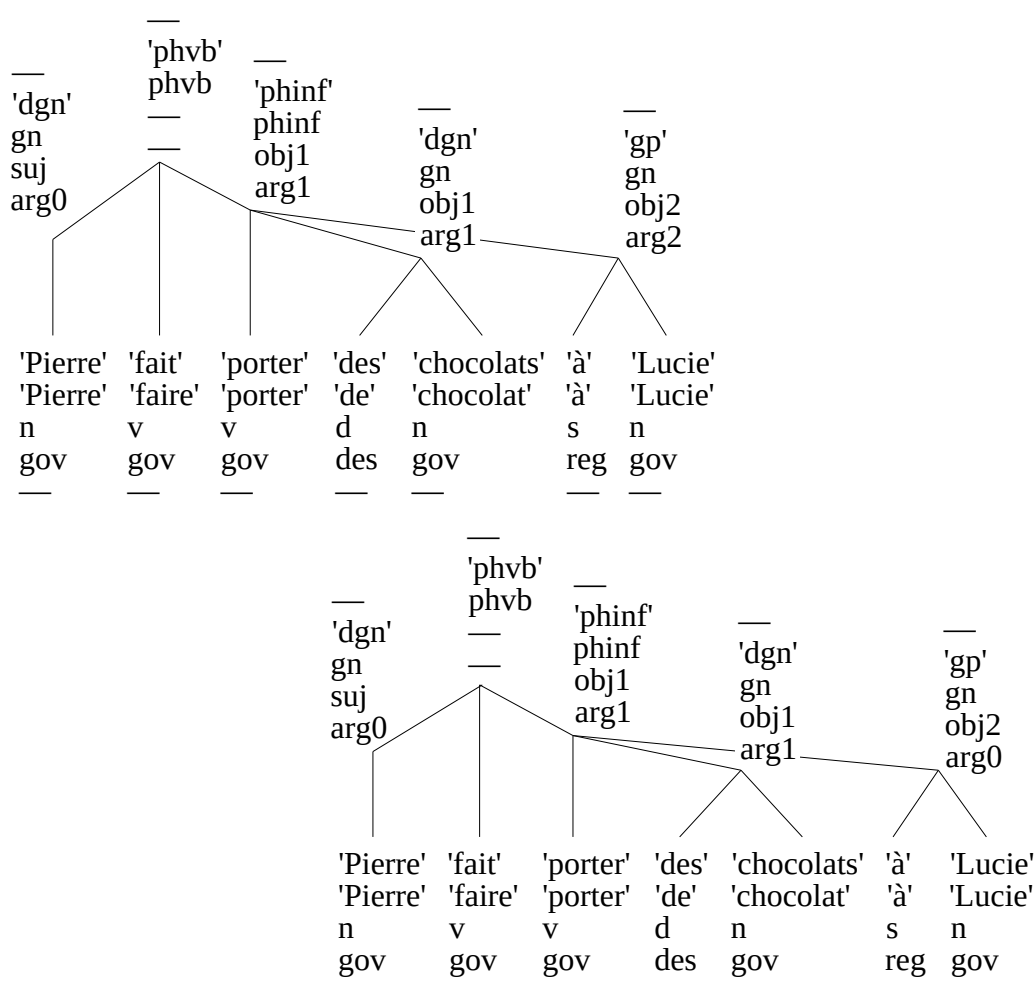


Figure 9.7 : Ambiguïté d'étiquetage logico-sémantique

⁵⁵ Un groupe simple est constitué de tout gouverneur accompagné de sa complémentation à gauche. Exemple "le chien", "le tout petit chien". "tout petit" est un groupe simple inclus. Un gouverneur est un lexème qui n'est pas un mot outil. Exemple : "le" et "tout" sont des mots outils.

Dans le premier cas (arbre du haut), le groupe nominal “à Lucie” est le troisième argument (noté arg2) du verbe “porter” — Lucie est la destinataire de chocolats envoyés par Pierre.

Dans le second (arbre du bas), le groupe nominal “à Lucie” est le premier argument (noté arg0) du verbe “porter” — Suivant les consignes de Pierre, Lucie porte des chocolats.

b. Étiquetage syntaxique

Lorsque les solutions d’analyse ont la même géométrie et que pour un, ou plusieurs, groupe(s) simple(s), l’analyseur n’a pu calculer un unique étiquetage en fonction syntaxique, nous parlons d’ambiguïté d’étiquetage syntaxique.

Le schéma suivant (figure 9.8) montre ainsi les deux arbres produits par l’analyse de la phrase “Je vous parle de la Tour Eiffel”.

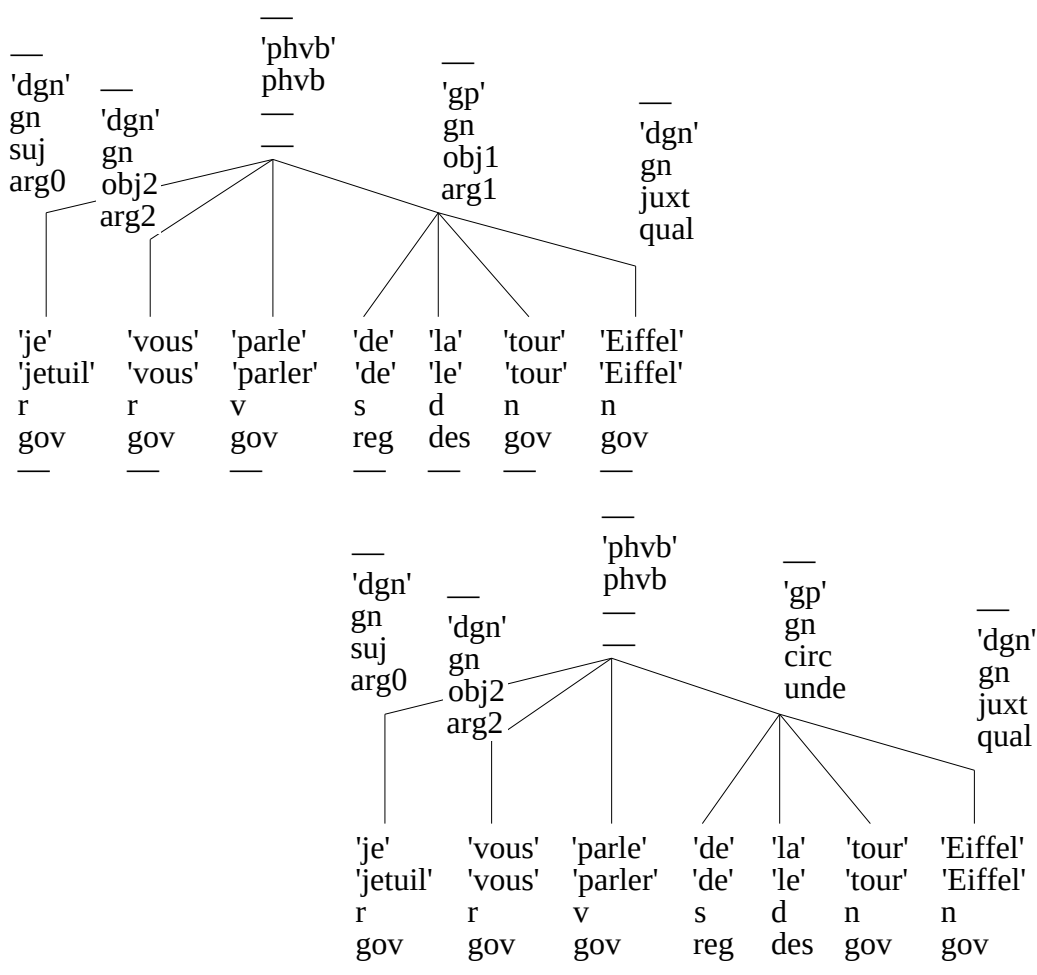


Figure 9.8 : Ambiguïté d’étiquetage syntaxique

Dans le premier cas (arbre du haut), le groupe nominal “de la Tour Eiffel” est le premier objet du verbe “parler” — Je vous parle à propos de la Tour Eiffel.

Dans le second (arbre du bas), le groupe nominal “de la Tour Eiffel” est un circonstant du verbe “parler” — Je vous parle depuis la Tour Eiffel.

c. Ordre des arguments d'un verbe transitif direct

Il y a ambiguïté d'ordre des arguments d'un verbe transitif direct lorsque la phrase comporte un verbe transitif direct pour lequel l'analyseur n'a pas pu calculer la place exacte.

Le schéma suivant (figure 9.9) montre ainsi les deux arbres produits par l'analyse de la phrase “Quel auteur cite ce conférencier ?”.

Dans le premier cas (arbre du haut), le groupe nominal “quel auteur” est le sujet du verbe “citer” et le groupe nominal “ce conférencier” est le premier objet du verbe “citer” — Un auteur parle de ce conférencier.

Dans le second (arbre du bas), le groupe nominal “quel auteur” est le premier objet du verbe “citer” et le groupe nominal “ce conférencier” est le sujet du verbe “citer” — Ce conférencier parle d'un auteur.

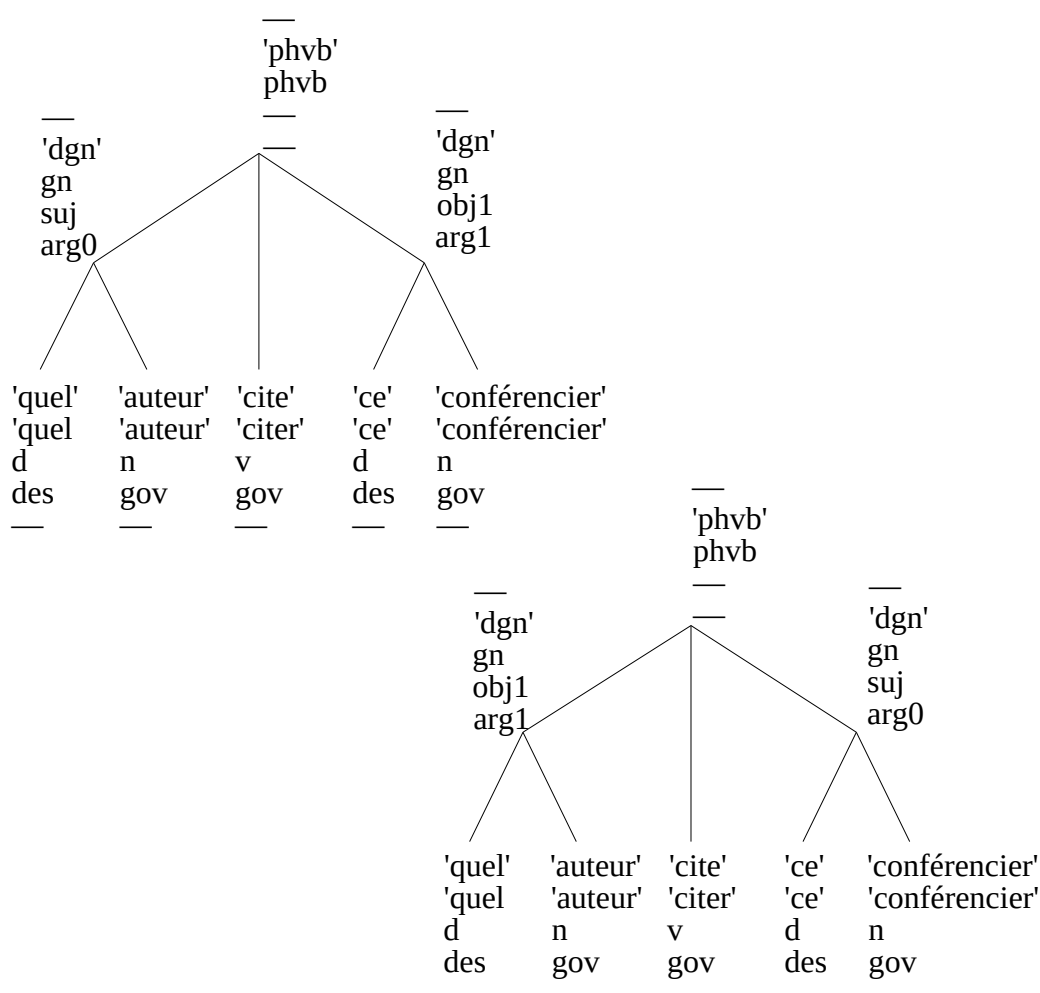


Figure 9.9 : Ambiguïté d'ordre des arguments d'un verbe transitif direct

9.2. Idées directrices

Afin de mettre en place le processus de désambiguïsation, nous avons dégagé des principes, une stratégie, des méthodes et des opérateurs. Nous avons procédé de cette façon pour rendre le processus de désambiguïsation le plus modulaire possible, pouvoir le construire de façon incrémentale et modifier facilement, si nécessaire, chaque étape du processus. Notre objectif à long terme est de proposer, sur la base du travail effectué, un environnement de description de processus de désambiguïsation afin de permettre une expérimentation plus complexe.

9.2.1. Principes

Les trois principes suivants fondent la stratégie décrite plus bas. Les voici, par ordre d'importance :

- 1 trouver d'abord les bons groupes simples,
- 2 trouver ensuite le sujet, les arguments et les circonstants de tous les prédicats,
- 3 trouver enfin la structure des arguments et des circonstants des prédicats.

Des considérations pragmatiques nous ont conduit à définir ces principes.

- Nous voulons, avant toute autre chose, trouver le bon découpage de la phrase en groupes simples. Ceux-ci forment, en effet, les éléments de base qui permettent de produire le sens.
- Nous voulons ensuite choisir la construction de la phrase au plus haut niveau, c'est-à-dire au niveau des prédicats.
- Finalement, ce n'est que lorsqu'on a construit correctement les prédicats que l'on veut s'intéresser à la structure interne des éléments qui leur sont attachés.

Ces critères nous semblent raisonnables et naturels. De plus, l'ordre des questions ne sera pas modifié pour améliorer la stratégie. La compréhensibilité du système en est accrue.

9.2.2. Stratégie

Comme plusieurs ambiguïtés peuvent apparaître dans un résultat d'analyse, elle sont levées dans un ordre qui respecte les principes que nous avons posés.

Les groupes simples étant pour nous les briques de la phrase, il est nécessaire de les obtenir le plus rapidement possible. On sait que, si pour toutes les occurrences d'une phrase, les classes syntaxiques associées à chacune d'elles dans les diverses solutions d'analyse produites sont les mêmes, alors ces solutions sont composées des mêmes groupes simples. Pour obtenir les bons groupes simples, nous commençons donc par résoudre, si elles se présentent, les ambiguïtés de classe simple et de coordination verbale.

Lorsqu'on dispose des bons groupes simples, on peut chercher à construire les arguments et les circonstants de tous les prédicats. On constate que, si on a une ambiguïté de décoration, le verbe est toujours impliqué. Nous cherchons donc ensuite à traiter ces ambiguïtés. Faisons remarquer au passage qu'une ambiguïté de décoration peut apparaître dans une proposition subordonnée qui est une phrase verbale. Dans la phrase suivante : "Je reviens de l'école de cuisine lyonnaise et je vous parle de la Tour Eiffel.", la

proposition coordonnée “et je vous parle de la Tour Eiffel” présente une ambiguïté de décoration alors que la phrase prise en entier présente aussi une ambiguïté de géométrie dans la proposition principale. Nous appellerons ce type de problème une “ambiguïté de décoration locale”.

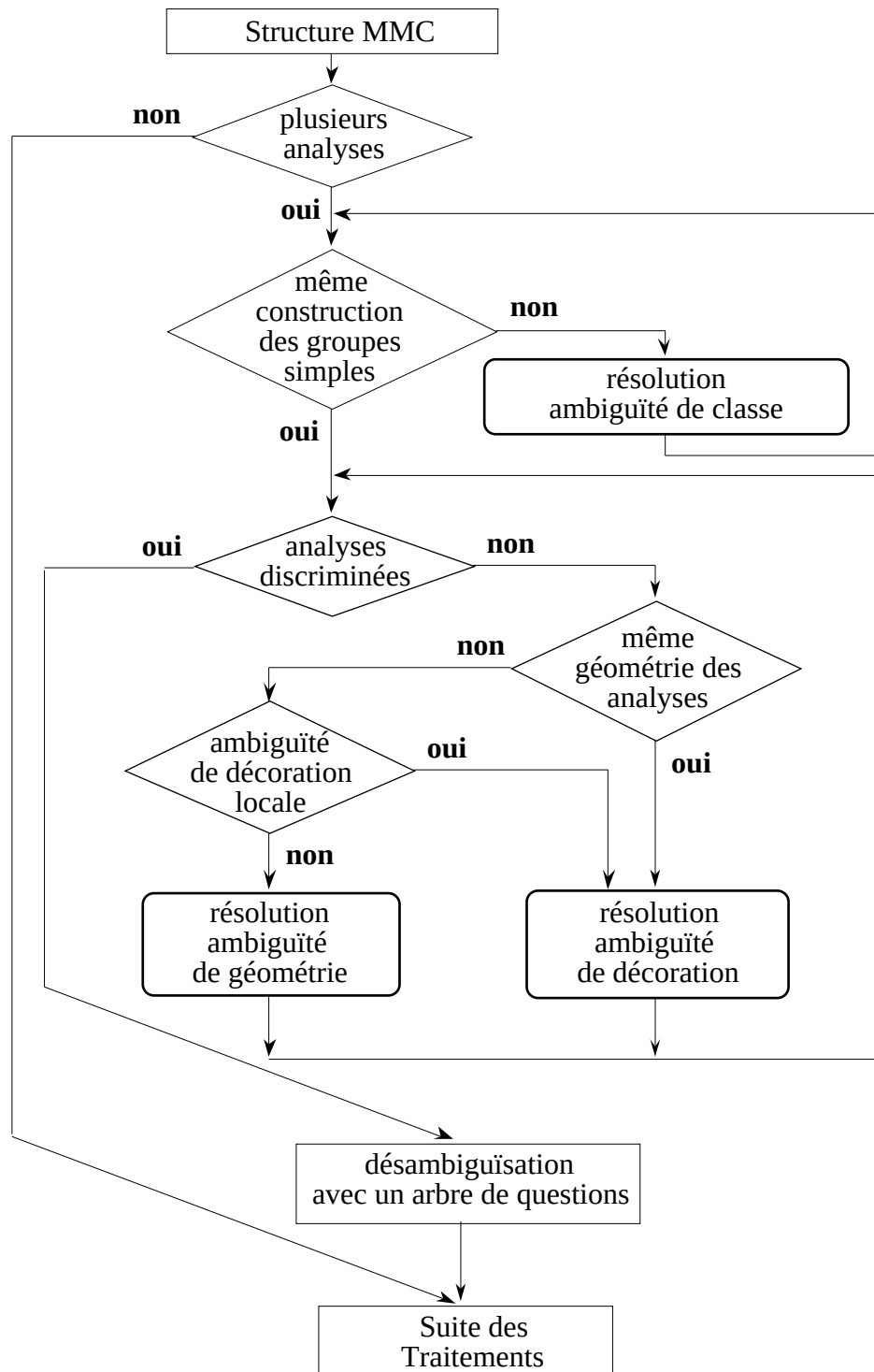


Figure 9.10 : Stratégie de production de l’arbre de questions

Nous itérons ensuite une série de tests et d’actions jusqu’à produire un arbre de questions permettant de désambiguïser totalement la phrase. Chaque action a pour rôle de produire une partition de l’ensemble des arbres qu’elle reçoit et engendre une question qui

permet de choisir la bonne classe parmi les classes créées. Chaque classe ainsi créée est une nouvelle donnée. Lors de l'itération suivante, on fabrique si nécessaire de nouvelles questions et de nouvelles classes. L'arbre de questions est totalement produit lorsque la partition de l'ensemble des analyses initialement produites ne comporte que des classes à un seul élément.

La stratégie de production de l'arbre de questions est résumée dans la figure 9.10. La figure 9.11 montre le résultat du processus pour une phrase fictive.

Les analyses A 1, A 2 et A 3 ont le même étiquetage en classes syntaxiques, A 1 et A 2 ont le même étiquetage en fonctions syntaxiques mais ont des étiquetages en relations logico-sémantiques différents ; A 3, lui, a un autre étiquetage en fonctions syntaxiques. De même, les analyses A 4 et A 5 ont le même étiquetage en classes syntaxiques, mais des étiquetages différents en relations syntaxiques.

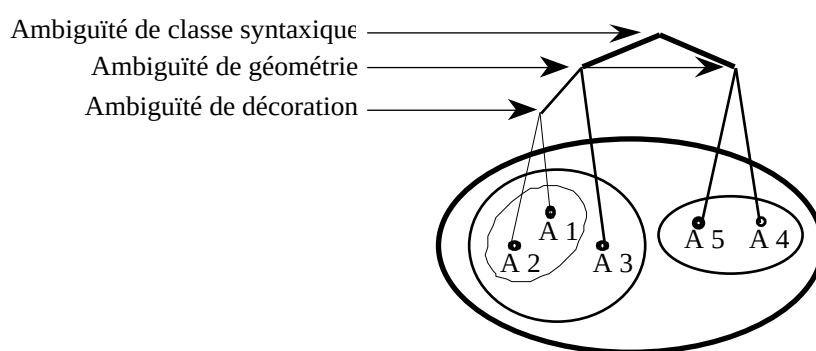


Figure 9.11 : Arbre de désambiguïsation

Examinons la phrase “Le professeur de l'école de cuisine lyonnaise parle de la Tour d'Argent, prend le menu et le lit⁵⁶.”

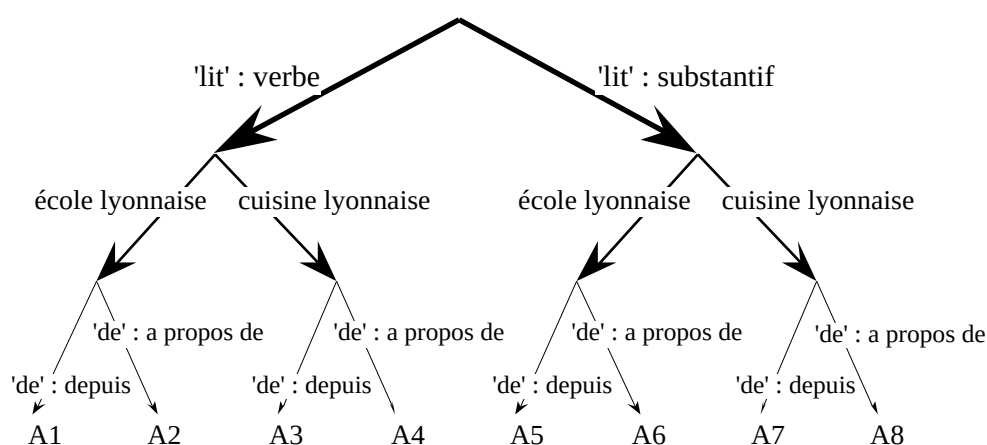


Figure 9.12 : Arbre de désambiguïsation concret

Voici les interprétations correspondant aux huit lectures de cette phrase :

⁵⁶ Cette phrase a seulement la prétention d'être un exemple illustratif.

- A1 Le professeur de l'école lyonnaise de cuisine parle depuis la Tour d'Argent, il prend et lit le menu.
- A2 Le professeur de l'école lyonnaise de cuisine parle à propos de la Tour d'Argent, il prend et lit le menu.
- A3 Le professeur de l'école où l'on enseigne la cuisine lyonnaise parle depuis la Tour d'Argent, il prend et lit le menu.
- A4 Le professeur de l'école où l'on enseigne la cuisine lyonnaise parle à propos de la Tour d'Argent, il prend et lit le menu.
- A5 Le professeur de l'école lyonnaise de cuisine parle depuis la Tour d'Argent, il prend le menu et prend le lit.
- A6 Le professeur de l'école lyonnaise de cuisine parle à propos de la Tour d'Argent, il prend le menu et prend le lit.
- A7 Le professeur de l'école où l'on enseigne la cuisine lyonnaise parle depuis la Tour d'Argent, il prend le menu et prend le lit.
- A8 Le professeur de l'école où l'on enseigne la cuisine lyonnaise parle à propos de la Tour d'Argent, il prend le menu et prend le lit.

9.2.3. Techniques, opérateurs et méthodes

Le dialogue de désambiguïsation est produit en utilisant quelques techniques simples mises en pratique avec un ensemble d'opérateurs de base. À chaque classe d'ambiguïtés est finalement associée une méthode qui permet de produire le dialogue souhaité.

a. Techniques

Aucune des techniques que nous avons retenues pour générer le dialogue n'utilise de générateur complet du français. Ce genre d'outils complexe ne convient à notre avis pas à la tâche.

En effet, il faudrait garantir que toutes les phases produites par le générateur — comme paraphrases de la phrase origine — soient différentes les unes des autres, mais aussi de la phrase d'origine.

Si on remplissait cette première contrainte, il faudrait, de plus, garantir que les paraphrases produites ne soient pas ambiguës, pour que l'utilisateur puisse choisir en connaissance de cause.

Plutôt que d'essayer de satisfaire ces contraintes, nous avons choisi des techniques beaucoup plus simples, mais efficaces :

- affichage de définitions du dictionnaire (celles qui sont dans la base Parax),
- affichage de la classe d'une occurrence,
- reconstitution des éléments élidés,
- permutation et parenthésage de l'énoncé original,
- remplacement sémantique permettant de discriminer les différents sens possibles.

b. Opérateurs

Pour produire les dialogues, nous utilisons les opérateurs de base suivants, qui opèrent sur des fragments de la mmc-structure :

Accordé produit une forme fléchiée d'un adjectif en fonction du genre et du nombre.

ex :	Accordé("noirs", (("gn" [...] (gnr fem nbr plu)) ([...])) -> "noires"
------	---

Texte produit le texte qui correspond aux arbres linguistiques passés en paramètres.

Sujet produit le texte du sujet des arbres linguistiques passés en paramètres. Il y a une telle fonction pour chaque classe syntagmatique et pour chaque fonction syntaxique : exemples : NoyauVerbal, Circonstant...

Coordonnant produit le coordonnant de l'arbre linguistique passé en paramètre

Moins_Coord produit le texte de l'arbre linguistique passé en paramètre auquel on a ôté le coordonnant.

Moins_Reg produit le texte de l'arbre linguistique passé en paramètre auquel on a ôté le régisseur.

Distribue distribue une occurrence ou un groupe d'occurrences sur d'autres groupes d'occurrences, et établit un lien entre les résultats des distributions locales.

ex :	Distribue((A, B C, D), ou, (1, 2), (1, 3)) -> A B C ou A D
------	--

Substitue substitue à un terme un autre terme recherché dans la BDLM (Parax) en fonction de conditions.

ex :	Substitue("de", #Objet_1) -> "à propos de"
------	--

Projette appliqué à un nœud de l'arbre, projette la classe syntaxique, et, en fonction de cette classe, certaines informations contenues dans Parax.

Parenthèse construit un texte unique à partir des paramètres et parenthèse ce texte.

ex :	Parenthèse("classeurs", "noirs") -> "(classeurs noirs)"
------	---

Définition produit la définition d'une acception donnée dans la base Parax.

c. Méthodes, vue informelle

Voici une vue informelle de la méthode de résolution que nous avons associée à chaque problème. On trouvera le résultat du processus au § 9.3.3.

i. Polysémie

Pour résoudre une ambiguïté de polysémie, nous montrons à l'utilisateur la définition de chaque sens retenu par l'analyseur.

ii. Ambiguïté de classe syntaxique

Coordination verbale

Pour résoudre un tel problème, nous produisons, par *permutation*, une paraphrase de chaque analyse qui met en valeur les membres de phrase qui ont pu être élidés. Dans ce cas, deux types d'élision sont possibles :

Si le groupe coordonné est constitué d'un groupe nominal, ce groupe nominal est l'objet du dernier verbe rencontré dans la phrase, le sujet de ce verbe restant inchangé.

Pour la phrase "Il prend la boule et la lance", on obtient "il prend la lance".

Si le groupe coordonné est constitué d'un pronom et d'un verbe, le sujet et l'objet de ce verbe ont été élidés. L'objet élidé repris par le pronom est l'objet du dernier verbe rencontré dans la phrase. Le sujet élidé est le sujet du dernier verbe rencontré dans la phrase.

Pour la phrase "Il prend la boule et la lance", on obtient "il lance la boule".

Classe simple

Pour résoudre une ambiguïté de classe, on *projette* la classe syntaxique de chaque occurrence ambiguë.

Pour la phrase "Le pilote ferme la porte", on obtient "le pilote(nom) ferme(adjectif) la(pronom) porte(verbe)." et "le pilote(nom) ferme(verbe) la(article) porte(nom)."

iii. Ambiguïté de géométrie

Structure argumentaire du verbe

Pour résoudre un problème de structure argumentaire du verbe, on projette la partie de la phrase qui permet de distinguer les différentes possibilités avec *remplacement* des prépositions qui introduisaient ces groupes par des prépositions non ambiguës.

Pour la phrase "Il parle de l'école de cuisine" on obtient "Il parle depuis l'école.", "il parle à propos de l'école" et " Il parle depuis l'école à propos de cuisine."

Coordination non verbale

Pour résoudre un problème de coordination non verbale, on distribue ce qui doit l'être afin de retrouver les interprétations sous une forme complètement explicite. Deux cas peuvent se produire :

Dans le premier cas, coordination nominale, c'est un groupe nominal coordonné qui peut être attaché de plusieurs façons aux groupes qui le précèdent dans la phrase.

Pour la phrase "l'évolution de la structure du réseau et des investissements est étudiée.", on obtient "l'évolution de la structure du réseau et l'évolution des investissements" ou "l'évolution de la structure du réseau et l'évolution de la structure des investissements".

Dans le second cas, coordination adjectivale, il faut connaître la distribution d'un groupe adjectival sur des groupes nominaux coordonnés.

Subordination

Dans ce cas, on rapproche par permutation les groupes subordonnés qui ont été séparés par un groupe intermédiaire. On se sert de parenthèses pour bien montrer les mots que l'on a regroupés.

Pour la phrase “L'école de cuisine lyonnaise ouvre ses portes.”, on obtient “l'école de (cuisine lyonnaise)” ou “l'(école lyonnaise) de cuisine”.

iv. Ambiguïté de décoration

Étiquetage logico-sémantique

Pour résoudre un problème d'étiquetage logico-sémantique, il suffit de remplacer la préposition introduisant le groupe à problème par les prépositions non ambiguës qui représentent chacune des relations possibles.

Pour la phrase “Pierre fait porter des chocolats à Lucie.”, on obtient “pour Lucie” ou “par Lucie”.

Étiquetage syntaxique

Pour résoudre un problème d'étiquetage syntaxique, il suffit de remplacer la préposition introduisant le groupe à problème par les prépositions non ambiguës qui représentent chacune des relations possibles.

Pour la phrase “Je vous parle de la Tour Eiffel.”, on obtient “parler depuis la Tour” ou bien “parler à propos de la Tour”.

Ordre des arguments d'un verbe transitif direct

On reconstruit par permutation toutes les interprétations selon un ordre “standard” sous une forme normalisée. On ne conserve que le gouverneur de chaque groupe sujet ou objet, et on le munit du bon article avant de faire les projections dans l'ordre sujet + verbe + objet 1 + objet 2.

Pour “quel hôtel gère ce bureau ?”, on obtient “un hôtel gère un bureau” ou bien “un bureau gère un hôtel”.

9.3. Mise en œuvre

Dans ce dernier paragraphe, nous expliquons comment est produit le dialogue de désambiguïsation. Nous concluons en montrant des exemples de dialogues obtenus.

9.3.1. Filtres et filtrage

Le processus de désambiguïsation est maintenant vu comme un processus de filtrage. La stratégie décrite dans le paragraphe 9.2.2. est mise en place. La structure produite par l'analyseur mmc est soumise à deux niveaux de filtres. Au premier niveau de filtrage, on cherche une méta-classe d'ambiguïté, au second, on cherche une classe de problèmes.

a. Schéma général

Les ambiguïtés de géométrie et de coordination verbale correspondent à des modèles (formes) facilement identifiables. Pour les repérer, nous utilisons une technique de filtrage sur les arbres. Cela nous a conduit à définir un langage de description de filtres et le programme associé pour vérifier si un filtre se superpose à une donnée.

Les ambiguïtés de classe simple sont repérées par discrimination : ce sont en effet les ambiguïtés de classe syntaxique qui ne sont pas des ambiguïtés de coordination verbale.

Les ambiguïtés de décoration sont reconnues par filtrage sur des listes.

La polysémie est résolue lorsque les autres problèmes ont été résolus.

b. Langage de description des filtres d'arbres

Voici une grammaire sous forme BNF de notre langage de filtres.

i. Grammaire

```

patron ::= variable      |
        ; reconnaît toute expression
        constante      |
        ; reconnaît seulement l'atom
        patron-segment |
        ; reconnaît une séquence
        patron-simple  |
        ; reconnaît une expression
        (patron . patron) |
        ; reconnaît le premier patron puis le reste

patron-simple ::= (? i s variable pred args)      |
                ; vérifie le prédicat pred (cf ii. commentaires sur la grammaire)
                (? or patron...)                  |
                ; reconnaît l'un des patrons sur une expression
                (? and patron...)                  |
                ; reconnaît tous les patrons sur une expression
                (? not patron...)                  |
                ; réussit si le(s) patron(s) ne sont pas reconnus

```

```
patron-segment ::= ((? * variable) ...)      |
                  ;reconnaît zéro ou plusieurs expressions
                  ((? + variable) ...)      |
                  ; reconnaît une ou plusieurs expressions
                  ((? ? variable) ...)      |
                  ; reconnaît zéro ou une expression
                  ((? if expression) ...)
                  ; teste si l'expression (qui peut contenir des variables) est vraie
```

```
variable ::= ? caractère+
           ; un ou plusieurs caractères précédés de ?
```

```
constante ::= atom
            ; un atome
```

ii. Commentaires sur la grammaire et les filtres

Pour le patron-simple “(?is variable pred args)” le prédicat “pred” est appliqué sur les arguments “variable” et “args” soit : pred (variable args)

iii. Exemples

```
;;;-----
;;;
;;; Title      : Pattern-matcher
;;;
;;;-----
;;; Author     : Mathieu Lafourcade
;;; Copyright  : 1994 GETA, All Rights Reserved
;;; Availability: SAMBRE & LIDIA projects
;;;-----
;;; Filename   : pat-match.lisp
;;; Version    : 1.0
;;; Examples   : (pat-match '(a (?* ?x) d) '(a b c d))
;;;             : -> ((?X B C))
;;;             : (pat-match '(a (?* ?x) (?* ?y) d) '(a b c d))
;;;             : -> ((?Y B C) (?X))
;;;             : (pat-match '(a (?* ?x) (?* ?y) ?x ?y) '(a b c d (b c) (d)))
;;;             : -> ((?Y D) (?X B C))
;;;             : (pat-match '(?x ?op ?y is ?z (?if (eql (?op ?x ?y) ?z))) '(3 + 4 is 7))
;;;             : -> ((?Z . 7) (?Y . 4) (?OP . +) (?X . 3))
;;;             : (pat-match '(?x ?op ?y (?if (?op ?x ?y))) '(4 > 3))
;;;             : -> ((?Y . 3) (?OP . >) (?X . 4))
```

```

;;;      : (setf (get 'lidia-patterns 'cv-1)
;;;      :      '(((is ?x node-prop-equal-p 'CS 'PHVB)
;;;      :      (?* ?w)
;;;      :      ((is ?y node-prop-equal-p 'FS 'OBJ) (?* ?p1))
;;;      :      ((is ?z node-prop-equal-p 'CS 'PHVB) (?* ?p2))
;;;      :      ?punc
;;;      :      )
;;;      :      )
;;;      :
;;;      : (match (get 'lidia-patterns 'cv-1)
;;;      :      (get-solution (get 'lidia-test 'boule) 2)
;;;      :      )
;;;      : -->
;;;      : (
;;;      : (?PUNC (". " ". " (CAT P)))
;;;      : (?P2 ((et "ET" (FS REG CAT C))) ((NIL "GPRON" (K GN FS OBJ RL ARG1
;;;      : CAT R GNR FEM PERS 3 NBR SING VALET N)) (("la" "BOULE" (FS GOV CAT
;;;      : (R N) GNR FEM PERS 3 NBR SING SENS 1)))) ((NIL "NV" (K PHVB FS COORD
;;;      : RL ID CAT V ENONCP DECL VOIX ACT MT IPR PHASE NONACC SUBV VF PERS 3
;;;      : LINKS OUI NBR SING VALET Q ARGS A1)) ("lance" "LANCER" (FS GOV CAT V
;;;      : MT IPR SUBV VF PERS 3 NBR SING SENS (6 5 4 3 2 1))))))
;;;      : (?Z NIL "PHVB" (K PHVB FS COORD RL ID CAT V ENONCP DECL VOIX ACT MT IPR
;;;      : PHASE NONACC SUBV VF PERS 3 LINKS OUI NBR SING VALET Q ARGS A1))
;;;      : (?P1 (("la" "LE" (FS DES CAT D GNR FEM NBR SING))) (("boule" "BOULE"
;;;      : (FS GOV CAT N SUBN NC GNR FEM NBR SING SENS 1))))
;;;      : (?Y NIL "DGN" (K GN FS OBJ RL ARG1 CAT (D N) SUBN NC GNR FEM PERS 3
;;;      : NBR SING VALET N))
;;;      : (?W ((NIL "GN" (K GN FS SUJ RL ARG0 CAT N SUBN NP GNR MAS PERS 3
;;;      : NBR SING VALET N)) ("Pierre" "PIERRE" (FS GOV CAT N SUBN NP GNR MAS
;;;      : PERS 3 NBR SING SENS 1)))) ((NIL "NV" (K PHVB CAT V ENONCP DECL
;;;      : VOIX ACT MT IPR PHASE NONACC SUBV VF PERS 3 LINKS OUI RECHTS OUI
;;;      : VCOOR OUI NBR SING VALET Q ARGS (A1 A0))) ("prend" "PRENDRE" (FS GOV
;;;      : CAT V MT IPR SUBV VF PERS 3 NBR SING SENS (4 2 1))))))
;;;      : (?X NIL "PHVB" (K PHVB CAT V ENONCP DECL VOIX ACT MT IPR PHASE NONACC
;;;      : SUBV VF PERS 3 LINKS OUI RECHTS OUI VCOOR OUI NBR SING VALET Q
;;;      : ARGS (A1 A0)))
;;;      : )
;;;
-----

```

c. Filtres pour la désambiguïsation

Nous examinons maintenant les filtres que nous avons définis pour effectuer la désambiguïsation.

Pour reconnaître une ambiguïté, il faut qu'un ensemble de plusieurs filtres (généralement deux) puissent chacun se superposer simultanément à une solution d'analyse et que les variables partagées par les filtres couvrent le même mot des feuilles.

Nous présentons ici une collection de filtres . Nous associons à chaque ensemble de filtre une méthode pour produire le dialogue de désambiguïsation.

i. Coordination verbale

Patron 1

```
(setf (get 'lidia-patterns 'cv-1)
      '(((is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1))
        ((is ?z node-prop-equal-p 'CS 'PHVB)
         (?* ?p2))
        ?punc )))
```

et

Patron 2

```
(setf (get 'lidia-patterns 'cv-2)
      '(((is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1))
        ((is ?z node-prop-equal-p 'FS 'COORD)
         (?* ?p2)))
        ?punc )))
```

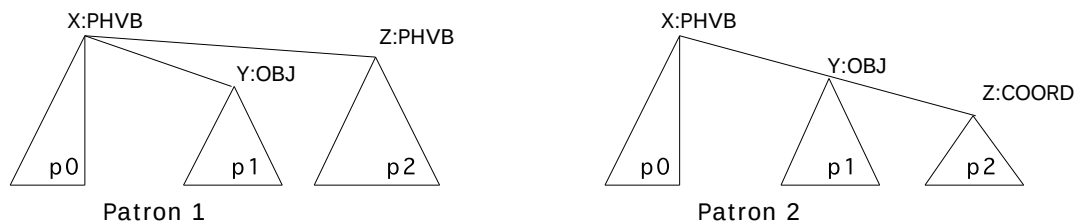


Figure 9.13 : Patrons pour la coordination verbale

ii. Structure argumentaire du verbe

Type 1 : il parle de l'école de cuisine

Patron 3

```
(setf (get 'lidia-patterns 'sav-t1-1)
      '(((is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1))
        ?punc )))
```

et

Patron 4

```
(setf (get 'lidia-patterns 'sav-t1-2)
      '(((is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((is ?y node-prop-equal-p 'FS 'CIRC)
         (?* ?p1))
        ?punc )))
```

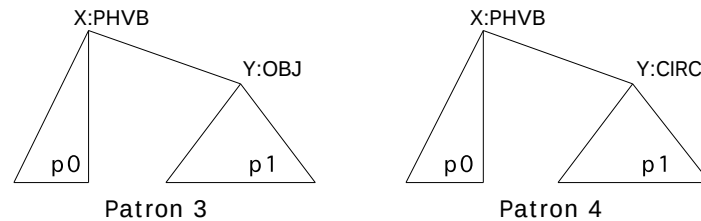


Figure 9.14 : Patrons pour la structure argumentaire du verbe – type 1

Type 2 : il parle depuis l'école de cuisine

Patron 5

```
(setf (get 'lidia-patterns 'sav-t2-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'CIRC)
          (?* ?p1))
        ((?is ?z node-prop-equal-p 'FS 'OBJ)
          (?* ?p2))
        ?punc ))
```

et

Patron 6

```
(setf (get 'lidia-patterns 'sav-t2-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'CIRC)
          (?* ?p1)
          ((?is ?z node-prop-equal-p 'FS 'COMP)
            (?* ?p2))))
        ?punc ))
```

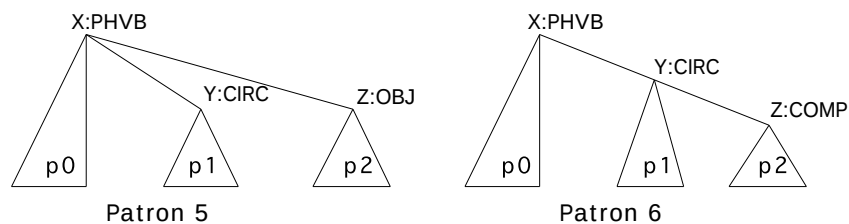


Figure 9.15 : Patrons pour la structure argumentaire du verbe – type 2

iii. Coordination nominale

Patron 7

```
(setf (get 'lidia-patterns 'cn-1)
      '(?x
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COMP)
          (?* ?p1)
          ((?is ?v node-prop-equal-p 'FS 'COMP)
            (?* ?p2)
            ((?is ?t node-prop-equal-p 'FS 'COORD)
```

(?* ?p3))))))

et

Patron 8

```
(setf (get 'lidia-patterns 'cn-2)
      '(?x
        (* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COMP)
         (* ?p1)
         ((?is ?v node-prop-equal-p 'FS 'COMP)
          (* ?p2))
         ((?is ?t node-prop-equal-p 'FS 'COORD)
          (* ?p3)))))
```

et si possible

Patron 9

```
(setf (get 'lidia-patterns 'cn-3)
      '(?x
        (* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COMP)
         (* ?p1)
         ((?is ?v node-prop-equal-p 'FS 'COMP)
          (* ?p2)))
        ((?is ?t node-prop-equal-p 'FS 'COORD)
         (* ?p3))))
```

;;On étudie l'évolution de la structure du réseau et des investissements

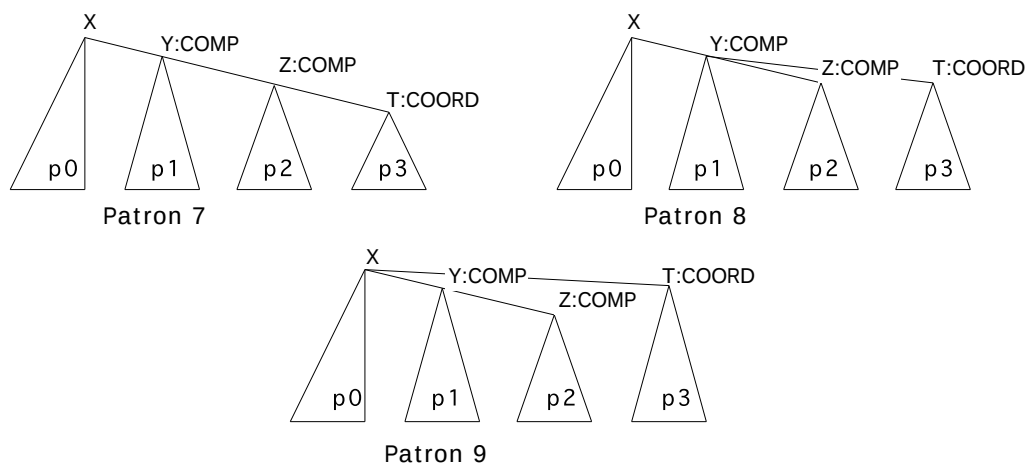


Figure 9.16 : Patrons pour la coordination nominale

iv. Coordination adjectivale

Patron 10

```
(setf (get 'lidia-patterns 'ca-1)
      '(?x
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COORD)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'CS 'GADJ)
          (?* ?p2)
          )))))
```

et

Patron 11

```
(setf (get 'lidia-patterns 'ca-2)
      '(?x
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COORD)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'CS 'GADJ)
         (?* ?p2))))
```

;; les cahiers et les classeurs noirs

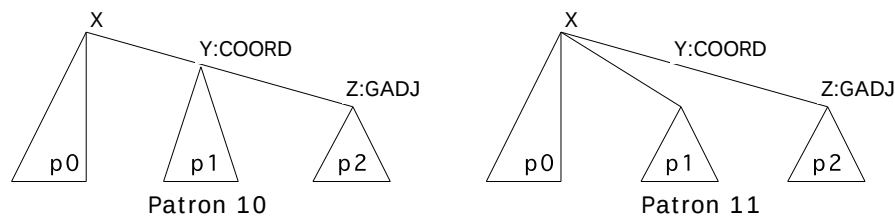


Figure 9.17 : Patrons pour la coordination adjectivale

v. Subordination

Type 1 : Marie voit l'homme dans le parc

Patron 12

```
(setf (get 'lidia-patterns 'sub-t1-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        ((?is ?y node-prop-equal-p 'FS 'SUJ) (?* ?p0))
        ((?is ?z node-prop-equal-p 'CS 'NV) (?* ?p1))
        ((?is ?u node-prop-equal-p 'FS 'OBJ)
         (?* ?p2)
         ((?is ?v node-prop-equal-p 'FS 'COMP) (?* ?p3)))
        ?punc ))
```

et

Patron 13

```
(setf (get 'lidia-patterns 'sub-t1-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        ((?is ?y node-prop-equal-p 'FS 'SUJ) (?* ?p0))
        ((?is ?z node-prop-equal-p 'CS 'NV) (?* ?p1))
        ((?is ?u node-prop-equal-p 'FS 'OBJ)
         (?* ?p2))
        ((?is ?v node-prop-equal-p 'FS 'CIRC) (?* ?p3))
        ?punc ))
```

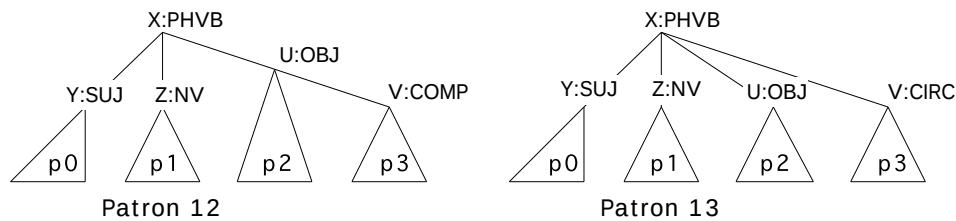


Figure 9.18 : Patrons pour la subordination – type 1

Type 2 : Elle épouse un professeur de droit anglais

Patron 14

```
(setf (get 'lidia-patterns 'sub-t2-1)
      '(?x
        (?* ?p0)
        ((?or (?is ?y node-prop-equal-p 'FS 'COMP)
              (?is ?y node-prop-equal-p 'FS 'OBJ))
         (?* ?p1))
        ((?or (?is ?z node-prop-equal-p 'FS 'ATG)
              (?is ?z node-prop-equal-p 'FS 'ATOBJ))
         (?* ?p2))))
```

et

Patron 15

```
(setf (get 'lidia-patterns 'sub-t2-2)
      '(?x
        (?* ?p0)
        ((?or (?is ?y node-prop-equal-p 'FS 'COMP)
              (?is ?y node-prop-equal-p 'FS 'OBJ))
         (?* ?p1))
        ((?or (?is ?z node-prop-equal-p 'FS 'ATG)
              (?is ?z node-prop-equal-p 'FS 'ATOBJ))
         (?* ?p2))))
```

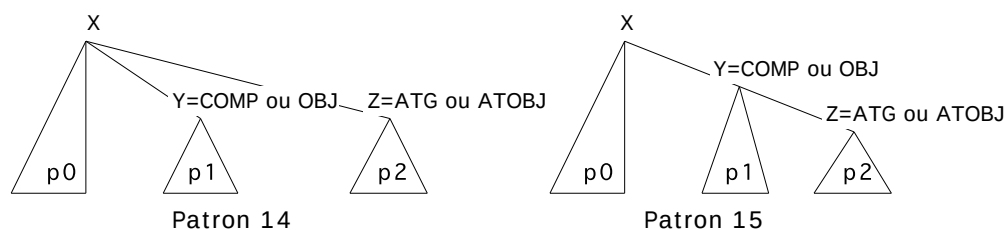


Figure 9.19 : Patrons pour la subordination – type 2

9.3.2. Méthodes

Nous pouvons maintenant décrire les méthodes utilisées pour produire le dialogue.

Pour chaque classe d'ambiguïtés reconnue avec des patrons déterminés, nous donnons la formule qui permet d'obtenir le texte du dialogue en utilisant les noms de variables que nous avons utilisés dans la description des patrons. Pour les autres cas, nous donnons la méthode en extension.

a. Polysémie

On parcourt le mot des feuilles de l'analyse finalement retenue. Pour chaque occurrence, on vérifie le nombre de valeurs que contient la variable sens. Si cette variable contient plusieurs valeurs, on présente la définition de chacun des numéros de sens retenus par l'analyseur.

b. Ambiguïté de classe syntaxique

i. Coordination verbale

La coordination verbale concerne les patrons 1 et 2.

Patron 1 `Sujet(?p0) NoyauVerbal(?p2) Texte(?p1)`

Patron 2 `Sujet(?p0) NoyauVerbal(?p0) Moins_Coord(?p2)`

ii. Classe simple

L'ambiguïté de classe simple est détectée par examen du mot des feuilles des solutions et l'absence d'une phrase verbale coordonnée dominant une occurrence ambiguë. Pour produire le dialogue, on projette les classes des occurrences pour lesquelles celles-ci sont différentes.

c. Ambiguïté de géométrie

i. Structure argumentaire du verbe

Patron 3 `Texte(?p0) Substitue (Régisseur(?p1) #Objet) Moins_Reg(?p1)`

Patron 4 `Texte(?p0) Substitue (Régisseur(?p1) #Circ) Moins_Reg(?p1)`

Patron 5 Texte(?p0) Texte(?p1) Substitue (Régisseur(?p2) #Objet) Moins_Reg(?p2)

Patron 6 Texte(?p0) Régisseur(?p1) Parenthèse(Moins_Reg(?p1) Texte(?p2))

ii. Coordination nominale

Patron 7 Distribue((?p0, ?p1, ?p2, Moins_coord(?p3)), Coordonnant(?p3),
(1, 2, 3), (1, 2, 4))

Patron 8 Distribue((?p0, ?p1, ?p2, Moins_coord(?p3)), Coordonnant(?p3),
(1, 2, 3), (1, 4))

Patron 9 Distribue_et_() ((?p0, ?p1, ?p2, Moins_coord(?p3)), Coordonnant(?p3),
(1, 2, 3), (4))

iii. Coordination adjectivale

Patron 10 Texte(?p0) Coordonnant(?p1) Parenthèse(Moins_Coord(?p1), Texte(?p2))

Patron 11 Parenthèse(Texte(?p0), Accordé(?p2))
Coordonnant(?p1)
Parenthèse(Moins_Coord(?p1), Texte(?p2))

iv. Subordination

Patron 12 Texte(?p0) Texte(?p1) Parenthèse(Texte(?p2), Texte(?p3))

Patron 13 Texte(?p3) , Texte(?p0) Texte(?p1) Texte(?p2)

Patron 14 Texte(?p0) Parenthèse(Texte(?p1), Texte(?p2))

Patron 15 Texte(?p0) Texte(?p2) Texte(?p1)

d. Ambiguïté de décoration

Pour les ambiguïtés de décoration, nous n'utilisons pas pour l'instant le mécanisme de filtrage, mais nous testons les projections des solutions produites par l'analyseur. Sur ces projections, nous pouvons facilement effectuer des opérations de tri et reconnaître les différences.

i. Étiquetage logico-sémantique et étiquetage syntaxique

On remplace la préposition qui introduit le(s) groupe(s) ambigu(s) par une préposition non ambiguë.

ii. Ordre des arguments d'un verbe transitif direct

On repère les groupes ambigus et on produit une phrase normalisée à l'aide de leur gouverneur muni des bons articles.

9.3.3. Résultats

a. Polysémie

Les sens qui figurent dans les deux dialogues suivants sont trouvés dans la base Parax.

ambiguïté	
Il y a plusieurs sens acceptables pour le mot : étudier	
Choisissez le bon.	
☒	Chercher à acquérir la connaissance.
☐	Chercher à comprendre par un examen.
OK	

Figure 9.20 : Dialogue pour l'ambiguïté de polysémie – exemple 1

ambiguïté	
Il y a plusieurs sens acceptables pour le mot : capitaine	
Choisissez le bon.	
☒	Officier qui commande une compagnie d'infanterie, un escadron de cavalerie, une batterie d'artillerie.
☐	Officier qui commande un navire de commerce
☐	Chef d'une équipe sportive
OK	

Figure 9.21 : Dialogue pour l'ambiguïté de polysémie – exemple 2

b. Ambiguïté de classe syntaxique

i. Coordination verbale

ambiguïté

Il y a plusieurs interprétations pour la phrase :
Il atteint la grange et la ferme.

Choisissez la bonne.

☒ **Il atteint la ferme.**

☐ **Il ferme la grange.**

OK

Figure 9.22 : Dialogue pour l’ambiguïté de coordination verbale

La première paraphrase est obtenue avec le patron 2 et la seconde avec le patron 1. On a reconstitué les ellipses potentielles dans le groupe coordonné “et la ferme”. Dans la première interprétation, on avait ellision du sujet (“il”) et du verbe (“ferme”). Dans la seconde interprétation, on avait ellision du sujet (“il”) et reprise de l’objet sous forme pronominale (“la” pour “la grange”).

ii. Classe simple

ambiguïté

Il y a plusieurs interprétations pour la phrase :
Un savant compromis a été arrêté hier.

Choisissez la bonne.

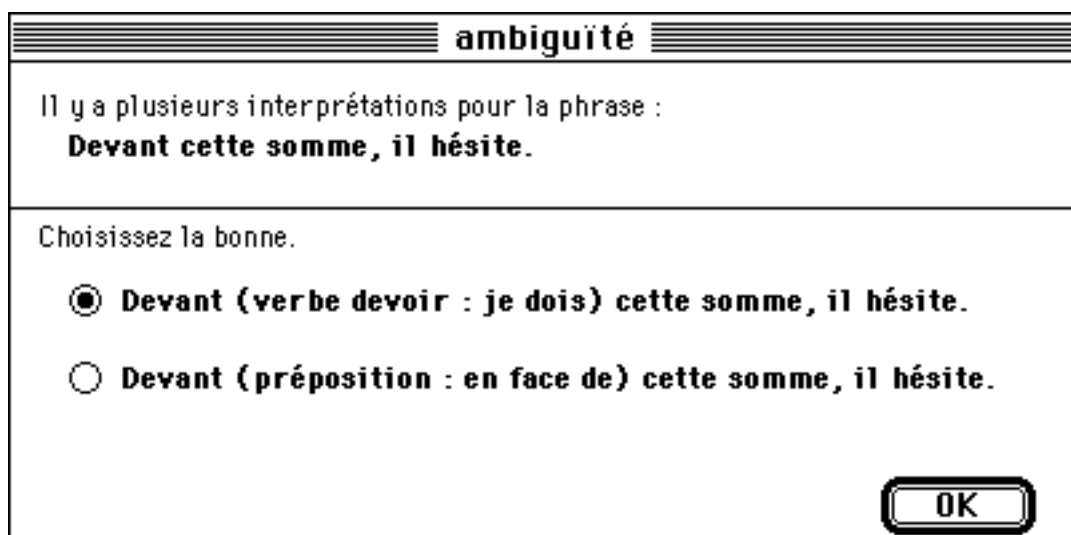
☒ **Un savant (adjectif) compromis (nom) a été arrêté hier.**

☐ **Un savant (nom) compromis (adjectif) a été arrêté hier.**

OK

Figure 9.23 : Dialogue pour l’ambiguïté de classe syntaxique – exemple 1

Pour la phrase “Un savant compromis a été arrêté hier.” on projette la classe syntaxique de savant et de compromis qui change selon la lecture de la phrase.



ambiguïté

Il y a plusieurs interprétations pour la phrase :
Devant cette somme, il hésite.

Choisissez la bonne.

☒ **Devant (verbe devoir : je dois) cette somme, il hésite.**

☐ **Devant (préposition : en face de) cette somme, il hésite.**

OK

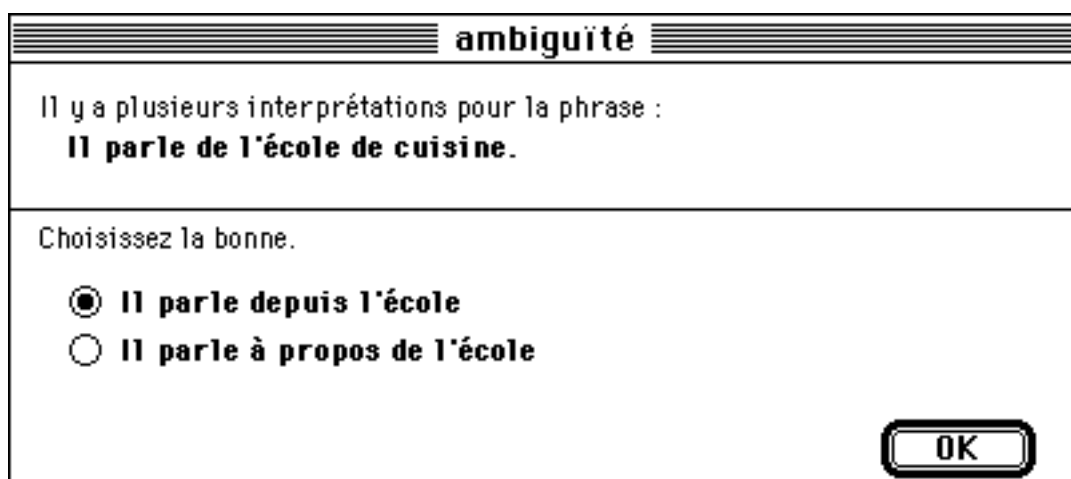
Figure 9.24 : Dialogue pour l'ambiguïté de classe syntaxique – exemple 2

Pour la préposition, on propose un synonyme, et pour le verbe on propose sa conjugaison à la première personne du présent de l'indicatif (pour les arbres, cf figure 9.2).

c. **Ambiguïté de géométrie**

i. **Structure argumentaire du verbe**

Type 1



ambiguïté

Il y a plusieurs interprétations pour la phrase :
Il parle de l'école de cuisine.

Choisissez la bonne.

☒ **Il parle depuis l'école**

☐ **Il parle à propos de l'école**

OK

Figure 9.25 : Dialogue pour l'ambiguïté de structure argumentaire – type 1

La première paraphrase est produite avec la méthode associée au patron 4, la seconde, avec la méthode associée au patron 3. On remplace la préposition ambiguë “de” par “depuis” lorsqu'elle introduit un circonstant et par “à propos de” lorsqu'elle introduit un objet (pour les arbres, cf figure 9.3 ; pour les patrons, figure 9.14).

Si l'utilisateur choisit la première interprétation, il se verra proposer le dialogue suivant (figure 9.26).

Type 2

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Il parle de l'école de cuisine.	
Choisissez la bonne.	
☒	Il parle depuis (l'école de cuisine.)
☐	Il parle depuis l'école à propos de cuisine.
OK	

Figure 9.26 : Dialogue pour l'ambiguïté de structure argumentaire– type 2

La première paraphrase est produite avec la méthode associée au patron 6, la seconde avec la méthode associée au patron 5.

ii. Coordination nominale

ambiguïté	
Il y a plusieurs interprétations pour la phrase : L'évolution de la structure du réseau et des investissements est étudiée.	
Choisissez la bonne.	
☒	L'évolution de la structure du réseau et l'évolution des investissements
☐	L'évolution de la structure du réseau et l'évolution de la structure des investissements
OK	

Figure 9.27 : Dialogue pour l'ambiguïté de coordination nominale

La première paraphrase est obtenue avec la méthode associée au patron 8, la seconde avec la méthode associée au patron 7. Dans les deux cas, on produit une représentation de la phrase dans laquelle aucun terme n'a été éliminé. "des investissements" peut se rapporter soit à "l'évolution" soit à "l'évolution de la structure" (pour les patrons, cf. figure 9.16).

iii. Coordination adjectivale

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Il prend les cahiers et les classeurs noirs.	
Choisissez la bonne.	
☒	Il prend les cahiers et (les classeurs noirs).
☐	Il prend (les cahiers noirs) et (les classeurs noirs).
OK	

Figure 9.28 : Dialogue pour l'ambiguïté de coordination adjectivale

Ici, il s'agit de décider comment doit se distribuer le groupe adjectival "noirs". La première paraphrase est produite avec la méthode associée au patron 11, la seconde, avec la méthode associée au patron 10. Dans le premier cas, la distribution ne produit pas de perturbation de l'ordre des occurrences, alors, on parenthèse. Dans le second cas, la distribution de l'adjectif est effectuée ; les groupes produits sont parenthésés (pour les arbres, cf figure 9.5 ; pour les patrons, figure 9.17).

iv. Subordination

Type 1

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Le capitaine a rapporté un vase de Chine.	
Choisissez la bonne.	
☒	de Chine, le capitaine a rapporté un vase.
☐	Le capitaine a rapporté (un vase de Chine).
OK	

Figure 9.29 : Dialogue pour l'ambiguïté de subordination type 1

La première paraphrase est obtenue avec la méthode associée au patron 13, la seconde, avec la méthode associée au patron 12. Dans le premier cas on place le

circonstant, suivi d'une virgule, avant la phrase. Dans le second cas, on parenthèse l'objet du verbe.

Type 1 généralisé

ambiguïté

Il y a plusieurs interprétations pour la phrase :
Marie voit l'homme dans le parc avec un télescope.

Choisissez la bonne.

☒ **Marie voit (l'homme dans le parc).**

☐ **dans le parc, Marie voit l'homme.**

OK

Figure 9.30 : Dialogue pour l'ambiguïté de subordination type 1 généralisé

Ce dialogue est construit en suivant la même technique que pour le dialogue précédent.

Si l'utilisateur choisit la première interprétation, il se verra proposer, le dialogue suivant (figure 9.31). S'il choisit la seconde, il se verra proposer le dialogue de la figure 9.32.

ambiguïté

Il y a plusieurs interprétations pour la phrase :
Marie voit l'homme dans le parc avec un télescope.

Choisissez la bonne.

☒ **dans le parc, Marie voit (l'homme avec un télescope).**

☐ **Marie voit l'homme (dans le parc avec un télescope).**

☐ **avec un télescope, Marie voit (l'homme dans le parc).**

OK

Figure 9.31 : Dialogue pour l'ambiguïté de subordination – seconde étape 1

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Marie voit l'homme dans le parc avec un télescope.	
Choisissez la bonne.	
☒	(dans le parc avec un télescope), Marie voit l'homme.
☐	 dans le parc, avec un télescope, Marie voit l'homme.
OK	

Figure 9.32 : Dialogue pour l'ambiguïté de subordination – seconde étape 2

Pour les deux dialogues ci-dessus, on utilise la même technique que pour le dialogue 9.30 mais les patrons s'appliquent cette fois sur toute la phrase. En effet, pour produire le dialogue 9.30, les patrons se sont appliqués sur le segment de phrase "Marie voit l'homme dans la parc."

Type 2

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Elle épouse un professeur de droit anglais.	
Choisissez la bonne.	
☒	Elle épouse (un professeur anglais) de droit.
☐	Elle épouse un professeur (de droit anglais).
OK	

Figure 9.33 : Dialogue pour l'ambiguïté de subordination type 2 – exemple 1

La première paraphrase du dialogue précédent est obtenu avec la méthode associée au patron 14, la seconde, avec la méthode associée au patron 15. Ici on accole l'adjectif anglais au substantif qu'il modifie. On parenthèse le couple substantif, adjectif pour bien montrer ce que l'on a fait (pour les arbres cf. figure 9.6 ; pour les patrons, figure 9.19).

ambiguïté	
Il y a plusieurs interprétations pour la phrase : L'école de cuisine lyonnaise ouvre ses portes.	
Choisissez la bonne.	
☒	(L'école lyonnaise) de cuisine ouvre ses portes.
☐	L'école (de cuisine lyonnaise) ouvre ses portes.
OK	

Figure 9.34 : Dialogue pour l'ambiguïté de subordination type 2 – exemple 2

Ce dialogue est obtenu par la même technique que le dialogue précédent, cette fois le problème se situe avant le verbe de la phrase.

d. Ambiguïté de décoration

i. Étiquetage logico-sémantique

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Pierre fait porter des chocolats à Lucie.	
Choisissez la bonne.	
☒	Pierre fait porter des chocolats pour Lucie.
☐	Pierre fait porter des chocolats par Lucie.
OK	

Figure 9.35 : Dialogue pour l'ambiguïté d'étiquetage logico-sémantique

Pour produire ce dialogue, on remplace la préposition ambiguë “à” par “pour” si “à Lucie” est le second argument du verbe “porter” et par “par” si “à Lucie” est l’argument zéro du verbe “porter” (pour l’arbre cf. figure 9.7).

ii. Étiquetage syntaxique

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Je vous parle de la tour Eiffel.	
Choisissez la bonne.	
☒	Je vous parle depuis la tour Eiffel.
☐	Je vous parle à propos de la tour Eiffel.
OK	

Figure 9.36 : Dialogue pour l’ambiguïté d’étiquetage syntaxique – exemple 1

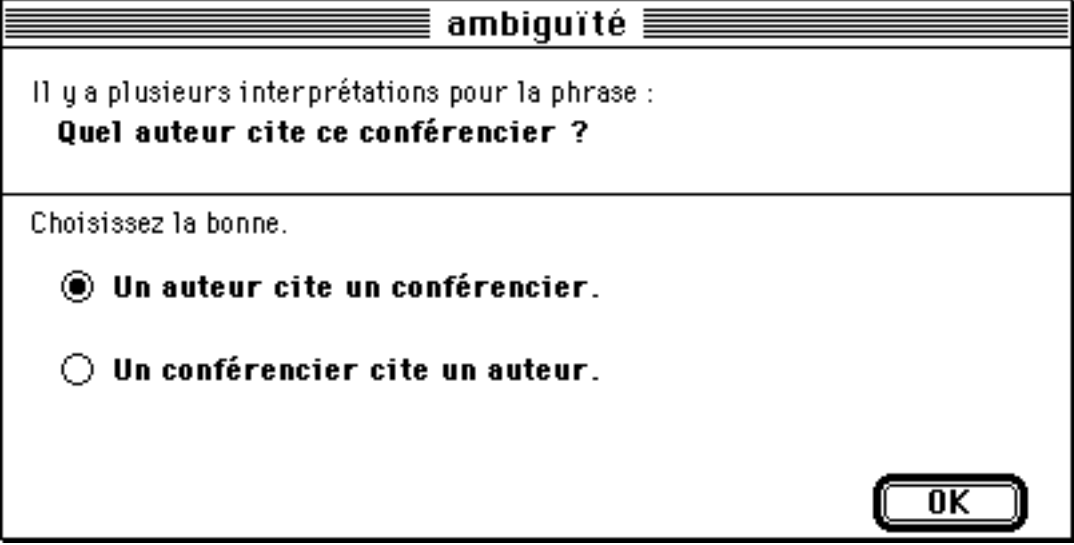
Pour produire ce dialogue, on remplace la préposition ambiguë “de” par “depuis” si elle introduit un circonstant du verbe “parler” et par “à propos de” si elle introduit le premier argument du verbe (pour les arbres cf. figure 9.8).

ambiguïté	
Il y a plusieurs interprétations pour la phrase : Je vote pour ma mère.	
Choisissez la bonne.	
☒	je vote en faveur de ma mère.
☐	je vote en prenant la place de ma mère.
OK	

Figure 9.37 : Dialogue pour l’ambiguïté d’étiquetage syntaxique – exemple 2

Pour obtenir ce dialogue, on procède comme précédemment. Ici, on remplace “pour”.

iii. Ordre des arguments d'un verbe transitif direct



ambiguïté

Il y a plusieurs interprétations pour la phrase :
Quel auteur cite ce conférencier ?

Choisissez la bonne.

☒ **Un auteur cite un conférencier.**

☐ **Un conférencier cite un auteur.**

OK

Figure 9.38 : Dialogue pour l'ambiguïté d'ordre des arguments d'un verbe

Ici, on organise la phrase selon un schéma *article sujet verbe article objet*. Les sujet verbe et objet sont pris dans la phrase originale. Les articles sont choisis en fonction du genre et du nombre des sujet et objets.

Remarques sur les techniques d'implémentation

L'implémentation que nous avons réalisée permet de jeter les bases d'une architecture logicielle pour la TAFD, avec trois mots clés : intégration, distribution et extensibilité.

La mise en œuvre de quatre serveurs qui communiquent par messages et fichiers permet d'intégrer des outils qui travaillent dans différents environnements matériels et/ou logiciels. La gestion des messages (acheminement et contrôle des files d'attente) est prise en charge par le système d'exploitation et permet de se concentrer sur la réalisation des services.

Le serveur de communication nous a permis d'utiliser un outil très rudimentaire de communication avec le serveur de traduction, sans que la performance du système global en soit affectée. Pour la "TAFD à la maison", un serveur de communication adéquat piloterait un modem (autonome ou celui du minitel) pour travailler non plus avec un serveur de traduction, mais avec un serveur LIDIA distribué. Avec une telle architecture, la machine de Monsieur Toutlemonde sera suffisante pour servir de station de rédaction.

Chaque fonctionnalité du système peut être facilement modifiée puisqu'elle est parfaitement identifiée et localisée. On peut changer définitivement une fonctionnalité en remplaçant son code, ou provisoirement en modifiant, dans la fonction d'installation du message concerné, le nom de la procédure de gestion. L'ajout d'une nouvelle fonctionnalité passe par la création et l'installation d'un nouveau message et de procédures de réponse associées. Il suffit d'insérer dans le code existant l'appel de la nouvelle fonctionnalité et la gestion du résultat qu'elle produit.

Conclusion

Apports de ce travail et discussion

Des travaux antérieurs ou parallèles sur les systèmes de traduction automatique ou automatisée qui intègrent un module de désambiguïsation interactive, nous retenons et reprenons à notre compte :

- les principes et la méthodologie de LMT,
- l’interface conviviale et productive de KBMT-89 et JETS.

Notre concept de Traduction Automatisée Fondée sur le Dialogue peut être vu comme une suite logique à ces projets. Mais il introduit des contraintes spécifiques qui nous conduisent à beaucoup plus insister sur la nécessité:

- de partager le linguiciel,
- de travailler dans l’optique de bases lexicales de très grande taille,
- de résoudre des problèmes cruciaux au niveau de l’interface.

Dans le cadre de ce paradigme, nous avons proposé quelques idées originales :

- une nouvelle architecture linguistique par le transfert multiniveau à acceptions, propriétés et relations interlingues,
- la notion de langage guidé,
- l’accessibilité des connaissances linguistiques du système,
- des critères de choix de l’approche par le dialogue, et des situations adaptées.

L’implémentation de la maquette LIDIA-1, première expérience concrète en vue de la “TAFD pour tous”, s’est développée “en largeur” sur certains points, et “en profondeur” sur d’autres. Ainsi, nous avons prototypé toutes les fonctionnalités prévues, mais n’avons implémenté complètement que le processus de désambiguïsation, traité en profondeur. Cela était inévitable au niveau d’une maquette que nous avons voulu extensible, grâce à l’architecture utilisée.

Il était très important que tous les aspects aient été abordés, car les expériences passées montrent qu’une conception globale est nécessaire pour qu’un système de TAO réussisse. En effet, et nous l’avons bien vu en travaillant sur LIDIA-1, des objectifs de nature ergonomique peuvent contraindre certains choix informatiques et linguistiques, et réciproquement pour les objectifs informatiques et linguistiques.

À notre sens, si les travaux précédents en TAFD n'ont pas (encore) mené à des systèmes utilisés en pratique, cela est dû dans une large mesure à une situation mal choisie et à un dialogue trop contraignant. Pour le projet LIDIA, nous avons étudié ces aspects dès la phase de conception afin d'éviter ces écueils. Même si un maquettage ne permettait pas d'intégrer toutes les propositions initiales, nous n'avons fait aucun choix qui en exclurait définitivement certains. Les rappeler ici permet de mesurer le chemin qui sépare notre maquettage du prototype auquel nous désirons maintenant nous livrer..

Dans la situation traductionnelle que nous envisageons, l'auteur doit être enclin et encouragé à participer au dialogue de standardisation et de clarification. De plus, afin de réduire le nombre de questions à poser, nous proposons deux éléments nouveaux : la notion d'accessibilité des connaissances du système (§ 2.2.3.) et la notion de langage guidé (§ 2.2.2.). Ces deux éléments ne sont pas encore intégrés dans la maquette et restent des thèmes de recherche actuels.

Si l'usage de langages guidés doit permettre de diminuer le degré d'ambiguïté des textes à traduire, l'accès aux connaissances du système est aussi un élément important. Nous pensons que l'auteur doit être en mesure d'apprendre à reconnaître les structures linguistiques (patrons) qui sont ambiguës pour le module d'analyse automatique. Dans cette optique, nous avons proposé différents niveaux de dialogue (§ 6.3.3.). Au fur et à mesure que l'auteur se familiarise avec le système, il pourra choisir d'éviter l'usage des patrons à problème.

On souhaite aussi que l'auteur consulte la structure linguistique finalement construite par l'analyseur. Cette consultation doit lui permettre de comprendre les éléments linguistiques simples manipulés par l'analyseur. Nous pensons particulièrement aux notions de classe syntaxique, groupe syntagmatique et fonction syntaxiques, qui sont faciles à comprendre pour un auteur d'un niveau d'expertise grammaticale équivalent à celui d'un bachelier. L'auteur pourrait alors insérer des marques de désambiguïsation au fur et à mesure qu'il rédige le texte à traduire.

Pour le processus de désambiguïsation, se pose le problème du nombre de patrons à définir et de la complexité du processus de reconnaissance des dits patrons sur des structures linguistiques complexes. Pour la maquette, nous avons défini un langage de description de patrons (§ 9.3.1.) qui a une puissance de description suffisante pour que l'on puisse factoriser dans le même patron plusieurs "formes" d'une même ambiguïté. Bien que la maquette ne permette pas encore de le démontrer, les linguistes nous disent que nous avons obtenu une très bonne couverture des ambiguïtés envisagées.

On a bien montré aussi que le mécanisme de reconnaissance de patrons ne pouvait pas être utilisé pour toutes les ambiguïtés, sous peine cette fois-ci de multiplier les patrons. Lorsque les diverses analyses produites pour une phrase possèdent une même géométrie, nous procédons d'ailleurs tout autrement, en comparant directement l'étiquetage des nœuds, le coût de la recherche étant alors intrinsèquement limité.

Comme le système est asynchrone, le temps passé à rechercher les patrons et à produire les dialogues n'empêche pas l'auteur de travailler. Certes, le processus de reconnaissance de patron doit avoir une efficacité maximale, ce qui n'a pas été l'objectif premier de l'implémentation que nous utilisons dans la maquette. Comme nous ne prétendons pas avoir défini les meilleurs patrons, nous proposons un environnement de description de processus de désambiguïsation dans le paragraphe suivant concernant les perspectives scientifiques.

Nous avons proposé que le linguiciel soit partagé (§ 2.1.1.) afin que la station de rédaction soit un ordinateur personnel de milieu de gamme avec une mémoire vive et une capacité de stockage limitées. Une telle architecture client-serveur permet de s'intégrer à toute application offrant une API (Interface de Programmation d'Application). Pour une utilisation de la TAFD à la maison, on peut imaginer que l'utilisateur se connecte au serveur de traduction via un minitel.

Ainsi, nous considérons comme une nécessité intrinsèque d'organiser un système de TAFD de façon distribuée, bien plus que comme une contrainte contingente venant du fait que notre maquette devait intégrer des composants tournant sur des machines différentes.

Si les ordinateurs personnels de bas de gamme du futur devenaient assez puissant pour effectuer des traitements longs, lourds en tâche de fond, voir même en temps réel, on pourra bien sûr faire tourner tous les serveurs sur la station de rédaction. Mais il n'est pas sûr que ce soit toujours souhaitable, car l'expérience prouve que les ressources disponibles sont très vite saturées par les systèmes d'exploitation et les applications toujours plus gourmandes. D'autre part, les bases de connaissances lexicales et grammaticales d'un système de TAO évoluent constamment, et il faudra très fréquemment que chaque utilisateur les remette à jour.

Perspectives scientifiques

À côté des projets que nous évoquons dans le paragraphe suivant, nous imaginons trois axes de recherche possibles dans la continuation directe de ce travail.

Tout d'abord, le processus de désambiguïsation que nous avons implémenté dans la maquette est figé et ne comporte qu'un niveau de dialogue. Si l'on veut pouvoir implémenter une autre stratégie, de nouveaux opérateurs et de nouvelles méthodes de désambiguïsation, il faut disposer d'un environnement de description de processus de désambiguïsation. Un tel environnement pourrait être utilisé directement par un linguiste-informaticien, et comporterait trois modules :

- un module de description de patrons,
- un module de description des méthodes de production de dialogues à partir des patrons,
- un module de description de stratégies de désambiguïsation.

En second lieu, il faudrait passer à un véritable prototype permettant une expérimentation en interne avec des utilisateurs réels. L'architecture informatique de LIDIA-1 pourrait être reprise telle quelle dans un système plus ambitieux. Les outils de standardisation sont restés à un stade embryonnaire dans la maquette et devront être repensés. Il faudra bien sûr une plus grande couverture de la langue tant au niveau du lexique que de la grammaire. Il faudrait enfin développer de nouveaux outils informatiques pour l'analyse, plus adaptés à la programmation ambiguë [Lafourcade 1993b] que ceux de Ariane-G5.

En troisième lieu, il faut continuer à étudier le typage textuel, qui est à notre avis un bon moyen pour réduire le dialogue de désambiguïsation. Pour cela, on pourrait réaliser une nouvelle maquette en intégrant les principes de LIDIA-1 dans un éditeur structural tel que Grif [Phan & Boitet 1992]. Cette maquette permettrait de pousser plus loin l'idée du typage textuel en ajoutant un niveau de description au document en terme de types de documents et genres de textes.

Autres projets

À partir du travail que nous avons effectué, qui constitue une sorte d'étude de faisabilité, on pourrait aussi passer à la construction d'un simulateur qui permette de faire des expériences autour du concept de TAFD sans devoir implémenter un logiciel de TA complet, ce qui est fort coûteux. Le cœur d'un tel simulateur serait un magicien d'Oz qui simulerait le comportement d'un vrai système de TAFD. On pourrait alors expérimenter d'autres modalités d'interaction entre le système et l'utilisateur. Pour l'instant, on pense bien sûr à la parole, mais aussi à l'utilisation d'icônes, pour les ambiguïtés de référence pronominale par exemple.

Avec un tel simulateur, nous pourrions collecter sous forme de bande vidéo sonorisée des données sur le comportement de l'utilisateur en fonction du comportement du système.

La construction d'un tel simulateur nécessite l'étude et l'intégration de nouveaux savoir-faire auxquels nous avons commencé à nous intéresser. Voici quelques références sur les aspects à notre avis les plus importants d'un tel simulateur :

- interfaces multimodales : [Amalberti, *et al.* 1993 ; Hauptmann & Rudnicky 1988 ; IJMMS 1988 ; Taylor, *et al.* 1989],
- interaction homme machine avec la parole : [Ainsworth & Pratt 1992 ; Dahlbäck 1991 ; Frankish, *et al.* 1992 ; Jönsson 1993 ; Kennedy, *et al.* 1988 ; Pierrel 1987 ; Streeter 1991],
- désambiguïsation par synthèse vocale : [DeHaemer & Wallace 1992 ; Klatt 1975 ; Lehisté 1973 ; Lehisté, *et al.* 1976 ; O'Shaughnessy 1989 ; Streeter 1978], et
- simulation de systèmes avec un magicien d'Oz : [Dahlbäck 1993 ; Dahlbäck & Jönsson 1986 ; Dahlbäck, *et al.* 1993 ; Fraser & Gilbert 1991 ; Whittaker & Stenton 1989].

Bibliographie

- 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.
- Amalberti, R., Carbonell, N. & Falzon, P., (1993). *User representation of computer systems in human-computer speech interaction*. in International Journal of Man-Machine Studies. vol. **38**(4) : pp. 547-566.
- Apple Computer, (1989). *Hypercard Stack Design Guidelines*. Addison-Wesley Publishing Company, Inc. 230 p.
- Apple Computer, (1987). *Human Interface Guidelines: The Apple Desktop Interface*. Addison-Wesley Publishing Company, Inc. 144 p.
- Apple Computer, (1992). *Macintosh Human Interface Guidelines*. Addison-Wesley Publishing Company, Inc. 384 p.
- Balkan, L., (1992). *Translation Tools*. in Meta. vol. **37**(3) : pp. 408-420.
- Bar-Hillel, Y., (1960). *The present Status of Automatic Translation of Languages*. in Advances in Computers, New York, Academic Press. vol. **1** : pp. 91-163.
- Bateman, R., (1985). *Introduction to Interactive Translation*. in Tools for the Trade. Aslib. London. pp. 193-197.
- Béjoint, H., (1990). *Dictionaries and the Acquisition of Polysemy*. Proc. The Łódź Session of the 1990 Maastricht-Łódź Duo Colloquium on "Translation and Meaning". Łódź, Poland. 20-22 septembre 1990 : pp. 307-312.
- Ben-Ari, D., Berry, D. m. & Rimón, M., (1988). *Translational Ambiguity Rephrased*. Proc. Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language. Carnegie-Mellon University, Pittsburgh. 12-14 juin 1988.
- Blanc, É. & Boitet, C. (eds.), (1990). *DBMT-90: Post-Coling Seminar on Dialogue-Based Machine Translation (Machine Translation of/with Dialogues)*. Grenoble, France. 328 p.
- Blanchon, H., (1990a). *LIDIA-1 : Un prototype de TAO personnelle pour rédacteur unilingue*. Proc. Avignon-90, conférence spécialisée : Le Traitement Automatique des Langues Naturelles et ses Applications. Avignon, France. 28 mai-1 juin 1990 : pp. 51-60.
- Blanchon, H., (1990b). *TAO personnelle et promotion des langues nationales : projet LIDIA du GETA. Jeu de transparents de présentation du projet*. Proc. Les industries de la langue : perspectives des années 1990. Montréal, Canada. 22-24 novembre 1990.
- Blanchon, H., (1991). *Problèmes de désambiguïsation interactive en TAO Personnelle*. Proc. L'environnement traductionnel ; La station de travail du traducteur de l'an 2001. Mons, Belgique. 25-27 avril 1991, vol. **1/1** : pp. 31-48.
- Blanchon, H., (1992a). *Désambiguïsation interactive en TAO personnelle : poser les bonnes questions*. Proc. Avignon-92, conférence spécialisée : Le Traitement Automatique Des Langues Naturelles et ses Applications. Avignon, France. 3-5 juin 1992, vol. **3/4** : pp. 69-80.

Blanchon, H., (1992b). *A Solution to the Problem of Interactive Disambiguation*. Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. 4/4 : pp. 1233-1238.

Blanchon, H., (1994). *Interagir pour traduire : la TAO personnelle pour rédacteur monolingue*. in La Tribune des Industries de la Langues. vol. 14.

Blanchon, H., Guilbaud, J. P. & Nédobekine, N., (1992). *LIDIA : the disambiguation process — le processus de désambiguïsation, Pile HyperCard présentée à l'exposition Coling-92, Nantes, 23-28 juillet 1992*. Geta, I. juillet 1992. 160 p.

Boitet, C., Réd., (1982). “DSE-1”— *Le point sur ARIANE-78 début 1982*. Rap. GETA, Grenoble. Contrat ADI/CAP-Sogeti/Champollion (3 vol.).

Boitet, C., (1989a). *Motivations, aims and architecture of the Lidia project*. Proc. MT SUMMIT II. Munich. 16-18 août 1989, vol. 1/1 : pp. 53-57.

Boitet, C., (1989b). *Speech Synthesis and Dialogue Based Machine Translation*. Proc. ATR Symp. on Basic Research for Telephone Interpretation. Kyoto. décembre 1989.

Boitet, C., (1990a). *La TAO à Grenoble en 1990*. in École d'été de LANNION sur le Traitement Automatique des Langues Naturelles. CNET, juillet 1990, 65 p.

Boitet, C., (1990b). *Towards Personal MT : on some aspects of the LIDIA project*. Proc. Coling-90. Helsinki. 20-25 Août 1990, vol. 3/3 : pp. 30-35.

Boitet, C., (1993). *La TAO comme technologie scientifique : le cas de la traduction automatique fondée sur le dialogue*. in La traductique. Les presses de l'Université de Montréal, AUPELF/UREF. pp. 109-148.

Boitet, C. & Blanchon, H., (1990). *TAO personnelle et promotion des langues nationales : le projet LIDIA du GETA*. Proc. Les industries de la langue : perspectives des années 1990. Montréal, Canada. 22-24 novembre 1990, vol. 1/2 : pp. 415-434.

Boitet, C. & Blanchon, H., (1993a). *Dialogue-based MT for monolingual authors and the LIDIA project*. Rap. IMAG. Rapport de Recherche. n° RR-918-I. mai 1993. 19 p.

Boitet, C. & Blanchon, H., (1993b). *Dialogue-based MT for monolingual authors and the LIDIA project*. Proc. NLPRS'93. Fukuoka, Japon. 6-7 décembre 1993, À paraître.

Boitet, C., Guillaume, P. & Quézel-Ambrunaz, M., (1982). *Ariane-78: an integrated environment for automatic translation and human revision*. Proc. Coling-82. Prague. 5-10 juillet 1982, vol. 1/1 : pp. 19-27.

Boitet, C., Guillaume, P. & Quézel-Ambrunaz, M., (1985). *A case study in software evolution: from ARIANE-78.4 to ARIANE-85*. Proc. Theoretical and Methodological Issues in Machine Translation of Natural Language. Colgate University, Hamilton, New York. 14-16 août 1985, vol. 1/1 : pp. 27-58.

Boitet, C. & Nédobekine, N., (1986). *Toward integrated dictionaries for M(a)T: motivations and linguistic organization*. Proc. Coling-86. Bonn, pp. 423-428.

Boitet, C. & Zaharin, Y., (1988). *Representation trees and string-tree correspondences*. Proc. Coling-88. Budapest. 22-27 août 1988, vol. 1/1 : pp. 59-64.

- Bonzi, S., (1990). *Syntactic Patterns in Scientific Sublanguages : A Study of Four Disciplines*. in Journal of the American Society for Information Science. vol. **41**(2) : pp. 121-131.
- Brown, R. D., (1989). *Augmentation*. in Machine Translation. vol. **4** : pp. 129-147.
- Brown, R. D. & Nirenburg, S., (1990). *Human-Computer Interaction for Semantic Disambiguation*. Proc. Coling-90. Helsinki. 20-25 août 1990, vol. **3/3** : pp. 42-47.
- Carbonell, J. G., Cullingford, R. E. & Gershman, A. V., (1981). *Steps Toward Knowledge-Based Machine Translation*. in IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. **3**(4) : pp. 376-392.
- Carbonell, J. G. & Tomita, M., (1985). *New Approaches to Machine Translation*. Proc. Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages. Colgate University, Hamilton, New York. 14-16 août 1985, vol. **1/1** : pp. 59-74.
- Carbonell, J. G. & Tomita, M., (1987). *Knowledge-based machine translation, the CMU approach*. in Machine translation. Cambridge University Press. Cambridge. pp. 68-89.
- Carré, R., Dégremont, J.-F., Gross, M., Pierrel, J.-M. & Sabah, G., (1991). *Langage humain et machine*. Presses du CNRS. Paris. 298 p.
- Carrol, J. M., Mack, R. L. & Kellogg, W. A., (1991). *Interface Metaphors and User Interface Design*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 67-85.
- Chandler, B., Holden, N., Horsfall, H., Pollard, E. & McGee Wood, M., (1987). *N-Tran Final Report. Alvey Project*. Rap. CCL/UMIST, Manchester. n° 87/9. 23 p.
- Chiang, T.-H., Lin, Y.-C. & Su, K.-Y., (1992). *Syntactic Ambiguity Resolution Using A Discrimination and Robustness Oriented Adaptive Learning Algorithm*. Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. **1/4** : pp. 352-358.
- Chignell, M. H. & Hancock, P. A., (1991). *Intelligent Interface Design*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 969-995.
- Church, K. & Patil, R., (1982). *Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table*. in American Journal of Computational Linguistics. vol. **8**(3-4) : pp. 139-149.
- Claris, (1991). *HyperCard Script Language Guide*. Claris. 583 p.
- Cottrell, G. W., (1989). *A Connectionist Approach to Word Sense Disambiguation*. Pitman. London. 220 p.
- Coutaz, J., (1988). *Interface Homme-ordinateur : Conception et Réalisation*. Thèse d'Etat. Université Joseph Fourier, Grenoble. 1988. 402 p.
- Cowie, J., Guthrie, J. & Guthrie, L., (1992). *Lexical Disambiguation Using Simulated Annealing*. Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. **1/4** : pp. 359-365.

Dahlbäck, N., (1991). *Empirical Analysis of a Discourse Model for Natural Language Interface*. Proc. Thirteenth Annual Meeting of the Cognitive Science Society. Chicago, Illinois. : pp. 1-6.

Dahlbäck, N., (1993). *Pronoun Usage in NLI-dialogues. A Wizard of Oz-study*. Proc. Third Nordic Conference on Text Comprehension in Man and Machine. Linköping, Suède.

Dahlbäck, N. & Jönsson, A., (1986). *A System for Studying Human Computer Dialogues in Natural Language*. Rap. Department of Computer and Information Science, Linköping University. Research Report. n° LiTH-IDA-R-86-42. 1986.

Dahlbäck, N., Jönsson, A. & Ahrenberg, L., (1993). *Wizard of Oz Studies - Why and How*. Proc. the 1993 International Workshop on Intelligent User Interfaces. Orlando, Floride. : pp. 193-200.

Daniel-Vatonne, M.-C., (1990). *Hypertextes : des principes communs et des variations*. in Technique et Science Informatique. vol. 9(6) : pp. 475-492.

DeHaemer, M. J. & Wallace, W. A., (1992). *The effects on decision tasks performance of computer synthetic voice output*. in International Journal of Man-Machine Studies. vol. 36(1) : pp. 65-80.

Denos, N., (1993). *Typage interactif de fragments de textes en "Style d'Énoncé" et "Genres de Textes"*. Rap. Institut National Polytechnique de Grenoble (ENSIMAG) & Université Joseph Fourier (UFR IMA). Rapport de DEA. 22 juin 1993. 79 p.

Doi, S. & Muraki, K., (1992). *Translation Ambiguity Resolution Based on Text Corpora of Source and Target Language*. Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. 2/4 : pp. 525-531.

Egan, D. E., (1991). *Individual Differences In Human-Computer Interaction*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 543-568.

Elkerton, J., (1991). *Online Aiding for Human-Computer Interfaces*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 345-364.

Flajolet, P. & Steyaert, J.-M., (1983). *A complexity calculus for recursive tree algorithms*. Rap. INRIA, centre de Rocquencourt. Rapport de Recherche. n° 239. Octobre 1983. 48 p.

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.

Fraser, M. & Gilbert, N. S., (1991). *Simulating speech systems*. in Computer, Speech and Language. vol. 5 : pp. 81-99.

Fuchs, C. (ed.), (1985). *Aspects de l'ambiguïté et de la paraphrase dans les langues naturelles*. Peter Lang. Berne. 215 p.

Fuchs, C. (ed.), (1987). *L'ambiguïté et la paraphrase : opérations linguistiques ; processus cognitifs ; traitements automatisés*. Centre de Publication de l'Université de Caen. 330 p.

- Good, R. L., (1988). *Automated Lookup: AutoTerm of ALP System*. in Technology as Translation Strategy. State University of New York at Binghamton. Binghamton. pp. 87-91.
- Goodman, K. & Nirenburg, S. (ed.), (1991). *The KBMT Project: A case study in knowledge-based machine translation*. Morgan Kaufmann. San Mateo, California. 331 p.
- Gould, J. D., (1991). *How to Design Usable Systems*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 757-789.
- Guilbaud, J.-P., (1993). *Le capitaine a rapporté un vase de Chine. Description du linguiciel Ariane-G5 francais-allemand pour la maquette LIDIA-1*. Rap. GETA, IMAG, Grenoble. Rapport Interne. 24 novembre 1993. 30 p.
- Guillaume, P., (1993). *Projet LIDIA. Le réseau LIDIA/IBM*. Rap. GETA, IMAG, Grenoble. Rapport interne. 13 octobre 1993. 42 p.
- Hauptmann, A. g. & Rudnicky, A. I., (1988). *Talking to computers: an empirical investigation*. in International Journal of Man-Machine Studies. vol. **28**(6) : pp. 583-604.
- Haverkort, K., (1990). *Some reflections on ambiguity and its consequences for MT*. Proc. The Maastricht Session of the 1990 Maastricht-Lódz Duo Colloquium on "Translation and Meaning". Maastricht, The Netherlands. 4-6 janvier 1990 : pp. 28-38.
- Heckel, P., (1991). *The Elements of Friendly Software design*. SYBEX. San Francisco. 319 p.
- Heidorn, G. E., Jensen, K. & Miller, L. A., (1982). *The EPISTLE Text-Critiquing System*. in IBM System Journal. vol. **21**(1) : pp. 305-326.
- Hirst, G., (1987). *Semantic interpretation and the resolution of ambiguity*. Cambridge University Press. Cambridge. 263 p.
- Hix, D. & Schulman, R. S., (1991). *Human-Computer Interface Development Tools: a Methodology for their Evaluation*. in Communications of the ACM. vol. **34**(3) : pp. 74-87.
- Huang, X., (1990). *A Machine Translation System for the Target Language Inexpert*. Proc. Coling-90. Helsinki. 20-25 août 1990, vol. **3/3** : pp. 364-367.
- Hutchins, W. J., (1986). *Machine Translation: past, present, future*. Ellis Horwood Limited. Chichester, England. 382 p.
- Hutchins, W. J. & Somers, H. L., (1992). *An Introduction to Machine Translation*. Academic Press. 362 p.
- IJMMS, (1988). *Special Issue: Multimodal Computer-Human Interaction*. in International Journal of Man-Machine Studies. vol. **28**(2 & 3) : pp. 215-336.
- Jensen, K., Heidorn, G. E. & Richardson, S. D. (ed.), (1993). *Natural Language Processing: The PLNLP Approach*. Kluwer Academic Publishers. London. 324 p.

- Johnson, R. & Whitelock, P., (1985). *Machine Translation as an Expert Task*. Proc. Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages. Colgate University, Hamilton, New York. 14-16 août 1985, vol. **1/1** : pp. 59-74.
- Jones, D. & Tsujii, J., (1990). *Interactive High-quality machine translation for monolinguals*. Proc. Third International Conference on Methodological Issues in Machine Translation of Natural Language. University of Texas at Austin.
- Jönsson, A., (1993). *Dialogue Management for Natural Language Interface - An Empirical Approach*. Phd these. Linköping. 1993. 226 p.
- Karat, J., (1991). *Software Evaluation Methodologies*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 891-903.
- Karat, J., McDonald, J. E. & Anderson, M., (1986). *A comparison of menu selection techniques: touch panel, mouse and keyboard*. in International Journal of Man-Machine Studies. vol. **25**(1) : pp. 73-88.
- Kay, M., (1973). *The MIND system*. in Courant Computer Science Symposium 8: Natural Language Processing. Algorithmics Press, Inc. New York. pp. 155-188.
- Kay, M., (1980). *The Proper Place of Men and Machines in Language Translation*. Rap. Xerox, Palo Alto Research Center. Research Report. n° CSL-80-11. octobre 1980. 20 p.
- Kay, M., (1982). *Machine Translation*. in American Journal of Computational Linguistics. vol. **8**(2) : pp. 74-78.
- Kempen, G. & Voose, T., (1989). *Incremental syntactic tree formation in human sentence processing: An interactive architecture based on activation decay and simulated annealing*. in Language and Cognition. vol. **10** : pp. 39-60.
- Kennedy, A., Wilkes, A., Elder, L. & Murray, W. S., (1988). *Dialogue with Machines*. in Cognition. vol. **30** : pp. 73-105.
- Kitano, H., (1992). *La traduction de la langue parlée*. in La traductique. Les presses de l'Université de Montréal, AUPELF/UREF. pp. 408-422.
- Kitano, H., Tomabechi, H. & Levin, L., (1989). *Ambiguity Resolution in the DMTRANS PLUS*. Proc. Fourth Conference of the European Chapter of the Association for Computational Linguistics. UMIST, Manchester, England. 10-12 Avril 1989, vol. **1/1** : pp. 72-79.
- Kittredge, R., (1983). *Sublanguages - Specific Aids to Translation - A survey of the most promising application areas*. Rap. Université de Montréal et bureau des traductions. n° 2-5273. mars 1983. 93 p.
- Kittredge, R., (1985). *The Significance of Sublanguages for Automatic Translation*. Proc. Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages. Colgate University, Hamilton, New York. 14-16 août 1985, pp. 154-166.
- Kittredge, R., (1986). *Analysing Language in Restricted Domains* in Sublanguage Description and Processing. Lawrence Erlbaum. Hillsdale, New-Jersey.

Kittredge, R. & Lehrberger, J. (eds.), (1982). *Sublanguage - Studies of Language in Restricted Domains*. Walter de Gruyter. Berlin. 240 p.

Klatt, D. H., (1975). *Vowel lengthening is syntactically determined in a connected discourse*. in *Journal of Phonetics*. vol. 3 : pp. 161-172.

Laenzlinger, C. & Wehrli, É., (1991). *Un analyseur syntaxique pour grammaires lexicales-fonctionnelles*. in *T.A. Information*. vol. 32(2) : pp. 35-49.

Lafourcade, M., (1993a). *Inside LT*. Rap. GETA-IMAG. Rapport Interne, version provisoire. 16 septembre 1993. 47 p.

Lafourcade, M., (1993b). *LEAF, ou comment garder l'origine de l'ambiguïté*. Proc. T-TA-TAO 93. Montréal, Canada. 30 septembre-2 octobre 1993, À paraître.

Landau, M.-C., (1990). *Solving Ambiguities in the Semantic Representation of Texts*. Proc. Coling-90. Helsinki. 20-25 août 1990, vol. 2/3 : pp. 239-244.

Landauer, T. K., (1991). *Research Methods in Human-Computer Interaction*. in *Handbook of Human-Computer Interaction*. Elsevier Science Publishers B.V. Amsterdam. pp. 905-938.

Laurian, A.-M., (1983a). *Pour une typologie des erreurs dans la traduction automatique*. in *Multilingua*. vol. 2(2) : pp. 65-78.

Laurian, A.-M., (1983b). *Typologie des discours scientifiques : deux approches*. in *Etudes de Linguistique Appliquée*. vol. 51 : pp. 8-20.

Laurian, A.-M., (1986). *Stylistics and Computing : Machine Translation as a Tool for a new Approach to Stylistics*. in *Computers and Translation*. vol. 1(4) : pp. 215-221.

Laurian, A.-M., (1987). *Enquête stylistique sur la traduction automatique de documents techniques*. in *International Review of Applied Linguistics*. vol. 20.

Lehiste, I., (1973). *Phonetic disambiguation of syntactic ambiguity*. in *Glossa*. vol. 7(2) : pp. 107-122.

Lehiste, I., Olive, J. P. & Streeter, L. A., (1976). *Role of duration in disambiguating syntactically ambiguous sentences*. in *Journal of the Acoustical Society of America*. vol. 60(5) : pp. 1199-1202.

Lepage, Y., (1985a). *Ambiguïtés et traduction automatique : les méthodes du GETA*. in *Brises*. vol. 7 : pp. 62-71.

Lepage, Y., (1985b). *Un langage de transcription*. Rap. USMG-INPG. Rapport de DEA. juin 1985. 50 p.

Lepage, Y., (1986). *A language for transcriptions*. Proc. COLING-86. Bonn, vol. 1/1 : pp. 402-404.

Lepage, Y., (1988). *Ambiguities and Second Generation MT Systems*. Proc. First European Conference on Information Technology for Organisational systems. Athens. 16-20 mai 1988.

- Lewis, C. H., (1990). *A Research Agenda for the Nineties in Human-Computer Interaction*. in Human-computer Interaction. vol. 5 : pp. 125-143.
- Mariani, J., (1989). *Recent Advances in Speech Processing*. Proc. IEEE International Conference on Acoustics, Speech, & Signal Processing. Glasgow, Scotland. 23-26 mai 1989, vol. 1/4 : pp. 429-440.
- Martin, G. L., (1989). *The utility of speech input in user-computer interfaces*. in International Journal of Man-Machine Studies. vol. 30(4) : pp. 355-375.
- Maruyama, H., Watanabe, H. & Ogino, S., (1990). *An Interactive Japanese Parser for Machine Translation*. Proc. Coling-90. Helsinki. 20-25 août 1990, vol. 2/3 : pp. 257-262.
- McCord, M., (1980). *Slot Grammars*. in Computational Linguistics. vol. 6(1) : pp. 31-43.
- McCord, M., (1989a). *Design of LMT: A Prolog-Based Machine Translation System*. in Computational Linguistics. vol. 15(1) : pp. 33-52.
- McCord, M., (1989b). *LMT*. Proc. MT SUMMIT II. Munich. 16-18 août 1989, vol. 1/1 : pp. 111-116.
- Melby, A. K., (1978). *Design and Implementation of a Computer-Assisted Translation System*. Proc. Coling-78. Bergen, Norvège. 14-18 août 1978, vol. 1/1.
- Melby, A. K., (1981). *Translators and Machines - Can they cooperate ?* in META. vol. 26(1) : pp. 23-34.
- Melby, A. K., (1982). *Multi-Level Translation Aids in a Distributed System*. Proc. Coling-82. Prague. 5-10 juillet 1982, vol. 1/1 : pp. 215-220.
- Melby, A. K., (1987). *On human-machine interaction in translation*. in Machine Translation. Cambridge University Press. Cambridge. pp. 145-154.
- Melby, A. K., (1991). *Pour le traducteur : un poste de travail à trois niveaux d'assistance*. Proc. L'environnement traductionnel ; La station de travail du traducteur de l'an 2001. Mons, Belgique. 25-27 avril 1991, vol. 1/1 : pp. 151-153.
- Melby, A. K., Smith, M. R. & Peterson, J., (1980). *ITS : An Interactive Translation System*. Proc. Coling-80. Tokyo. 30 septembre-4 octobre 1980, vol. 1/1 : pp. 424-429.
- Mitamura, T., (1991). *An Efficient Interlingua Translation system for Multi-lingual Document Production*. Proc. Machine Translation Summit III. Washington, D.C. 1-4 juillet 1991, vol. 1/1 : pp. 55-61.
- Nielsen, J., (1990). *Hypertext and Hypermedia*. Academic Press, Inc. San Diego. 263 p.
- Nijholt, A., (1988). *Computers and Languages; Theory and Practice*. Elsevier Science Publishers B.V. Amsterdam. 482 p.
- Nirenburg, S., (1989). *Knowledge-based Machine Translation*. in Machine Translation. vol. 4 : pp. 5-24.
- Nirenburg, S., (1992). *Tools for Machine-aided Translation: the CMU TWS*. in Meta. vol. 37(4) : pp. 709-720.

- Nirenburg, S. & al. (1989). *KBMT-89 Project Report*. Rap. Center for Machine Translation, Carnegie Mellon University, Pittsburg. avril 1989. 286 p.
- Nyberg, E. H. & Mitamura, T., (1992). *The KANT system: Fast, Accurate, High-Quality Translation in Practical Domains*. Proc. COLING-92. Nantes. 23-28 juillet 92, vol. **3/4** : pp. 1069-1073.
- O'Malley, M. H., Kloker, D. R. & Dara-Abrams, B., (1973). *Recovering Parentheses From Spoken Algebraic Expressions*. in IEEE Transactions on Audio and Electroacoustics. vol. **AU-21**(3) : pp. 217-220.
- O'Shaughnessy, D., (1979). *Linguistic features in fundamental frequency patterns*. in Journal of Phonetics. vol. **7** : pp. 119-145.
- O'Shaughnessy, D., (1989). *Specifying accent marks in French text for teletext and speech synthesis*. in International Journal of Man-Machine Studies. vol. **31**(4) : pp. 405-414.
- Ogden, W. C., (1991). *Using Natural Language Interfaces*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam.
- Paap, K. R., (1991). *Design of Menus*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 205-235.
- Papagaaij, B. C., (1986). *Word Expert Semantics, an Interlingual Knowledge-Based Approach*. Floris Publications. Dordrecht, Holland. 240 p.
- Perlman, G., (1991). *Software Tools for User Interface Development*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 819-833.
- Phan, H. K. & Boitet, C., (1992). *Multilinguization of an editor for structured documents. Application to a trilingual dictionary*. Proc. COLING-92. Nantes, juillet 1992, vol. **3/4** : pp. 966-971.
- Phillips, M. D., Bashinski, H. S., Ammerman, H. L. & Fligg, C. M. J., (1991). *A Task Analytic Approach to Dialogue Design*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 835-857.
- Pierrel, J.-M., (1987). *Dialogue oral homme-machine*. Hermès. Paris. 239 p.
- Quézel-Ambrunaz, M., (1989). *Ariane-G5, le moniteur, manuel d'utilisation*. Rap. GETA, IMAG, Grenoble. mai 1989. 206 p.
- Quézel-Ambrunaz, M., (1993). *Environnement de traitement LIDIA*. Rap. GETA, IMAG, Grenoble. Rapport interne, document provisoire. Octobre 1993. 6 p.
- Ravin, Y., (1988). *Grammars Errors and Style Weaknesses in a Text-Critiquing System*. in IEEE Transaction on Professional Communication. vol. **31**(3) : pp. 108-115.
- Richardson, S. D., (1985). *Enhanced Text Critiquing Using a Natural Language Parser : the CRITIQUE system*. Rap. IBM, Thomas J. Watson Research Center, Yorktown Heights. n° RC 11332. 17 p.

- Rimon, M., McCord, M. C., Schwall, U. & Martínez, P., (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.
- Roberts, T. L., (1991). *Text Editors*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 655-672.
- Roche, E., (1992). *Text Disambiguation by Finite State Automata, an Algorithm and Experiments on Corpora*. Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. 3/4 : pp. 993-997.
- Sadler, V., (1989). *Working with analogical semantics : Disambiguation techniques in DLT*. Floris Publications. Dordrecht, Holland. 256 p.
- Saito, H. & Tomita, M., (1986). *On Automatic Composition of Stereotypic Documents in Foreign Languages*. Rap. CMU. Rapport de recherche. n° CMU-CS-86-107. 1986.
- Sano, H. & Fukumoto, F., (1990). *Ambiguity Resolution: A Context-Based Approach*. Rap. ICOT (Institute for New Generation Computer Technology) Research Center. Technical Memorandum. n° TM-0953. Septembre 1990. 8 p.
- Savoy, J., (1990). *Les sources des hypertextes : une bibliographie commentée*. in Technique et Science Informatique. vol. 9(6) : pp. 515-524.
- Schubert, K., (1986). *Linguistic and Extra-Linguistic Knowledge. A catalog of language-related rules and their computational application in machine translation*. in Computers and Translation. vol. 1(3) : pp. 125-152.
- Schubert, K., (1988). *The Architecture of DLT - Interlingual or Double Direct?* Proc. New Directions in Machine Translation. Budapest. 18-19 août 1988 : pp. 131-143.
- Selinker, L., Trimble, L. & Vroman, R., (1972). *Working papers in English for science and technology*. Rap. University of Washington College of Engineering.
- Sérasset, G., (1992a). *Defining a database — An example*. Rap. GETA, IMAG. Rapport Multilex (ESPRIT). mai 1992, 33 p.
- Sérasset, G., (1992b). *Defining a database — The language*. Rap. GETA, IMAG. Rapport Multilex (ESPRIT). mai 1992, 17 p.
- Sérasset, G., (1992c). *Multilex*. Rap. GETA, IMAG. Contrat EDR, rapport intermédiaire, partie I. août 1992, 37 p.
- 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, À paraître.
- Sibley, J., (1988). *Le système ALPS*. Proc. Traduction Assistée par Ordinateur — Perspectives technologiques, industrielles et économiques envisageable à l'horizon 1990 — l'offre, la demande, les marchés et les évolutions en cours. Paris. mars 1988, vol. 1/1 : pp. 95-104.

- Slocum, J., (1984). *How one might automatically identify and adapt to a sublanguage : an initial exploration*. Rap. The University of Texas, Linguistic Research Center. n° LRC-84-1. Janvier 1984.
- Somers, H. & Jones, D., (1992). *La génération de textes multilingues par un utilisateur monolingue*. in Meta. vol. **37**(4) : pp. 647-656.
- Somers, H. L., Tsujii, J.-I. & Jones, D., (1990). *Machine Translation without a source text*. Proc. Coling-90. Helsinki. 20-25 août 1990, vol. **3/3** : pp. 271-276.
- Steyaert, J.-M. & Flajolet, P., (1983). *Patterns and Pattern-Matching in Trees : An Analysis*. in Information and Control. vol. **58**(1-3) : pp. 19-58.
- Streeter, L. A., (1978). *Acoustic determinants of phrase boundary perception*. in Journal of the Acoustical Society of America. vol. **60**(6) : pp. 1582-1592.
- Streeter, L. A., (1991). *Applying Speech Synthesis to User Interface*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 321-343.
- Sullivan, J. W. & Tyler, S. W. (ed), (1991). *Intelligent User Interfaces*. ACM Press. New-York. 475 p.
- Sumita, K., Ukita, T. & Amano, S., (1990). *Disambiguation in natural language interpretation based on amount of information*. Rap. ICOT (Intsitute for New Generation Computer Technology). Resarch Center. Technical Report. n° TR-603. novembre 1990. 15 p.
- Tagliacozzo, R., (1978). *Some stylistic variations in scientific writing*. in Journal of the American Society for Information Science(29) : pp. 36-40.
- Taylor, M. M., Néel, F. & Bouwhuis, D. G. (eds.), (1989). *The Structure of Multimodal Dialogue*. North-Holland. Amsterdam. 539 p.
- Tenny, M. D., (1985). *Machine translation, machine-aided translation and machine-impeded translation*. in Tools for the Trade. Aslib. London. pp. 105-113.
- Tomita, M., (1984). *Disambiguating Grammatically Ambiguous Sentences by Asking*. Proc. Coling-84. Stanford University, California. 2-6 juillet 1984, vol. **1/1** : pp. 476-480.
- Tomita, M., (1985). *Feasibility Study of Personal/Interactive Machine Translation System*. Proc. Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language. Colgate University, Hamilton, New York. 14-16 août 1985, vol. **1/1** : pp. 289-297.
- Tomita, M., (1986). *Sentence disambiguation by asking*. in Computers and Translation. vol. **1**(1) : pp. 39-51.
- Tsujii, J.-I., (1986). *Future Directions of Machine Translation*. Proc. Coling-86. Bonn. 25-29 août 1986, vol. **1/1** : pp. 655-668.
- Tsutsumi, T., Watanabe, H., Maruyama, H., Uramoto, N., Morohashi, M., Takeda, K. & Nasukawa, T., (1993). *Example-Based Approach to Machine Translation*. Proc. Premières journées franco-japonaises sur la traduction assistée par ordinateur. Ambassade de France au Japon, Tokyo, Japon. 15-16 mars 1993, vol. **1/1** : pp. 161-169.

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

Vauquois, B. & Boitet, C., (1985). *Automated Translation at Grenoble University.* in Computational Linguistics. vol. **11**(1) : pp. 28-36.

Vauquois, B. & Chappuy, S., (1985). *Static grammars : a formalism for the description of linguistics models.* Proc. Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages. Colgate University, Hamilton, N. Y. 14-16 août 1985, vol. **1/1** : pp. 298-322.

Vauquois, B. & Nédobekine, N., (1977). *Ambiguïtés de la langue écrite.* Rap. Action Thématique Programmée de Linguistique Générale, Table ronde organisée au CNRS. 11 janvier 1977. 8 p.

Veronis, J. & Ide, N. M., (1990). *Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries.* Proc. Coling-90. Helsinki. 20-25 août 1990, vol. **2/3** : pp. 389-394.

Wærn, Y., (1990). *Cognitive Aspects of Computer Supported Tasks.* John Wiley & Sons. Chichester. 327 p.

Weaver, A., (1988). *Two Aspects of Interactive Machine Translation.* in Technology as Translation Strategy. State University of New York at Binghamton. Binghamton. pp. 116-123.

Wehrli, É., (1990). *STS: An Experimental Sentence Translation System.* Proc. Coling-90. Helsinki. 20-25 août 1990, vol. **1/3** : pp. 76-78.

Wehrli, É., (1992). *The IPS System.* Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. **3/4** : pp. 870-874.

Wehrli, É., (1993). *Vers un système de traduction automatique.* in La traductique. Les presses de l'Université de Montréal, AUPELF/UREF. pp. 423-432.

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, vol. **1/1** : pp. 25-29.

Whittaker, S. & Stenton, P., (1989). *User Studies and the Design of Natural Language Systems.* Proc. Fourth Conference of the European Chapter of the Association for Computational Linguistics. Manchester, Angleterre. : pp. 116-123.

Wioland, F., (1987). *Jointure et ambiguïté.* in L'ambiguïté et la paraphrase. Centre de Publication de l'Université de Caen. Caen. pp. 53-57.

Witkam, A. P. M., (1983). *Distributed Language Translation — Feasibility Study of a Multilingual Facility for Videotex Information Networks.* BSO. Utrecht, The Netherlands. 340 p.

Witkam, T., (1988). *DLT - An Industrial R&D Project for Multilingual MT.* Proc. Coling-88. Budapest. 22-27 août 1988, vol. **2/2** : pp. 576-579.

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.

Wood, M. M. G. & Chandler, B., (1988). *Machine Translation For Monolinguals*. Proc. Coling-88. Budapest. 22-27 août 1988, vol. 2/2 : pp. 760-763.

Zaki, A. B. & Noor, H., (1991). *Malay Official Letters Translation*. Proc. International Conference on Current Issues in Computational Linguistics. Universiti Sains Malaysia, Penang, Malaisie. 10-14 juin 1991, vol. 1/1 : pp. 413-420.

Ziegler, J. E. & Fähnick, K.-P., (1991). *Direct Manipulation*. in Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V. Amsterdam. pp. 123-133.

Zirkle, A. T., (1990). *The role of computer-aided translation in translation services*. in Translation and the computer 10. Aslib. London. pp. 11-17.

-0-0-0-0-0-0-0-0-

Index

- acception 83, 84, 174
- ALPNET, *voir* TSS
- ALPS 17, 87
voir aussi TSS 17, 23
- ambiguïté
voir ambiguïté dans LIDIA-1
voir ambiguïtés linguistiques
voir DLT
voir LMT
- ambiguïtés dans LIDIA-1
ambiguïté d'étiquetage logico-sémantique **215**
ambiguïté d'étiquetage syntaxique **216**
ambiguïté d'ordre des arguments d'un verbe transitif direct **217**
ambiguïté de classe simple **209**
ambiguïté de classe syntaxique 153
ambiguïté de coordination adjectivale **212**
ambiguïté de coordination nominale **211**
ambiguïté de coordination verbale **208**
ambiguïté de décoration **215**
ambiguïté de géométrie **210**
ambiguïté de relation logique et de relation sémantique 155
ambiguïté de structure 154
ambiguïté de structure argumentaire du verbe **210**
ambiguïté de subordination **213**
polysémie 153, **208**
- ambiguïtés linguistiques
ambiguïté anaphorique **151**
ambiguïté de classe syntaxique **149**
ambiguïté de détermination **152**
ambiguïté de portée des quantificateurs **153**
ambiguïté de structure de constituants grammaticaux **150**
ambiguïté morphologique **150**
homophonie **148**
jointure **148**
mots composés, tournures **150**
polysémie **149**
- annotations 89, 92, 122
annotations de l'utilisateur 118
annotations du système 117
- Apple events 177, **195**
- Ariane-G5 **93**, 178, 181, 184
- clarification 5, **8**, 13, 86, 88, 103, 106, 147, 155, 158
demande de clarification 193
dialogue de clarification 74
processus de clarification 169
types de dialogues de clarification **159**
- compréhension
compréhension explicite 5, **88**
compréhension implicite **88**
compréhension indirecte 5, **88**
compréhension profonde 88
- CRITIQUE 92
- désambiguïisation
voir clarification
voir désambiguïisation avec LIDIA-1
voir filtres
voir MIND, ITS, TSS, KBMT-89, LMT, DLT, JETS, ITS2 243
- désambiguïisation avec LIDIA-1
méthode de désambiguïisation
ambiguïté de décoration 234
classe simple 233
coordination adjectivale 233
coordination nominale 233
coordination verbale 233
polysémie 232
structure argumentaire du verbe 233
subordination 234
- opérateurs **221**
- principes **218**
- stratégie **218**
- voir aussi filtres
- dialogue
dialogue avec explications **160**
dialogue avec remplacement **159**
dialogue par complétion **159**
dialogue par paraphrasage **159**
dialogue pour expert non linguiste 163
voir aussi dialogue de désambiguïisation

dialogue de désambiguïsation

avec LIDIA-1

classe simple 236

coordination adjectivale 238

coordination nominale 238

coordination verbale 235

étiquetage logico-sémantique 242

étiquetage syntaxique 242

ordre des arguments d'un verbe transitif

direct 243

polysémie 234

structure argumentaire du verbe 237

subordination 239

voir aussi MIND, ITS, TSS, KBMT-89, LMT, DLT, JETS, ITS2

DLT 44-58

ambiguïtés dans DLT

ambiguïté d'attachement 56

ambiguïté de portée des modifieurs 56

ambiguïté fonctionnelle 56

ambiguïté lexicale 57

ambiguïté référencielle 56

ambiguïté structurale 55

fichiers

fichier miroir 203

fichier requête 183

filtres 224

grammaire de filtres 225

pour la désambiguïsation 228

coordination adjectivale 231

coordination nominale 230

structure argumentaire du verbe 229

subordination 231

HyperCard 91, 92

commandes et fonctions externes 173

messages HyperCard 171, 195, 197

objets HyperCard 107

pile HyperCard 92, 107

pile traduisible 109

hypertexte 91

ITS 17-22, 23, 79

ITS2 36, 73-77

JETS 36, 63-73, 82, 249

KANT 36

KBMT-89 26-36, 43, 73, 89, 249

langages

langage contrôlé 5

langage guidé 5

voir aussi type de texte

sous-langage 5, 84

LIDIA-1 91

LMT 36, 43, 249

ambiguïtés dans LMT

ambiguïté de partie du discours 38, 39, 40, 41, 42

ambiguïté de structure 38, 39, 40, 41, 42

ambiguïté implicite 38, 40, 41, 42, 43

polysémie 38, 39, 41, 42

menus 119

menu LIDIA 119, 123

menu Messages 123

messages échangés

messages LIDIA 105, 169, 177, 192, 195, 196, 197, 199

voir aussi Apple events

voir aussi HyperCard

voir aussi scripts

MIND 15-17

N-Tran 58-63

palettes 124

Parax 173, 188

pile, voir HyperCard

rétroaction 125

scripts 177, 198

serveurs

serveur de communication 177, 194

serveur de traduction 178

machines interfaces 179

machines nœud de réseau 180

machines serveurs 179

réseau LIDIA/IBM 178

serveur LIDIA 176

serveur de désambiguïsation 193

superviseur LIDIA 192

- standardisation 5, **8**, 13, 86, 88, **98**, 103, 106, 112, 125, 127, 147
 - orthographe 127
 - standardisation terminologique 133, 134, 137
 - tournures 138
 - typage textuel 141
- structures linguistiques manipulées
 - structure concrète **96**
 - structure gma 98
 - structure mmc **95**, 98
 - structure multiresolution **96**
 - structure multiniveau **96**
 - structure uma 98
 - structure umc 98
- TAFD 5, **7**, 13-**89**, 92, 97
 - TA Fondée sur le Dialogue 4
 - TAFD sans texte source 89
 - TAO personnelle 4
 - TAO pour le rédacteur 4, 5
 - Traduction Automatisée Fondée sur le Dialogue 3
- TAO
 - TAO du réviseur **3**, 4, 84, 88
 - TAO du traducteur **4**, 15, 88
 - TAO du veilleur **3**, 4, 82, 88
- transcription 200, 202
 - Langage de Transcription 202
- Translation Support system, voir TSS
- TSS **25**
 - AutoTerm **23**
 - TransActive **23**
- TWS 36
- typage textuel, voir *standardisation*
- type de texte
 - genre de texte **85**
 - style d'énoncé **85**, 142
 - type de document **86**
 - voir aussi standardisation

Annexes

Annexe 1

Phases AM, AX et AY fvx

17 NOVEMBRE 1993 20H 48MN 41S
 *** GROUPE D'ETUDE POUR LA TRADUCTION AUTOMATIQUE ***
 --*-*-*-*-*-*-*-*

AY - (Info pour l'UL)

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA CHINE** CODE LANGUE : FVX-*

1: 'ULTXT' (2: 'ULFRA' (3: 'ULSOL' (4: '-111-' (5: 'LE', 6: 'JETUIL' (7: '-S-')), 8: 'CAPITAINE' (9: '-S-'), 10: 'AVOIR' (11: 'C2A'), 12: 'RAPPORTER' (13: 'C3A', 14: 'C3B', 15: 'C3C', 16: 'C3C', 17: 'C2A'), 18: 'UN', 19: '-111-' (20: 'VASE/LA' (21: '-S-'), 22: 'VASE/LE' (23: '-S-')), 24: 'DE' (25: '-S-', 26: '-S-'), 27: '*CHINE' (28: '-S-'), 29: '.'))))

- 1 '': UL('ULTXT').
- 2 '': UL('ULFRA').
- 3 '': UL('ULSOL').
- 4 '*LE': UL('-111-'), CAT(D,R), GNR(MAS), NBR(SING), TYPOG(PCAP).
- 5 '*LE': UL('LE'), CAT(D), GNR(MAS), NBR(SING), TYPOG(PCAP).
- 6 '*LE': UL('JETUIL'), CAS(CASACC), CAT(R), GNR(MAS), NBR(SING), PERS(3), TYPOG(PCAP).
- 7 '*LE': UL('-S-'), CAS(CASACC), CAT(R), GNR(MAS), NBR(SING), PERS(3), SEMN(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), TYPOG(PCAP).
- 8 'CAPITAINE': UL('CAPITAINE'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING).
- 9 'CAPITAINE': UL('-S-'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(ANIME), SENS(1,2,3), SSANIM(HUM).
- 10 'A': UL('AVOIR'), SUBV(VF), CAT(V), MT(IPR), NBR(SING), PERS(3).
- 11 'A': UL('C2A'), AUX(AVOIR), SUBV(VF), VAL0(N), VAL1(N), CAT(V), MT(IPR), NBR(SING), PERS(3), SEM1(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), SEMZ(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), SENS(1).
- 12 'RAPPORTE!1': UL('RAPPORTER'), SUBV(PPA), CAT(V), GNR(MAS), NBR(SING).
- 13 'RAPPORTE!1': UL('C3A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(CONCRET), SEM2(ANIME), SEMZ(ANIME), SENS(1), SSANIM2(HUM).
- 14 'RAPPORTE!1': UL('C3B'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(ABSTRAIT), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
- 15 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(Q), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
- 16 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A1,A2), GNR(MAS), NBR(SING), SEM1(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), SEM2(ANIME), SEMZ(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), SENS(3), SSANIM2(HUM).
- 17 'RAPPORTE!1': UL('C2A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), CAT(V), GNR(MAS), NBR(SING), SEM1(CONCRET), SEMZ(ANIME), SENS(2), SSANIM2(HUM).
- 18 'UN': UL('UN'), CAT(D), GNR(MAS), NBR(SING).
- 19 'VASE': UL('-111-'), SUBN(NC), CAT(N), GNR(FEM, MAS), NBR(SING).
- 20 'VASE': UL('VASE/LA'), SUBN(NC), CAT(N), GNR(FEM), NBR(SING).
- 21 'VASE': UL('-S-'), SUBN(NC), CAT(N), GNR(FEM), NBR(SING), SEMN(CONCRET), SENS(1), SCONC(MAT).
- 22 'VASE': UL('VASE/LE'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING).
- 23 'VASE': UL('-S-'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(CONCRET), SENS(1), SCONC(RECIP).
- 24 'DE': UL('DE'), CAT(S).
- 25 'DE': UL('-S-'), VALET(DEN), VAL1(N), CAT(S).
- 26 'DE': UL('-S-'), VALET(DEI), VAL1(I), CAT(S).
- 27 '*CHINE': UL('*CHINE'), SUBN(NP), CAT(N), GNR(FEM), NBR(SING).
- 28 '*CHINE': UL('-S-'), SUBN(NP), CAT(N), GNR(FEM), NBR(SING), SEMN(ANIME, LIEU), SENS(1), SSANIM(HUM, PERS).
- 29 '.': UL('.'), CAT(P), PONC(DR).

Annexe 2

Phase AS1 fvx

17 NOVEMBRE 1993 21H 11MN 57S
 *** GROUPE D'ETUDE POUR LA TRADUCTION AUTOMATIQUE ***
 --*-*-*-*-*-*-*-*-*-*

AS - Analyse Structurale

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA CHINE GRAMMAIRE 1** CODE LANGUE : FVX

1: 'ULTXT' (2: 'ULFRA' (3: 'ULSOL' (4: 'LE', 5: 'CAPITAINE' (6: '-S-'), 7: 'AVOIR' (8: 'C2A'), 9: 'RAPPORTER' (10: 'C3A', 11: 'C3B', 12: 'C3C', 13: 'C3C', 14: 'C2A'), 15: 'UN', 16: 'VASE/LE' (17: '-S-'), 18: 'DE' (19: '-S-', 20: '-S-'), 21: '*CHINE' (22: '-S-'), 23: '.')))

1 '': UL('ULTXT').
 2 '': UL('ULFRA').
 3 '': UL('ULSOL').
 4 '*LE': UL('LE'), CAT(D), GNR(MAS), NBR(SING), TYPOG(PCAP).
 5 'CAPITAINE': UL('CAPITAINE'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING).
 6 'CAPITAINE': UL('-S-'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(ANIME), SENS(1,2,3), SSANIM(HUM).
 7 'A': UL('AVOIR'), SUBV(VF), CAT(V), MT(IPR), NBR(SING), PERS(3).
 8 'A': UL('C2A'), AUX(AVOIR), SUBV(VF), VAL0(N), VAL1(N), CAT(V), MT(IPR), NBR(SING), PERS(3), SEM1(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SEMZ(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SENS(1).
 9 'RAPPORTE!1': UL('RAPPORTER'), SUBV(PPA), CAT(V), GNR(MAS), NBR(SING).
 10 'RAPPORTE!1': UL('C3A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(CONCRET), SEM2(ANIME), SEMZ(ANIME), SENS(1), SSANIM2(HUM).
 11 'RAPPORTE!1': UL('C3B'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(ABSTRAIT), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
 12 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(Q), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
 13 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A1,A2), GNR(MAS), NBR(SING), SEM1(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SEM2(ANIME), SEMZ(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SENS(3), SSANIM2(HUM).
 14 'RAPPORTE!1': UL('C2A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), CAT(V), GNR(MAS), NBR(SING), SEM1(CONCRET), SEMZ(ANIME), SENS(2), SSANIMZ(HUM).
 15 'UN': UL('UN'), CAT(D), GNR(MAS), NBR(SING).
 16 'VASE': UL('VASE/LE'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING).
 17 'VASE': UL('-S-'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(CONCRET), SENS(1), SSCONC(RECIP).
 18 'DE': UL('DE'), CAT(S).
 19 'DE': UL('-S-'), VALET(DEN), VAL1(N), CAT(S).
 20 'DE': UL('-S-'), VALET(DEI), VAL1(I), CAT(S).
 21 '*CHINE': UL('*CHINE'), SUBN(NP), CAT(N), GNR(FEM), NBR(SING).
 22 '*CHINE': UL('-S-'), SUBN(NP), CAT(N), GNR(FEM), NBR(SING), SEMN(ANIME,LIEU), SENS(1), SSANIM(HUM,PERS).
 23 '.': UL('.'), CAT(P), PONC(DR).

Annexe 3

Phase AS2 fvx

17 NOVEMBRE 1993 21H 13MN 36S
 *** GROUPE D'ETUDE POUR LA TRADUCTION AUTOMATIQUE ***
 --*-*-*-*-*-*-*-*-*

AS - Analyse Structurale

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA CHINE GRAMMAIRE 2** CODE LANGUE : FVX

1: 'ULTXT' (2: 'ULFRA' (3: 'ULSOL' (4: 'PHVB' (5: 'DGN' (6: 'LE' (7: 'CAPITAINE' (8: '-S-')), 9: 'AVOIR' (10: 'C2A')), 11: 'RAPPORTER' (12: 'C3A', 13: 'C3B', 14: 'C3C', 15: 'C3C', 16: 'C2A')), 17: 'DGN' (18: 'UN', 19: 'VASE/LE' (20: '-S-')), 21: 'GP' (22: 'DE' (23: '-S-', 24: '-S-'), 25: '*CHINE' (26: '-S-')), 27: '.'))

1 '': UL('ULTXT').
 2 '': UL('ULFRA').
 3 '': UL('ULSOL').
 4 'A': UL('PHVB'), LINKS(OUI), ENONCP(DECL), K(PHVB), PHASE(ACC), SUBV(VF), VALET(Q), VOIX(ACT), ARGS(A0), CAT(V), MT(IPR), NBR(SING), PERS(3).
 5 'CAPITAINE': UL('DGN'), FS(SUJ), K(GN), RL(ARG0), SUBN(NC), VALET(N), CAT(N,D), GNR(MAS), NBR(SING), PERS(3).
 6 '*LE': UL('LE'), FS(DES), CAT(D), GNR(MAS), NBR(SING), TYPOG(PCAP).
 7 'CAPITAINE': UL('CAPITAINE'), FS(GOV), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), PERS(3).
 8 'CAPITAINE': UL('-S-'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(ANIME), SENS(1,2,3), SSANIM(HUM).
 9 'A': UL('AVOIR'), FS(AUX), SUBV(VF), CAT(V), MT(IPR), NBR(SING), PERS(3).
 10 'A': UL('C2A'), AUX(AVOIR), SUBV(VF), VAL0(N), VAL1(N), CAT(V), MT(IPR), NBR(SING), PERS(3), SEM1(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SEMZ(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SENS(1).
 11 'RAPPORTE!1': UL('RAPPORTER'), FS(GOV), SUBV(PPA), CAT(V), GNR(MAS), NBR(SING), PERS(3).
 12 'RAPPORTE!1': UL('C3A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(CONCRET), SEM2(ANIME), SEMZ(ANIME), SENS(1), SSANIM2(HUM).
 13 'RAPPORTE!1': UL('C3B'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(ABSTRAIT), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
 14 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(Q), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
 15 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A1,A2), GNR(MAS), NBR(SING), SEM1(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SEM2(ANIME), SEMZ(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SENS(3), SSANIM2(HUM).
 16 'RAPPORTE!1': UL('C2A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), CAT(V), GNR(MAS), NBR(SING), SEM1(CONCRET), SEMZ(ANIME), SENS(2), SSANIMZ(HUM).
 17 'VASE': UL('DGN'), K(GN), SUBN(NC), VALET(N), CAT(N,D), GNR(MAS), NBR(SING), PERS(3).
 18 'UN': UL('UN'), FS(DES), CAT(D), GNR(MAS), NBR(SING).
 19 'VASE': UL('VASE/LE'), FS(GOV), SUBN(NC), CAT(N), GNR(MAS), NBR(SING).
 20 'VASE': UL('-S-'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(CONCRET), SENS(1), SSSCONC(RECIP).
 21 '*CHINE': UL('GP'), K(GN), SUBN(NP), VALET(DEN), CAT(N,S), GNR(FEM), NBR(SING), PERS(3).
 22 'DE': UL('DE'), FS(REG), CAT(S).
 23 'DE': UL('-S-'), VALET(DEN), VAL1(N), CAT(S).
 24 'DE': UL('-S-'), VALET(DEI), VAL1(I), CAT(S).
 25 '*CHINE': UL('*CHINE'), FS(GOV), SUBN(NP), CAT(N), GNR(FEM), NBR(SING).
 26 '*CHINE': UL('-S-'), SUBN(NP), CAT(N), GNR(FEM), NBR(SING), SEMN(ANIME,LIEU), SENS(1), SSANIM(HUM,PERS).
 27 '.': UL('.'), CAT(P), PONC(DR).

Annexe 4

Phase AS3 fvx

- 27 '.': UL('.'), CAT(P), PONC(DR).
- 28 '': UL('ULSOL'), MODIF(NON).
- 29 'A': UL('PHVB'), LINKS(OUI), RECHTS(OUI), ENONCP(DECL), K(PHVB), PHASE(ACC), SUBV(VF), VALET(Q), VOIX(ACT), ARGS(A0,A1), CAT(V), MT(IPR), NBR(SING), PERS(3).
- 30 'CAPITAINE': UL('DGN'), FS(SUJ), K(GN), RL(ARG0), SUBN(NC), VALET(N), CAT(N,D), GNR(MAS), NBR(SING), PERS(3).
- 31 '*LE': UL('LE'), FS(DES), CAT(D), GNR(MAS), NBR(SING), TYPOG(PCAP).
- 32 'CAPITAINE': UL('CAPITAINE'), FS(GOV), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), PERS(3).
- 33 'CAPITAINE': UL('-S-'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(ANIME), SENS(1,2,3), SSANIM(HUM).
- 34 'A': UL('AVOIR'), FS(AUX), SUBV(VF), CAT(V), MT(IPR), NBR(SING), PERS(3).
- 35 'A': UL('C2A'), AUX(AVOIR), SUBV(VF), VAL0(N), VAL1(N), CAT(V), MT(IPR), NBR(SING), PERS(3), SEM1(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SEMZ(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SENS(1).
- 36 'RAPPORTE!1': UL('RAPPORTER'), FS(GOV), SUBV(PPA), CAT(V), GNR(MAS), NBR(SING), PERS(3).
- 37 'RAPPORTE!1': UL('C3A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(CONCRET), SEM2(ANIME), SEMZ(ANIME), SENS(1), SSANIM2(HUM).
- 38 'RAPPORTE!1': UL('C3B'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(ABSTRAIT), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
- 39 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(Q), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM2(ANIME), SEMZ(ANIME), SENS(4), SSANIM2(HUM).
- 40 'RAPPORTE!1': UL('C3C'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A1,A2), GNR(MAS), NBR(SING), SEM1(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SEM2(ANIME), SEMZ(ANIME,CONCRET,ACTION,ABSTRAIT,UNITE,LIEU,TEMPS), SENS(3), SSANIM2(HUM).
- 41 'RAPPORTE!1': UL('C2A'), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), CAT(V), GNR(MAS), NBR(SING), SEM1(CONCRET), SEMZ(ANIME), SENS(2), SSANIMZ(HUM).
- 42 'VASE': UL('DGN'), FS(OBJ), K(GN), RL(ARG1), SUBN(NC), VALET(N), CAT(N,D), GNR(MAS), NBR(SING), PERS(3).
- 43 'UN': UL('UN'), FS(DES), CAT(D), GNR(MAS), NBR(SING).
- 44 'VASE': UL('VASE/LE'), FS(GOV), SUBN(NC), CAT(N), GNR(MAS), NBR(SING).
- 45 'VASE': UL('-S-'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(CONCRET), SENS(1), SSANIM(RECIP).
- 46 '*CHINE': UL('GP'), FS(CIRC), K(GN), RS(UNDE), SUBN(NP), VALET(DEN), CAT(N,S), GNR(FEM), NBR(SING), PERS(3).
- 47 'DE': UL('DE'), FS(REG), CAT(S).
- 48 'DE': UL('-S-'), VALET(DEN), VAL1(N), CAT(S).
- 49 'DE': UL('-S-'), VALET(DEI), VAL1(I), CAT(S).
- 50 '*CHINE': UL('*CHINE'), FS(GOV), SUBN(NP), CAT(N), GNR(FEM), NBR(SING).
- 51 '*CHINE': UL('-S-'), SUBN(NP), CAT(N), GNR(FEM), NBR(SING), SEMN(ANIME,LIEU), SENS(1), SSANIM(HUM,PERS).
- 52 '.': UL('.'), CAT(P), PONC(DR).

Annexe 5

Phase AS5 fvx

- 29 '*LE': UL('LE'), FS(DES), CAT(D), GNR(MAS), NBR(SING), TYPOG(PCAP).
- 30 'CAPITAINE': UL('CAPITAINE'), FS(GOV), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), PERS(3).
- 31 'CAPITAINE': UL('-S-'), FLAG(CPRED), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(ANIME), SENS(1,2,3), SSANIM(HUM).
- 32 'A': UL('AVOIR'), FS(AUX), SUBV(VF), CAT(V), MT(IPR), NBR(SING), PERS(3).
- 33 'A': UL('C2A'), AUX(AVOIR), SUBV(VF), VAL0(N), VAL1(N), CAT(V), MT(IPR), NBR(SING), PERS(3), SEM1(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), SEMZ(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), SENS(1).
- 34 'RAPPORTE!1': UL('RAPPORTER'), FS(GOV), SUBV(PPA), CAT(V), GNR(MAS), NBR(SING), PERS(3).
- 35 'RAPPORTE!1': UL('C3A'), FLAG(CPRED), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A2), GNR(MAS), NBR(SING), SEM1(CONCRET), SEM2(ANIME), SEMZ(ANIME), SENS(1), SSANIM2(HUM).
- 36 'RAPPORTE!1': UL('C3C'), FLAG(CPRED), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), VAL2(AN), CAT(V), ARFAC(A1,A2), GNR(MAS), NBR(SING), SEM1(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), SEM2(ANIME), SEMZ(ANIME, CONCRET, ACTION, ABSTRAIT, UNITE, LIEU, TEMPS), SENS(3), SSANIM2(HUM).
- 37 'RAPPORTE!1': UL('C2A'), FLAG(CPRED), AUX(AVOIR), SUBV(PPA), VAL0(N), VAL1(N), CAT(V), GNR(MAS), NBR(SING), SEM1(CONCRET), SEMZ(ANIME), SENS(2), SSANIMZ(HUM).
- 38 'VASE': UL('DGN'), FS(OBJ), K(GN), RL(ARG1), SUBN(NC), VALET(N), CAT(N,D), GNR(MAS), NBR(SING), PERS(3).
- 39 'UN': UL('UN'), FS(DES), CAT(D), GNR(MAS), NBR(SING).
- 40 'VASE': UL('VASE/LE'), FS(GOV), SUBN(NC), CAT(N), GNR(MAS), NBR(SING).
- 41 'VASE': UL('-S-'), FLAG(CPRED), SUBN(NC), CAT(N), GNR(MAS), NBR(SING), SEMN(CONCRET), SENS(1), SSANIM(ANIME).
- 42 '*CHINE': UL('GP'), FS(CIRC), K(GN), RS(UNDE), SUBN(NP), VALET(DEN), CAT(N,S), GNR(FEM), NBR(SING), PERS(3).
- 43 'DE': UL('DE'), FS(REG), CAT(S).
- 44 'DE': UL('-S-'), VALET(DEN), VAL1(N), CAT(S).
- 45 'DE': UL('-S-'), VALET(DEI), VAL1(I), CAT(S).
- 46 '*CHINE': UL('*CHINE'), FS(GOV), SUBN(NP), CAT(N), GNR(FEM), NBR(SING).
- 47 '*CHINE': UL('-S-'), FLAG(CPRED), SUBN(NP), CAT(N), GNR(FEM), NBR(SING), SEMN(ANIME, LIEU), SENS(1), SSANIM(HUM, PERS).
- 48 '.': UL('.'), CAT(P), PONC(DR).

Annexe 6

Phase AS6 fvx

Annexe 7

Phase GM fvx-lsp

(18/11/93) - TRADUCTION DU 18 NOVEMBRE 1993 09 H 40 MN 23 S
LIDIA CHINE 18 NOVEMBRE 1993 9H 40MN 35S
LANGUES DE TRAITEMENT: FVX-LSP

-- TEXTE ORIGINE --

Le capitaine a rapporté un vase de Chine.

-- TEXTE TRADUIT --

((nil ""ULTXT"" nil) ((nil ""ULFRA"" nil) ((nil ""ULSOL"" nil) ((nil ""PHVB"" (K phvb CAT v ENONCP decl VOIX act PHASE acc MT ipr SUBV vf PERS 3 LINKS oui R echts oui NBR sing VALET q ARGS (a1 a0))) ((nil ""DGN"" (K gn FS suj RL arg0 CAT (d n) SUBN nc GNR mas PERS 3 NBR sing VALET n)) ("Le" ""LE"" (FS des CAT d GNR mas NBR sing TYPOG pcap))) ("capitaine" ""CAPITAINE"" (FS gov CAT n SUBN nc GNR mas PERS 3 NBR sing SENS (3 2 1)))) ((nil ""NV"" (K phvb CAT v ENONCP decl VOIX act PHASE acc MT ipr SUBV vf PERS 3 LINKS oui R echts oui NBR sing VALET q ARGS (a1 a0))) ("a" ""AVOIR"" (FS aux CAT v MT ipr SUBV vf PERS 3 NBR sing SENS 1))) ("rapporté" ""RAPPORTER"" (FS gov CAT v SUBV ppa GNR mas PERS 3 NBR sing SENS (3 2 1)))) ((nil ""DGN"" (K gn FS obj RL arg1 CAT (d n) SUBN nc GNR mas PERS 3 RECHTS oui NBR sing VALET n)) ("un" ""UN"" (FS des CAT d GNR mas NBR sing))) ("vase" ""VASE/LE"" (FS gov CAT n SUBN nc GNR mas NBR sing SENS 1))) ((nil ""GP"" (K gn FS comp RS unde CAT (s n) SUBN np GNR fem PERS 3 NBR sing VALET den)) ("de" ""DE"" (FS reg CAT s))) ("Chine" ""CHINE"" (FS gov CAT n SUBN np GNR fem NBR sing SENS 1)))) (("." ""."" (CAT p)))) ((nil ""ULSOL"" nil) ((nil ""PHVB"" (K phvb CAT v ENONCP decl VOIX act PHASE acc MT ipr SUBV vf PERS 3 LINKS oui R echts oui NBR sing VALET q ARGS (a1 a0))) ((nil ""DGN"" (K gn FS suj RL arg0 CAT (d n) SUBN nc GNR mas PERS 3 NBR sing VALET n)) ("Le" ""LE"" (FS des CAT d GNR mas NBR sing TYPOG pcap))) ("capitaine" ""CAPITAINE"" (FS gov CAT n SUBN nc GNR mas PERS 3 NBR sing SENS (3 2 1)))) ((nil ""NV"" (K phvb CAT v ENONCP decl VOIX act PHASE acc MT ipr SUBV vf PERS 3 LINKS oui R echts oui NBR sing VALET q ARGS (a1 a0))) ("a" ""AVOIR"" (FS aux CAT v MT ipr SUBV vf PERS 3 NBR sing SENS 1))) ("rapporté" ""RAPPORTER"" (FS gov CAT v SUBV ppa GNR mas PERS 3 NBR sing SENS (3 2 1)))) ((nil ""DGN"" (K gn FS obj RL arg1 CAT (d n) SUBN nc GNR mas PERS 3 NBR sing VALET n)) ("un" ""UN"" (FS des CAT d GNR mas NBR sing))) ("vase" ""VASE/LE"" (FS gov CAT n SUBN nc GNR mas NBR sing SENS 1))) ((nil ""GP"" (K gn FS circ RS unde CAT (s n) SUBN np GNR fem PERS 3 NBR sing VALET den)) ("de" ""DE"" (FS reg CAT s))) ("Chine" ""CHINE"" (FS gov CAT n SUBN np GNR fem NBR sing SENS 1)))) (("." ""."" (CAT p)))))))

Annexe 8

Phase AS7 fvx

17 NOVEMBRE 1993 20H 49MN 51S
*** GROUPE D'ETUDE POUR LA TRADUCTION AUTOMATIQUE ***
* * * * *

AS - Analyse Structurale

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA** **CHINE** **GRAMMAIRE 7** CODE LANGUE : FVX

```

1: 'ULTXT' (2: 'ULFRA' (3: 'ULSOL' (4: 'PHVB' (5: 'DGN' (6: 'LE', 7: 'CAPITAINE' (8:
'CAPITAINE')), 9: 'RAPPORTER' (10: 'RAPPORTER'), 11: 'DGN' (12: 'UN', 13: 'VASE/LE'
(14: 'VASE/LE'), 15: 'GP' (16: 'DE', 17: '*CHINE' (18: '*CHINE'))), 19: '.'))))
1 '': UL('ULTXT').
2 '': UL('ULFRA').
3 '': UL('ULSOL'), MODIF(NON).
4 'A': UL('PHVB'), LINKS(OUI), RECHTS(OUI), ENONCP(DECL), K(PHVB), PHASE(ACC),
SUBV(VF), VALET(Q), VOIX(ACT), ARGS(A0,A1), CAT(V), MT(IPR), NBR(SING),
PERS(3).
5 'CAPITAINE': UL('DGN'), FS(SUJ), K(GN), RL(ARG0), SUBN(NC), VALET(N),
CAT(N,D), GNR(MAS), NBR(SING), PERS(3).
6 '*LE': UL('LE'), FS(DES), CAT(D), GNR(MAS), NBR(SING), TYPOG(PCAP).
7 'CAPITAINE': UL('CAPITAINE'), FS(GOV), SUBN(NC), CAT(N), GNR(MAS),
NBR(SING), PERS(3), SENS(2).
8 'CAPITAINE': UL('CAPITAINE'), SUBN(NC), CAT(N), GNR(MAS),
NBR(SING), PERS(3), SENS(2).
9 'RAPPORTE!1': UL('RAPPORTER'), FS(GOV), SUBV(PPA), CAT(V), GNR(MAS),
NBR(SING), PERS(3), SENS(2).
10 'RAPPORTE!1': UL('RAPPORTER'), SUBV(PPA), CAT(V), GNR(MAS),
NBR(SING), PERS(3), SENS(2).
11 'VASE': UL('DGN'), RECHTS(OUI), FS(OBJ), K(GN), RL(ARG1), SUBN(NC),
VALET(N), CAT(N,D), GNR(MAS), NBR(SING), PERS(3).
12 'UN': UL('UN'), FS(DES), CAT(D), GNR(MAS), NBR(SING).
13 'VASE': UL('VASE/LE'), FS(GOV), SUBN(NC), CAT(N), GNR(MAS),
NBR(SING), SENS(1).
14 'VASE': UL('VASE/LE'), SUBN(NC), CAT(N), GNR(MAS), NBR(SING),
SENS(1).
15 '*CHINE': UL('GP'), FS(COMP), K(GN), RS(UNDE), SUBN(NP), VALET(DEN),
CAT(N,S), GNR(FEM), NBR(SING), PERS(3).
16 'DE': UL('DE'), FS(REG), CAT(S).
17 '*CHINE': UL('*CHINE'), FS(GOV), SUBN(NP), CAT(N), GNR(FEM),
NBR(SING), SENS(1).
18 '*CHINE': UL('*CHINE'), SUBN(NP), CAT(N), GNR(FEM), NBR(SING),
SENS(1).
19 '.': UL('.',), CAT(P), PONC(DR).

```


Annexe 9

Phase TL fvx-sax

17 NOVEMBRE 1993 20H 49MN 57S
*** GROUPE D'ETUDE POUR LA TRADUCTION AUTOMATIQUE ***
* * * * *

TL - Transfert Lexical

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA** **CHINE** CODE LANGUE : FVX-SAX

1: 'ULTXT' (2: 'ULFRA' (3: 'ULSOL' (4: 'PHVB' (5: 'DGN' (6: 'DER', 7: 'GOV' (8: '*KAPITAEN')), 9: 'GOV' (10: 'MITBRINGEN-V'), 11: 'DGN' (12: 'EIN', 13: 'GOV' (14: '*VASE'), 15: 'GP' (16: 'AUS', 17: 'GOV' (18: '*CHINA'))), 19: '.'))))

```

1 '': UL('ULXT').
2 '': UL('ULFRA').
3 '': UL('ULSOL').
4 'A': UL('PHVB'), ENONCP(DECL), K(PHVB), PHASE(ACC), SUBV(VF), VOIX(ACT),
MT(IPR), ARGS(A0,A1), CAT(V), NBR(SING), PERS(3).
5 'CAPITAINE': UL('DGN'), RL(ARG0), K(GN), SUBN(NC), FS(SUJ), GNR(MAS),
CAT(N,D), NBR(SING), PERS(3).
6 '*LE': UL('DER'), FS(DES), GNR(MAS), CAT(D), NBR(SING), TYPOG(PCAP).
7 'CAPITAINE': UL('GOV'), SUBN(NC), FS(GOV), GNR(MAS), CAT(N),
NBR(SING), PERS(3), SENS(2).
8 'CAPITAINE': UL('*KAPITAEN'), SUBN(NC), GENUS(MAS), GNR(MAS),
CAT(N), NBR(SING), PERS(3), SENS(2).
9 'RAPPORTE!1': UL('GOV'), SUBV(PPA), MT(PPA), FS(GOV), GNR(MAS), CAT(V),
NBR(SING), PERS(3), SENS(2).
10 'RAPPORTE!1': UL('MITBRINGEN-V'), SUBV(PPA), MT(PPA), AUXV(HABEN),
CAS1(ACC), GNR(MAS), CAT(V), NBR(SING), PERS(3), SENS(2), VAL0(GN),
VAL1(GN).
11 'VASE': UL('DGN'), RL(ARG1), K(GN), SUBN(NC), GNR(MAS), CAT(N,D),
NBR(SING), PERS(3).
12 'UN': UL('EIN'), FS(DES), GNR(MAS), CAT(D), NBR(SING).
13 'VASE': UL('GOV'), SUBN(NC), FS(GOV), GNR(MAS), CAT(N), NBR(SING),
SENS(1).
14 'VASE': UL('*VASE'), SUBN(NC), GENUS(FEM), GNR(MAS), CAT(N),
NBR(SING), SENS(1).
15 '*CHINE': UL('GP'), RS(UNDE), K(GN), SUBN(NP), GNR(FEM), CAT(N,S),
NBR(SING), PERS(3).
16 'DE': UL('AUS'), FS(REG), CAS1(DAT), CAT(S), VAL1(GN).
17 '*CHINE': UL('GOV'), SUBN(NP), FS(GOV), GNR(FEM), CAT(N),
NBR(SING), SENS(1).
18 '*CHINE': UL('*CHINA'), SUBN(NP), GENUS(NEU), GNR(FEM), CAT(N),
NBR(SING), SEMN(LIEU), SENS(1), POTDRV(NOMADJ).
19 '': UL(''), CAT(P), PONC(DR).

```


Annexe 10

Phase TX fvx-sax

17 NOVEMBRE 1993 20H 50MN 00S
 *** GROUPE D'ETUDE POUR LA TRADUCTION AUTOMATIQUE ***
 * * * * *

TX (Locutions en cible)

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA** **CHINE** CODE LANGUE : FVX-SAX

[illegible]

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA** **CHINE** **GRAMMAIRE 1** CODE LANGUE : FVX-SAX

```

1 '': UL('ULXT').
2 '': UL('ULFRA').
3 '': UL('ULSOL').
4 'A': UL('PHVB'), MT(IPR), ENONCP(DECL), K(PHVB), PHASE(ACC), SUBV(VF),
  VOIX(ACT), ARGS(A0,A1), CAT(V), NBR(SING), PERS(3).
5 'CAPITAINE': UL('DGN'), FS(SUJ), RL(ARG0), K(GN), SUBN(NC), CAT(N,D),
  NBR(SING), PERS(3).
6 '*LE': UL('DER'), FS(DES), CAT(D), NBR(SING), TYPOG(PCAP).
7 'CAPITAINE': UL('GOV'), FS(GOV), SUBN(NC), CAT(N), NBR(SING), PERS(3),
  SENS(2), TYPOG(PCAP).
8 'CAPITAINE': UL('*KAPITAEN'), SUBN(NC), GENUS(MAS), CAT(N),
  NBR(SING), PERS(3), SENS(2), TYPOG(PCAP).
9 'RAPPORTE!1': UL('GOV'), MT(PPA), FS(GOV), SUBV(PPA), CAT(V), NBR(SING),
  PERS(3), SENS(2).
10 'RAPPORTE!1': UL('BRINGEN-V'), MT(PPA), SUBV(PPA), AUXV(HABEN),
  CAS1(ACC), CAT(V), NBR(SING), PERS(3), SENS(2), VAL0(GN), VAL1(GN).
11 'RAPPORTE!1': UL('MIT'), CAT(PSP).
12 'VASE': UL('DGN'), RL(ARG1), K(GN), SUBN(NC), CAT(N,D), NBR(SING),
  PERS(3).
13 'UN': UL('EIN'), FS(DES), CAT(D), NBR(SING).
14 'VASE': UL('GOV'), FS(GOV), SUBN(NC), CAT(N), NBR(SING), SENS(1),
  TYPOG(PCAP).
15 'VASE': UL('*VASE'), SUBN(NC), GENUS(FEM), CAT(N), NBR(SING),
  SENS(1), TYPOG(PCAP).
16 '*CHINE': UL('GP'), RS(UNDE), K(GN), SUBN(NP), CAT(N,S), NBR(SING),
  PERS(3).
17 'DE': UL('AUS'), FS(REG), CAS1(DAT), CAT(S), VAL1(GN).
18 '*CHINE': UL('GOV'), FS(GOV), SUBN(NP), CAT(N), NBR(SING),
  SENS(1), TYPOG(PCAP).
19 '*CHINE': UL('*CHINA'), SUBN(NP), GENUS(NEU), CAT(N),
  NBR(SING), SEMN(LIEU), SENS(1), TYPOG(PCAP), POTDRV(NOMADJ).
20 '': UL(''), CAT(P), PONC(DR).

```


Annexe 12

Phase GS1 sax

17 NOVEMBRE 1993 20H 50MN 25S
*** GROUPE D'ETUDE POUR LA TRADUCTION AUTOMATIQUE ***
*_**_**_**_**_**_**_**_**_*

GS - Génération Syntaxique

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA CHINE** GRAMMAIRE 1 CODE LANGUE : SAX

1: 'ULTXT' (2: 'ULFRA' (3: 'ULSOL' (4: 'PHVB' (5: 'GN' (6: 'DER' (7: '*KAPITAEN' (8: '*KAPITAEN')), 9: 'BRINGEN-V' (10: 'BRINGEN-V' (11: 'MIT')), 12: 'GN' (13: 'EIN' (14: '*VASE' (15: '*VASE') , 16: 'GADJ' (17: '*CHINA' (18: '*CHINA'))), 19: '.'))))

1 '': UL('ULTXT').
2 '': UL('ULFRA').
3 '': UL('ULSOL').
4 'A': UL('PHVB'), MT(IPR), VXLOCK(1), FSLOCK(1), KLOCK(1), ENONCP(DECL), K(PHVB), PHASE(ACC), SUBV(VF), VOIX(ACT), ARGS(A0,A1), CAT(V), NBR(SING), PERS(3).
5 'CAPITAINE': UL('GN'), FSLOCK(1), KLOCK(1), FS(SUJ), RL(ARG0), K(GN), SUBN(NC), CAS(NOM), CAT(N,D), NBR(SING), PERS(3), VALET(GN).
6 'LE': UL('DER'), FS(DES), CAT(D), NBR(SING), TYPOG(PCAP).
7 'CAPITAINE': UL('*KAPITAEN'), CATLOCK(1), FS(GOV), SUBN(NC), GENUS(MAS), CAS(NOM), CAT(N), NBR(SING), PERS(3), SENS(2), TYPOG(PCAP).
8 'CAPITAINE': UL('*KAPITAEN'), SUBN(NC), GENUS(MAS), CAT(N), NBR(SING), PERS(3), SENS(2), TYPOG(PCAP).
9 'RAPPORTE!1': UL('BRINGEN-V'), MT(PPA), CATLOCK(1), FS(GOV), SUBV(PPA), AUXV(HABEN), CAS1(ACC), CAS2(DAT), CAT(V), NBR(SING), PERS(3), SENS(2), VAL0(GN), VAL1(GN), VAL2(GN).
10 'RAPPORTE!1': UL('BRINGEN-V'), MT(PPA), SUBV(PPA), AUXV(HABEN), CAS1(ACC), CAT(V), NBR(SING), PERS(3), SENS(2), VAL0(GN), VAL1(GN).
11 'RAPPORTE!1': UL('MIT'), CAT(PSP).
12 'VASE': UL('GN'), FSLOCK(1), KLOCK(1), FS(OBJ1), RL(ARG1), K(GN), SUBN(NC), CAS(ACC), CAT(N,D), NBR(SING), PERS(3), VALET(GN).
13 'UN': UL('EIN'), FS(DES), CAT(D), NBR(SING).
14 'VASE': UL('*VASE'), CATLOCK(1), FS(GOV), SUBN(NC), GENUS(FEM), CAS(ACC), CAT(N), NBR(SING), SENS(1), TYPOG(PCAP).
15 'VASE': UL('*VASE'), SUBN(NC), GENUS(FEM), CAT(N), NBR(SING), SENS(1), TYPOG(PCAP).
16 '*CHINE': UL('GADJ'), FSLOCK(1), KLOCK(1), DRVLOCK(1), FS(ATG), RS(UNDE), K(GADJ), SUBN(NP), DRV(NOMADJ), CAT(ADJ), PERS(3).
17 '*CHINE': UL('*CHINA'), CATLOCK(1), DRVLOCK(1), FS(GOV), SUBN(NP), DRV(NOMADJ), CAT(ADJ), SEMN(LIEU), SENS(1), POTDRV(NOMADJ).
18 '*CHINE': UL('*CHINA'), SUBN(NP), GENUS(NEU), CAT(N), NBR(SING), SEMN(LIEU), SENS(1), TYPOG(PCAP), POTDRV(NOMADJ).

19 ' ': UL(' '), CAT(P), POND(DR).

Annexe 13

Phase GS2 sax

17 NOVEMBRE 1993 20H 50MN 42S
*** GROUPE D'ETUDE POUR LA TRADUCTION AUTOMATIQUE ***

GS - Génération Syntaxique

RESULTAT DE L'EXECUTION, TEXTE : **LIDIA** **CHINE** **GRAMMAIRE 2** CODE LANGUE : SAX

1: 'ULTXT' (2: 'ULFRA' (3: 'ULSOL' (4: 'PHVB' (5: 'GN' (6: 'DER', 7: '*KAPITAEN'), 8: 'HABEN-V', 9: 'GN' (10: 'EIN', 11: 'GADJ' (12: '*CHINA'), 13: '*VASE'), 14: 'MIT', 15: 'BRINGEN-V', 16: '.')))

```

1  '': UL('LXTXT').
2  '': UL('ULFRA').
3  '': UL('ULSOL').
4  'A': UL('PHVB'), MT(IPR), VXLOCK(1), FSLOCK(1), KLOCK(1), ENONCP(DECL),
    K(PHVB), PHASE(ACC), SUBV(VF), VOIX(ACT), ARGS(A0,A1), CAT(V), NBR(SING),
    PERS(3).
5  'CAPITAINE': UL('GN'), FSLOCK(1), KLOCK(1), PONLOCK(1), FS(SUJ),
    RL(ARG0), K(GN), SUBN(NC), GENUS(MAS), CAS(NOM), CAT(N,D), NBR(SING),
    PERS(3), VALET(GN), POND(-45).
6  '*LE': UL('DER'), PONLOCK(1), FS(DES), GENUS(MAS), CAS(NOM), CAT(D),
    NBR(SING), TYPOG(PCAP), POND(-45).
7  'CAPITAINE': UL('*KAPITAEN'), CATLOCK(1), FS(GOV), SUBN(NC),
    GENUS(MAS), CAS(NOM), CAT(N), NBR(SING), PERS(3), SENS(2), TYPOG(PCAP).
8  '': UL('HABEN-V'), MT(IPR), PONLOCK(1), FS(AUX), CAT(V), NBR(SING),
    PERS(3).
9  'VASE': UL('GN'), FSLOCK(1), KLOCK(1), PONLOCK(1), FS(OBJ1), RL(ARG1),
    K(GN), SUBN(NC), GENUS(FEM), CAS(ACC), CAT(N,D), NBR(SING), PERS(3),
    VALET(GN), POND(+25).
10 'UN': UL('EIN'), PONLOCK(1), FS(DES), GENUS(FEM), CAS(ACC), CAT(D),
    NBR(SING), POND(-45).
11 '*CHINE': UL('GADJ'), FLXA(FORT), FSLOCK(1), KLOCK(1), DRVLOCK(1),
    PONLOCK(1), FS(ATG), RS(UNDE), K(GADJ), SUBN(NP), GENUS(FEM),
    DRV(NOMADJ), CAS(ACC), CAT(ADJ), NBR(SING), PERS(3), POND(-40).
12 '*CHINE': UL('*CHINA'), FLXA(FORT), CATLOCK(1), DRVLOCK(1),
    FS(GOV), SUBN(NP), GENUS(FEM), DRV(NOMADJ), CAS(ACC), CAT(ADJ),
    NBR(SING), SEMN(LIEU), SENS(1), POTDRV(NOMADJ).
13 'VASE': UL('*VASE'), CATLOCK(1), FS(GOV), SUBN(NC), GENUS(FEM),
    CAS(ACC), CAT(N), NBR(SING), SENS(1), TYPOG(PCAP).
14 'RAPPORTE!1': UL('MIT'), COLLER(OUI), PONLOCK(1), CAT(PSP), POND(+39).
15 'RAPPORTE!1': UL('BRINGEN-V'), MT(PPA), CATLOCK(1), PONLOCK(1), FS(GOV),
    SUBV(PPA), AUXV(HABEN), CAS1(ACC), CAS2(DAT), CAT(V), SENS(2), VAL0(GN),
    VAL1(GN), VAL2(GN), POND(+40).

```

16 '.': UL('.'), PONLOCK(1), CAT(P), PONC(DR), POND(+50).

Phase GY sax

GY (Interface qs-qm)

1: 'ULTXT' (2: 'ULFRA' (3: 'ULSOL' (4: 'PHVB' (5: 'GN' (6: 'DER', 7: '*KAPITAEN'), 8: 'HABEN-V', 9: 'GN' (10: 'EIN', 11: 'GADJ' (12: '*CHINA'), 13: '*VASE'), 14: 'MIT', 15: 'BRINGEN-V', 16: '.'))))

315

Annexe 15

Phase GM fvx-sax

(17/11/93) - TRADUCTION DU 17 NOVEMBRE 1993 20 H 51 MN 08 S
LIDIA CHINE 17 NOVEMBRE 1993 20H 51MN 18S
LANGUES DE TRAITEMENT: FVX-SAX

-- TEXTE ORIGINE --

Le capitaine a rapporté un vase de Chine.

-- TEXTE TRADUIT --

Der Kapitaen hat eine chinesische Vase mitgebracht.

Modèles Linguistiques Ariane pour LIDIA-1 en Chiffres

Quelques chiffres sur les modèles linguistiques FVX, LSP et SAX utilisés dans le cadre de LIDIA-1.

DONNÉES 23/11/93	Phases	Fichiers	Lignes	Dict.	Gram.	Règles	Variables	UL	Formats
FVX	4	31	14563	---	---	---	---	---	---
AM	1	10	7906	5	1	89	16	5213	385
AX	1	5	401	2	---	---	24	100	3
AY	1	7	1760	3	---	---	61	214	7
AS	1	9	4496	---	81	177	61	54	7
AS1	---	1	323	---	7	23	---	11	---
AS2	---	1	517	---	8	18	---	27	---
AS3	---	1	1091	---	20	55	---	31	---
AS4	---	1	1088	---	20	56	---	31	---
AS5	---	1	653	---	11	17	---	12	---
AS6	---	1	108	---	3	3	---	20	---
AS7	---	1	155	---	5	5	---	22	---
FVX-LSP	2	7	800	---	---	---	---	---	---
TL	1	4	356	1	---	---	40	3-3	1
TS	1	3	444	---	1	6	40	14	2
LSP	3	29	18138	---	---	---	---	---	---
GS	1	4	611	---	8	10	41	17	6
GS1	---	1	134	---	5	5	---	17	---
GS2	---	1	115	---	3	5	---	17	---
GY	1	5	687	1	---	---	68	9	2
GM	1	15	15748	9	3	321	67	5540	508
GM1	---	1	133	---	---	24	---	---	---
GM2	---	1	648	---	---	279	---	---	---
GM3	---	1	62	---	---	18	---	---	---
FVX-SAX	3	16	2398	---	---	---	---	---	---
TL	1	7	1403	3	---	---	58	137-349	4
TX	1	6	548	2	---	---	57	80	12
TS	1	3	447	---	4	5	56	5	1
SAX	3	20	6332	---	---	---	---	---	---
GS	1	4	1514	---	32	42	58	63	3
GS1	---	1	592	---	7	12	---	28	---
GS2	---	1	578	---	25	30	---	46	---
GY	1	5	395	1	---	---	57	1	2
GM	1	9	3253	5	1	14	24	1523	356

Légende :

FVX :	Données linguistiques pour l'analyse du français.
LSP :	Données linguistiques pour transformer les arbres d'analyse du français en forme LISP.
SAX :	Données linguistiques pour la génération de l'allemand.
FVX-LSP :	Données linguistiques de transfert du français vers la forme LISP.
FVX-SAX :	Données linguistiques de transfert du français vers l'allemand.
AM :	Analyse morphologique
AX :	Dictionnaire d'analyse lemme-UL
AY :	Dictionnaire d'analyse avec description des propriétés des UL
AS :	Analyse structurale
AS1 :	Fichier de grammaire d'analyse n° 1
TL :	Transfert lexical
TX :	Transfert lexical (production des taxèmes en cible)
TS :	Transfert structural
GS :	Génération syntaxique
GS1 :	Fichier de grammaire de génération n° 1
GY :	Interface GS-GM
GM :	Génération morphologique

Résumé

L'objet de cette thèse est la définition et l'étude, au moyen d'une maquette, du concept de Traduction Automatisée Fondée sur le Dialogue pour auteur monolingue.

Nous situons d'abord l'objet de cette étude dans une perspective historique en montrant pourquoi les travaux précédents en TAFD n'ont pas (encore) conduit à des systèmes utilisés en pratique. Nous proposons alors le concept de TAFD pour auteur monolingue qui offre de nouvelles perspectives concrètes et permet de poser des questions scientifiques intéressantes.

Le but est de permettre à un auteur monolingue de documentation technique de traduire les documents qu'il rédige. Dans ce cadre, l'auteur aide le système à traduire dans une ou plusieurs langues cibles, via une standardisation et une clarification interactives effectuées une seule fois quel que soit l'ensemble des langues cibles visées. Afin que la station de rédaction de l'auteur soit un ordinateur personnel de milieu de gamme, le système est distribué. Les traitements linguistiques coûteux (analyse, transfert et génération) — écrits en Ariane-G5 par les linguistes du GETA — sont donc effectués sur un serveur de traduction distribué en mode asynchrone.

La maquette LIDIA-1 permet d'étudier les contraintes informatiques, ergonomiques et linguistiques d'un tel système, et de proposer des solutions. Elle permet de traduire une pile de démonstration qui explique certaines ambiguïtés du français, vers l'allemand, l'anglais et le russe. Pour vérifier les traductions produites, l'auteur peut demander des rétrotraductions. Nous avons utilisé une architecture distribuée dans laquelle trois serveurs communiquent au moyen de messages et de boîtes aux lettres. Le processus de clarification est basé sur un mécanisme de reconnaissance de patrons à chacun desquels est associée une méthode de production du dialogue en langue naturelle.

Mots clefs

TAO (Traduction Automatisée par Ordinateur), Traduction Automatisée Fondée sur le Dialogue, TADF pour auteur monolingue, Désambiguïsation interactive, Production de dialogues de désambiguïsation, Communication homme-machine, Architecture distribuée, Ambiguïtés linguistiques.

Abstract

The aim of this thesis is to define and to study, with the help of a mock-up, the concept of Dialogue-Based Machine Translation for monolingual authors.

First we examine the question from a historical viewpoint. We show why the other attempts in the direction of DBMT have not (yet) led to practical systems. Then, we suggest the concept of DBMT for monolingual authors which offers new concrete prospects and allows us to study interesting scientific questions.

In the scenario we propose, a monolingual French engineer creating technical documentation can translate his documents. In this framework, the author helps the system to translate into one, or several languages via interactive standardisation and clarification. These interactive treatments are made only once, whatever the target language(s). The system is a distributed one. With such an architecture, a low-cost personal computer would be usable for authoring and translating. Thus, the 'heavy' linguistic tasks (analysis, transfer and generation) — written in Ariane-G5 by the linguists of the GETA — are made asynchronously on a distributed translation server.

The LIDIA-1 mock-up allows the study of the computational, ergonomic and linguistic constraints of such a system and the proposal of some solutions. It allows the translation of a demonstration stack which illustrates certain ambiguities concerning French into German, English and Russian. To check the translations produced in each target language, the author can ask for 'reverse translations'. Three distributed servers are involved in the translation process, communicating through messages and mail boxes. The clarification process is organized around a pattern-matcher. A dialogue construction method is associated with each pattern. The methods rely on a set of operators.

Keywords

MT (Machine Translation), Dialogue-Based Machine Translation, DBMT for monolingual author, Interactive disambiguation, Disambiguation dialogue generation, Man-Machine Communication, Distributed architecture, Linguistic ambiguities.