



HAL
open science

Référence aux actions dans des dialogues de commande homme-machine

Fabrice Duermael

► **To cite this version:**

Fabrice Duermael. Référence aux actions dans des dialogues de commande homme-machine. Informatique [cs]. Institut National Polytechnique de Lorraine, 1994. Français. NNT : 1994INPL158N . tel-01751279

HAL Id: tel-01751279

<https://hal.univ-lorraine.fr/tel-01751279>

Submitted on 29 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

94INPL158N

Institut National Polytechnique de Lorraine

UFR S.T.M.I.A.

Département de Formation Doctorale en Informatique
École Doctorale IAE + M

Référence aux actions dans des dialogues de commande homme-machine

THÈSE

présentée et soutenue publiquement le 21 décembre 1994

pour l'obtention du

Doctorat de l'Institut National Polytechnique de Lorraine
(Spécialité Informatique)

par

Fabrice DUERMAEL

Service Commun de la Documentation
INPL
Nancy-Brabois

Composition du jury

Président : Jeanine SOUQUIÈRES

Rapporteurs : Jacques MOESCHLER
Gérard SABAH

Examinateurs : Bertrand GAIFFE
Jean-Marie PIERREL
Françoise SIMONOT



D 136 001423 6

Institut National Polytechnique de Lorraine

UFR S.T.M.I.A.

Département de Formation Doctorale en Informatique
École Doctorale IAE + M

(M) 1994 DUERMAEL, F.

Référence aux actions dans des dialogues de commande homme-machine

THÈSE

présentée et soutenue publiquement le 21 décembre 1994

pour l'obtention du

Doctorat de l'Institut National Polytechnique de Lorraine
(Spécialité Informatique)

par

Fabrice DUERMAEL

Composition du jury

Président : Jeanine SOUQUIÈRES

Rapporteurs : Jacques MOESCHLER
Gérard SABAH

Examineurs : Bertrand GAIFFE
Jean-Marie PIERREL
Françoise SIMONOT

Résumé

Dès que l'on propose une interface langagière pour commander une application informatique, il se pose la question des fonctions à activer en réponse à un énoncé de l'utilisateur. L'identification de ces fonctions constitue le problème de la référence aux actions. Classiquement, l'action (simple ou sous la forme d'un script de fonctions) à effectuer est dérivée directement de l'analyse syntaxico-sémantique, et les modifications que subissent les objets en sont déduites. Dans notre thèse, nous défendons le point de vue que cette identification suppose la prise en compte d'une temporalité inhérente aux énoncés de commande. Cela amène à considérer un énoncé comme la spécification d'un état final à atteindre, à travers les ruptures de propriétés qu'il prédique sur des objets. Dans cette optique, nous proposons un modèle de construction d'objets du discours pour référer aux actions en introduisant des graphes de déroulement et des éléments de contrôle (des promoteurs) associés aux noeuds représentant les phases du déroulement. L'utilité de ces promoteurs est de prendre en compte le contexte dynamique des portions de graphe qu'ils contrôlent, de restructurer ce graphe en fonction des propriétés aspectuelles mises en présence et d'alimenter le processus de référence. Cette approche de la référence aux actions permet de traiter plus simplement un langage qui reprend certaines régularités de construction propres au langage naturel, ce qui a pour conséquence de simplifier le modèle de l'application.

Mots-clés: Dialogue homme-machine, langue naturelle, référence aux actions.

Abstract

When designing an human-computer interface to control an application, the question may be raised of what functions are to be called, in response to the interpretation of a given user's utterance. The identification of these functions corresponds the general problem of the reference to actions. Classically, the action to be performed is derived from the parsing, and the modifications undergone by the objects are deduced. In our thesis, we argue that this identification involves taking into account a temporality inherent in command utterances. That leads to consider an utterance as the specification of a final state to be reached. We then propose a model of building of discourse objets by introducing some so-called development graphs and promotors to control them dynamically. This approach of the reference to actions enables us to process more easily a language that uses construction regularities of building which characterize natural languages. This results in a simplification of the application model.

Keywords: Man-Machine Dialogue, Natural Language, Reference to Actions.

Remerciements

Je tiens tout d'abord à remercier mes rapporteurs, Jacques Moeschler (Professeur à l'Université de Genève), Gérard Sabah (Directeur de Recherche au LIMSI, Orsay) et Jeanine Souquières (Professeur à l'Université de Nancy II) d'avoir accepté de corriger et de juger mon travail. Je suis aussi très content que Françoise Simonot, qui m'enseignait l'informatique à l'ENSEM, fasse partie de mon jury.

Ma reconnaissance va à Jean-Marie Pierrel, qui a dirigé ma thèse et m'a accueilli dans son équipe, ainsi qu'à Bertrand Gaiffe, qui m'a co-encadré avec beaucoup de gentillesse et de disponibilité, tout en faisant preuve d'un esprit critique éclairé.

Merci encore à Anne Reboul, qui a entièrement relu mon mémoire, et à Jacques Moeschler, pour les lectures complémentaires qu'ils m'ont conseillées, ainsi que pour les diverses corrections, tant de forme que de fond, qu'ils m'ont aimablement suggérées.

Merci aussi à mes collègues Laurent Romary, Yannick Toussaint, Noëlle Carbonell, Denis Finck, Arnould Matthieu et Emmanuel Bouyer pour les remarques qu'ils ont gracieusement effectuées sur mon mémoire et qui ont contribué à améliorer sa lisibilité. Un bravo au passage à Denis Roegel, qui s'est investi dans le développement du style \LaTeX 2e avec lequel j'ai écrit ce mémoire.

Merci plus généralement à tous mes amis du laboratoire, qui font que le CRIN n'est pas simplement pour moi un lieu de travail, mais une communauté au sens fort.

Enfin, merci à mes parents pour le soutien et les encouragements qu'ils m'ont toujours apportés.

À mes parents.

Table des matières

Résumé 3

Abstract 3

Table des figures

Introduction

I La référence aux actions : présentation et analyse du problème

1

Le cadre spécifique du dialogue de commande

- 1 La finalité de la communication : commander une application 25
 - 1.1 Une finalité induite par la tâche de l'utilisateur 26
 - 1.1.1 La notion sous-jacente d'application 26
 - 1.1.2 La notion sous-jacente d'environnement applicatif 26
 - 1.1.3 Récapitulatif 28
 - 1.2 La classe des applications de type commande 28
 - 1.2.1 Le but de l'utilisateur 28
 - 1.2.2 La visualisation de l'environnement 28
 - 1.2.3 La nécessité d'instaurer un dialogue 31

	1.2.4	Un schéma général d'interaction	33
	1.2.5	Le choix de la communication : l'acte de langage directif	33
2		L'intérêt du langage naturel dans une interface	34
	2.1	L'adaptation à l'utilisateur	34
	2.1.1	Une compétence déjà acquise	35
	2.1.2	Une bonne efficacité d'emploi	35
	2.2	Les facteurs limitatifs de l'adaptation à l'utilisateur	35
	2.2.1	Une taille de lexique restreinte	35
	2.2.2	Un fonctionnement restreint de la langue	36
	2.2.3	La limitation par les connaissances	36
	2.2.4	Inférences : les limitations des techniques d'IA	37
	2.3	Entre langages artificiels et langues naturelles	37
	2.3.1	Les langages artificiels	37
	2.3.2	Vers des langages à consonance naturelle	40
3		Les fonctionnalités de l'interface	42
	3.1	Gérer la tâche	42
	3.1.1	Un mécanisme de raisonnement	42
	3.1.2	Modes et rôles de coopération	43
	3.2	Reconnaître les messages	44
	3.2.1	La reconnaissance vocale	44
	3.2.2	La reconnaissance gestuelle	45
	3.2.3	La question de l'articulation entre modalités	46
	3.3	Interpréter l'énoncé en termes d'actions	47
	3.3.1	Sur quels objets porte l'énoncé?	47
	3.3.2	Quelles actions appliquer?	48
	3.4	Dialoguer avec l'utilisateur	50
	3.4.1	Gestion du canal de communication	50
	3.4.2	Communication motivée par des difficultés de compréhension	51
	3.4.3	Communication motivée par la tâche à réaliser	52
4		Conclusion : formulation du problème	52

2

L'identification des actions dans les systèmes classiques de dialogue

1	Le modèle de l'application sous-jacent	56
---	--	----

1.1	La nécessité de recourir à des préconditions d'utilisation	56
1.2	Représentation de l'état de l'application	56
1.2.1	Une observation directe	57
1.2.2	Une simulation de l'application	57
1.3	Le modèle de l'application : définitions de base	58
2	Description du modèle « classique » de référence aux actions	59
2.1	Présentation générale	59
2.1.1	La sélection de l'action	60
2.1.2	Bilan	61
2.2	L'analyse structurale de l'énoncé	62
2.2.1	Les données en jeu	62
2.2.2	L'analyse syntaxique standard	65
2.2.3	Les grammaires transformationnelles	66
2.2.4	Les grammaires sémantiques	67
2.2.5	Les grammaires à base de caractéristiques	68
2.2.6	Grammaires lexicales fonctionnelles	68
2.3	La structure prédicative de l'énoncé	69
2.3.1	La notion de valence	69
2.3.2	Problèmes sous-jacents de classification des compléments .	70
2.3.3	Prédication et relations casuelles	73
2.4	La mise en correspondance avec une fonction	76
2.4.1	L'arbitraire de l'en-tête de fonction	76
2.4.2	La complétude des arguments dans l'en-tête de fonction .	77
2.4.3	Quelques procédés représentatifs d'association structure d'énoncé-fonction	79
2.5	Récapitulatif	84
3	Les limites concernant l'adjonction de naturel	85
3.1	Un objectif à atteindre : une certaine régularité langagière	85
3.2	Une solution (trop) complexe	86
3.2.1	L'incorporation d'opérateurs complexes	86
3.2.2	Conséquences sur la prise en compte de la régularité des groupes nominaux	88
3.2.3	Conséquences sur la prise en compte de la régularité des groupes prépositionnels	89

4 Conclusion 90

3

Une approche exploitant la dynamique temporelle de l'énoncé

1 L'association « énoncé-but » 92

1.1 La prise en compte des effets des fonctions 92

1.1.1 La dissimulation des postconditions dans les approches classiques 92

1.1.2 Une approche alternative de la référence aux actions 94

1.2 Une approche adaptée à une régularité langagière 95

1.3 La validité du point de vue psychologique 96

1.4 Un partage d'activité envisageable 97

2 Les moyens à développer 98

2.1 Le recours à un planificateur 98

2.1.1 Un exemple de décomposition d'action 98

2.1.2 Bref état de l'art dans le domaine de la planification 99

2.1.3 Un exemple de système utilisant une planification intégrée 101

2.1.4 Conclusion : une mise en parallèle de deux temporalités 103

2.2 Une analyse linguistique en termes de but à atteindre 104

2.2.1 Un exemple simple d'analyse 104

2.2.2 La dynamique temporelle de l'énoncé 105

3 Conclusion 105

II De l'énoncé au but qu'il décrit

4

Représentation des ruptures de propriétés véhiculées par l'énoncé

1 La spécificité de l'acte de langage directif 109

1.1 Les conditions associées aux actes de langage directifs 110

1.1.1	Conditions de réussite	110
1.1.2	Conditions de satisfaction	111
1.2	La forme impérative et ses conditions d'emploi	112
2	Rupture et évolutivité de propriétés d'objets du discours	113
2.1	Les différents types de propriétés	113
2.2	La nécessité d'introduire un modèle temporel	115
2.2.1	Le temps sur un ensemble d'instants	116
2.2.2	Le temps relativement à un intervalle	117
2.2.3	Le modèle minimal temporel adapté	119
2.3	Importance de l'autonomie référentielle	121
2.4	Interaction entre référence aux objets et aux actions	122
2.4.1	Schémas de références	123
2.4.2	La participation des prédicats	125
3	Conclusion	126

5

Analyse linguistique de l'énoncé en termes de ruptures de propriétés

1	Les marques linguistiques de la dynamique temporelle de l'énoncé	127
1.1	Les classes aspectuelles de verbes	128
1.1.1	La catégorie de l'aspect	128
1.1.2	Les classes sémantiques de verbe	129
1.2	Le participe passé comme expression d'un résultat	132
1.2.1	Comparaison entre participe passé et adjectif	132
1.2.2	Le passif: tournures courtes et longues	134
1.2.3	L'acquis du parfait	135
1.2.4	Caractérisation de l'état résultant en termes événementiels	135
1.3	Conception sémantique de la transitivité	136
2	Les structures sous-jacentes de l'énoncé	138
2.1	La dynamique de l'aspect verbal selon Guillaume	139
2.2	Le verbe comme élément structurant de l'énoncé	140
2.2.1	Le verbe comme structure d'accueil	140
2.2.2	L'élément de fonction sujet	142
2.2.3	L'élément de fonction complément d'objet	143
2.2.4	Les éléments à caractère adverbial	145
2.3	Fonctions grammaticales et structures de dépendance	153

2.3.1	Rappels sur la notion de fonction syntaxique	153
2.3.2	Une première approche : la primauté aux emboîtements de syntagmes	154
2.3.3	Une seconde approche : la primauté aux fonctions	157
3	Conclusion	161

III Proposition d'un modèle et validation

6

Un modèle de construction d'objets du discours pour référer aux actions
--

1	L'élaboration de schèmes de déroulement	166
1.1	La structure des schèmes de déroulement	166
1.2	Identification des objets et degré d'autonomie référentielle	167
1.2.1	La hiérarchie des degrés d'autonomie	167
1.2.2	La définition de l'objet	167
1.3	Les phases du déroulement	169
1.3.1	Le point de départ	169
1.3.2	La contribution du verbe seul	169
1.3.3	La contribution de l'objet	171
1.3.4	La contribution des prépositions	174
1.4	La modélisation de l'interaction contextuelle: l' intervention des promoteurs	181
1.4.1	Les différentes classes de promoteurs	181
1.4.2	L'interaction contextuelle entre les nœuds du graphe de déroulement	182
1.4.3	Éléments de formalisation	185
2	Les promoteurs comme pivots de la construction de zones temporelles . . .	187
2.1	De la dépendance structurelle à la dépendance sémantique	188
2.1.1	Critère d'inclusion-dépendance	189

Table des matières

2.1.2	Le problème de la compositionnalité	190
2.1.3	La question de la dépendance sémantique	191
2.2	L'influence des spécifications sémantiques	191
2.2.1	L'influence des domaines sémantiques	192
2.2.2	Le cadre sémantique du verbe	193
2.3	Formalisation des règles de construction de zones temporelles	195
2.3.1	Gabarit associé au lien <i>inchoatif</i>	195
2.3.2	Gabarit associé au lien <i>terminatif</i>	196
2.3.3	Gabarits associés aux liens de localisation	196
2.3.4	Gabarit associé au lien « jusqu'à »	197
3	Conclusion	198

7

Choix de mise en œuvre

1	Implantation des principales composantes	202
1.1	Les principes architecturaux relatifs à notre démarche	202
1.2	Analyse structurale de l'énoncé	203
1.2.1	Une analyse tabulaire	203
1.2.2	Un mécanisme d'unification adapté aux grammaires de dépendance	203
1.3	Schémas de déroulement	205
1.3.1	La représentation des nœuds	205
1.3.2	Algorithme	206
1.3.3	L'interaction entre les diverses composantes	209
1.4	Processus inférentiels	210
1.4.1	Inférences liées à la sémantique lexicale	210
1.4.2	Inférences d'ordre pragmatique	212
1.4.3	La prise en compte d'un modèle de l'application	212
1.5	Résolution de problèmes	213
2	Tests et résultats	214
2.1	Choix du cadre applicatif	214
2.2	Un exemple détaillé complet	216
3	Conclusion	221

Conclusion

1	Bilan	223
2	Pour aller plus loin	223
	2.1 Une meilleure prise en compte du déroulement des processus	223
	2.2 Prise en compte d'autres actes de langages	224
3	Perspectives à long terme	225

Bibliographie

Table des figures

1.1	Interaction traditionnelle	30
1.2	Manipulation virtuelle d'objets	30
1.3	Communication homme-machine avec environnement physique réel	33
1.4	Approche « morpho-syntaxique » (1) de la multimodalité	46
1.5	Approche « sémantico-pragmatique » (2) de la multimodalité	46
2.1	Approche 1 : observation de l'application	57
2.2	Approche 2 : simulation de l'application	58
2.3	Modèle statique de l'application	59
2.4	Ecran où s'effectue la désignation du fichier à imprimer	60
2.5	Bijection classique	62
2.6	Flux général de données dans les approches classiques de référence aux actions	63
2.7	Une approche simpliste de la constitution d'un énoncé	64
2.8	Une approche arborescente de la constitution d'un énoncé	65
2.9	Exemple de grammaire RTN. Terminaux : VERBE, DET, ADJ, NOM, PREP. Non-terminaux : SN, SP, S.	67
2.10	Première approche de la valence	70
2.11	Approche complémentaire de la valence	70
2.12	Procédé de calcul de la référence aux actions dans SHRDLU	82
2.13	Une métaphore de l'orthogonalité de la syntaxe	86
2.14	Modèle structuré de l'application	87
2.15	Bijection classique (2)	88
3.1	Comparatif de deux approches de référence aux actions	94
3.2	Partage d'activité entre utilisateur et système	97
3.3	Structure d'un planificateur	100
3.4	La mise en parallèle de deux temporalités	103
4.1	Propriétés nécessaires et accidentelles	115
4.2	Les points de vue de localisation	116
4.3	L'arbre des états de Mc Dermott	117
4.4	Un univers indécis pour le prédicat : $TRUE(t1,t2,COLOR(HOUSE17,RED))$	117
4.5	Les 13 relations possibles d'Allen entre deux intervalles	118
4.6	Conditions de vérité du verbe abstrait BECOME	119

4.7	L'inclusion temporelle	119
4.8	La précédence temporelle	120
4.9	Représentation d'une rupture temporelle	121
4.10	Schème de référence pour <i>le N</i>	124
4.11	Schème de référence pour <i>un N</i>	124
4.12	Schème de référence pour <i>ce N</i>	124
4.13	Schème de référence pour <i>Dessine un cercle rouge</i>	125
4.14	Schème de référence pour <i>Un homme entra</i>	125
5.1	Tests d'Aktionsart d'après Dowty	130
5.2	Représentation d'une promenade	131
5.3	Structure sémantique de <i>x reddens</i> (x rougit) d'après McCawley	132
5.4	Structure sémantique de <i>You close the door</i> d'après McCawley	133
5.5	Compositionnalité de l'aspect chez Verkuyl (1)	137
5.6	Compositionnalité de l'aspect chez Verkuyl (2)	137
5.7	L'aspect verbal selon Guillaume	139
5.8	La transitivité selon Pottier	143
5.9	La sémantèse du verbe transitif selon Moignet	143
5.10	Le cadre conceptuel selon Pottier	146
5.11	Classement aspectuel des prépositions selon Pottier	146
5.12	Classification des verbes de mouvement d'après Laur	148
5.13	Classification des prépositions spatiales d'après Laur	149
5.14	Dynamique des verbes de transaction selon Langacker	150
5.15	L'opposition directionnelle entre <i>à</i> et <i>de</i> selon Guillaume	151
5.16	Le point de vue du locuteur	152
5.17	Arbre syntagmatique standard de <i>Il écrit une lettre en deux exemplaires</i>	155
5.18	Structure fonctionnelle	156
5.19	Un exemple simple d'énoncé illustrant les connexion, translation, jonction	158
5.20	Le même énoncé avec le modèle de Mel'Cuk	160
6.1	Ensemble d'objets borné	168
6.2	Ensemble d'objet non borné	168
6.3	Processus d'amorçage	169
6.4	La rupture temporelle du verbe	170
6.5	La dualité entre <i>potentiel d'accomplissement</i> et <i>déjà accompli</i>	170
6.6	Graphe d'un verbe intransitif	171
6.7	La rupture temporelle du verbe transitif sans objet	171
6.8	La rupture temporelle avec objet direct	172
6.9	Graphe d'un déroulement avec verbe transitif	172
6.10	Graphe réduit d'un déroulement avec verbe transitif et objet non borné	173
6.11	Graphe d'un déroulement avec verbe transitif et objet non borné	173
6.12	La dualité entre <i>potentiel d'accomplissement</i> et <i>déjà accompli</i>	174
6.13	La rupture induite par « jusqu'à »	175
6.14	L'intégration de <i>en</i> (a)	176
6.15	L'intégration de <i>en</i> (b)	176

Table des figures

6.16	L'intégration de <i>pendant</i>	177
6.17	L'intégration de <i>à</i>	178
6.18	L'intégration de <i>à</i>	178
6.19	L'intégration de <i>et</i>	179
6.20	L'intégration de <i>de</i>	179
6.21	L'intégration de <i>dans</i>	180
6.22	La partie commune initiale du fonctionnement des promoteurs	182
6.23	Réaction d'un promoteur face à un objet non borné	183
6.24	Réaction d'un promoteur face à un objet borné	184
6.25	Les schèmes de déroulement comme représentation intermédiaire	188
6.26	Critère d'inclusion-dépendance	189
6.27	Une bonne représentation temporelle d' <i>imprimer avec entête</i>	190
6.28	Une mauvaise représentation temporelle d' <i>imprimer avec entête</i>	190
6.29	Les quatre types de combinaisons de propriétés de C par rapport à B	192
6.30	Critères sémantiques des verbes collectifs	194
6.31	Ruptures temporelles avec verbes collectifs	194
6.32	Gabarit de zones temporelles rattaché à <i>inchoatif</i>	196
6.33	Gabarit de zones temporelles rattaché à <i>terminatif</i>	196
6.34	Gabarit de zones temporelles rattaché aux prépositions de localisation	197
6.35	Gabarit de zones temporelles rattaché à <i>jusqu'à</i>	198
6.36	Zones temporelles associées à <i>Marche jusqu'à cette balise</i>	198
7.1	Organigramme général	202
7.2	Comparatif entre représentation arborescente (syntagmatique) et tabulaire	204
7.3	Le sujet implicite dans l'impératif	206
7.4	Administration d'un réseau hétérogène	216
7.5	Table d'analyse (1)	216
7.6	Table d'analyse (2)	217
7.7	Structure de dépendance (1)	217
7.8	Structure de dépendance (2)	217
7.9	Schème de déroulement auquel on aboutit	218
7.10	Représentation sous forme de zones temporelles (1)	219
7.11	Représentation sous forme de zones temporelles (2)	220

Introduction

Dès que l'on envisage une communication entre un homme et une machine, aussi élaboré que puisse en être le mode, la portée d'une telle communication se trouve fortement restreinte par rapport à une communication naturelle entre individus. Il est important de mettre en évidence les spécificités de la communication homme-machine par rapport aux communications entre individus. Dans ces dernières, les motivations sont d'une grande diversité potentielle, souvent entremêlées les unes aux autres, sans que l'on puisse toutes les appréhender. Il peut s'agir par exemple de faire rire son entourage, de chercher à le séduire, ou encore de le tromper. Considérée dans toute sa généralité, on peut ainsi dire que la communication revêt une dimension sociale. Il est par exemple révélateur que dans nos conversations de tous les jours, la communication puisse être une fin en soi, ce qui revient simplement au fait de parler pour parler. D'une manière générale, on peut affirmer qu'il existe dans la communication naturelle diverses finalités à caractère social plus ou moins appréhendables.

En revanche, sauf à prêter à un ordinateur une personnalité humaine, on ne saurait le considérer comme un compagnon de conversation. En règle générale, le dialogue entre un individu et un ordinateur, simulé ou non, est plus économique et stéréotypé qu'avec une autre personne. Même si l'on n'impose pas de contraintes, il apparaît que l'individu restreint de lui-même son langage, comme pour s'adapter aux mécanismes de raisonnement qu'il attribue à l'ordinateur. Ainsi, au cours d'expériences menées par la SNCF pour étudier les caractéristiques des dialogues en langue naturelle entre une personne et un système informatique, on a pu constater que les justifications personnelles n'intervenaient pas, alors qu'il serait tout à fait concevable de dire à un guichetier :

(0.1) Je dois aller voir ma belle-mère qui est malade.

De ce fait, les seules motivations que l'on sache traiter en pratique, se limitent à la satisfaction directe d'un service ou d'un besoin. La communication se voit ainsi réduite à son aspect utilitaire et nous désignerons alors le protagoniste humain de la conversation par le terme *utilisateur*. Une conception purement utilitaire de la communication a une conséquence majeure au niveau du déroulement du dialogue entre les deux protagonistes : ce dernier se trouve finalisé en termes exclusifs de besoins. Une telle finalité trouve d'ailleurs son explication en psychologie cognitive. Pour analyser une activité quelle qu'elle soit, les psychologues montrent qu'il est fondamental de faire référence à la tâche qui est effectuée, caractérisée par un but à atteindre ([Hoc, 1988], p.16 ; [Hoc, 1980]).

Nous nous placerons dans le cadre d'une application donnée et, notre centre d'intérêt consistant à identifier une action, nous nous intéresserons aux applications de type

commande. Dans ces applications, l'utilisateur doit faire évoluer l'état de l'application de manière à ce qu'il y ait convergence avec son but propre. On peut alors envisager une manipulation directe, où l'utilisateur manipule lui-même les objets de l'application, ou une manipulation indirecte, où il passe par un intermédiaire : l'ordinateur. Dans ce dernier cas, l'utilisateur donne des ordres à l'ordinateur via une communication verbale. C'est ce cas qui nous intéresse et que nous entendons traiter, en partie du moins, dans notre thèse. Dans un tel cadre, le problème de la référence aux actions est d'identifier, dans les énoncés de l'utilisateur, la séquence d'opérations que l'application doit exécuter.

La première partie de notre thèse visera à présenter le problème et les solutions qui ont été envisagées pour le résoudre. Dans le premier chapitre, nous délimiterons le cadre dans lequel nous souhaitons apporter une solution. Notamment, nous donnerons des arguments en faveur du bien-fondé des délimitations de ce cadre en nous reposant sur les avantages sous-jacents aux langages à consonance naturelle. Dans le deuxième chapitre, nous effectuerons une étude critique des approches classiquement employées pour référer aux actions, consistant à associer une classe d'énoncés à une fonction de l'application. Nous mettrons en évidence les limites auxquelles elles se heurtent. Enfin, dans le troisième et dernier chapitre de cette partie, nous examinerons comment les méthodes traditionnelles dissimulent la dynamique temporelle de l'énoncé. Nous posons ainsi notre thèse selon laquelle une prise en compte de cette dynamique temporelle lors de l'analyse de l'énoncé améliore les systèmes de dialogues de commande.

Dans la deuxième partie, nous jalonnons les étapes d'une approche qui nous semble mieux adaptée pour résoudre le problème, en présentant diverses théories allant dans le sens d'un but à atteindre à partir des spécifications de l'énoncé. Le quatrième chapitre sera consacré à la définition des caractéristiques minimales que doit posséder un modèle temporel pour représenter les évolutions de propriétés que véhiculent les actes de langage directifs. Dans le cinquième chapitre, nous mettrons en évidence la façon dont on peut utiliser les marques linguistiques de l'énoncé pour spécifier une dynamique d'évolution sur les propriétés des objets de l'application.

Dans la troisième partie, nous serons alors en mesure de proposer un modèle qui en rende compte, ce qui fera l'objet du sixième chapitre. Nous présenterons alors dans le septième et dernier chapitre les choix que nous avons effectués pour mettre en œuvre un prototype avec ses tests de validation.

Première partie

La référence aux actions : présentation et analyse du problème

Chapitre 1

Le cadre spécifique du dialogue de commande

Sommaire

1	La finalité de la communication : commander une application	25
1.1	Une finalité induite par la tâche de l'utilisateur	26
1.2	La classe des applications de type commande	28
2	L'intérêt du langage naturel dans une interface	34
2.1	L'adaptation à l'utilisateur	34
2.2	Les facteurs limitatifs de l'adaptation à l'utilisateur	35
2.3	Entre langages artificiels et langues naturelles	37
3	Les fonctionnalités de l'interface	42
3.1	Gérer la tâche	42
3.2	Reconnaître les messages	44
3.3	Interpréter l'énoncé en termes d'actions	47
3.4	Dialoguer avec l'utilisateur	50
4	Conclusion : formulation du problème	52

1 La finalité de la communication : commander une application

D'une manière générale, on peut caractériser le rôle d'un ordinateur comme étant de satisfaire un utilisateur par l'intermédiaire de services qu'il est susceptible de lui rendre. Les services en question sont d'une grande diversité potentielle, correspondant chacun à des domaines de compétence particuliers. On peut d'emblée citer le traitement de texte et le tableur, mais encore des applications aussi hétéroclites que le renseignements administratifs dans une mairie, la réservation de billets de train, la conduite d'un robot dans

une centrale nucléaire, le contrôle de trafic aérien, . . . Du point de vue de l'utilisateur, chacun des services proposés correspondra à une tâche particulière à accomplir.

1.1 Une finalité induite par la tâche de l'utilisateur

Pour analyser une activité quelle qu'elle soit, les psychologues montrent qu'il est fondamental de faire référence à la tâche qui est effectuée, définie par un but à atteindre moyennant diverses contraintes ([Hoc, 1988, p. 16]; [Hoc, 1980; Leplat et Hoc, 1983]). Du point de vue où nous nous plaçons, cela correspond à une conception très générale de la tâche, le but consistant en la satisfaction d'un service donné. Dans le cas d'une partie d'échecs, il peut aussi bien s'agir pour l'utilisateur de remporter la partie que de jouer. La réalisation d'une tâche suppose que l'on fasse évoluer un état. Si tel n'est pas le cas, cela signifie que le but projeté est déjà réalisé. La question qui se pose alors est de pouvoir recenser les moyens qui sont offerts pour permettre à cet état d'évoluer, conformément au but que s'est fixé l'utilisateur. C'est à ce titre qu'intervient la notion d'application.

1.1.1 La notion sous-jacente d'application

Pour prendre un exemple concret emprunté à la vie quotidienne, imaginons une tâche consistant à mettre une cheville dans un mur. L'accomplissement de la tâche ne sera pas le même, selon que notre bricoleur dispose d'un vilebrequin ou d'une perceuse électrique. Transposé sur le plan informatique, c'est exactement au même titre qu'intervient la notion d'*application*: il s'agit du programme qui fournit les moyens logiciels pour réaliser une tâche. On peut donc concevoir l'application comme un ensemble de compétences logicielles destinées à satisfaire des services. *La notion de tâche, du point de vue de l'informaticien, ne pourra se définir qu'à l'intérieur du cadre d'une application.* Directement lié à l'application se trouve l'environnement auquel elle accède.

1.1.2 La notion sous-jacente d'environnement applicatif

Une tâche peut être considérée effectuée lorsque les conditions pour atteindre le but qui la caractérise ont été remplies. Puisqu'un but n'est rien d'autre que la projection temporelle d'un état, cela revient à caractériser un état à atteindre dans un monde de référence [Falzon, 1984, p. 5]. Nous appellerons ce monde *l'environnement applicatif* étant donné qu'il est directement accessible par l'application.

La diversité des services que peut rendre la machine fait que les objets informatiques composant l'environnement sont de nature très variable. Ce peut être aussi bien des informations que des objets plus tangibles. A priori, chaque protagoniste de l'interaction dispose de ses connaissances propres sur les objets composant l'environnement applicatif. On peut toutefois se placer dans deux hypothèses. Dans un cas, on peut supposer que les protagonistes disposent des mêmes connaissances sur les objets de l'environnement, ce qui revient à se placer dans le cadre d'une objectivité totale. Dans l'autre cas, on cherchera au contraire à exploiter la subjectivité des connaissances, par rapport à tel ou tel protagoniste.

Supposer une objectivité totale. Dans cette première hypothèse, l'environnement applicatif est complètement dissocié des protagonistes de l'interaction et il devient l'objet-même de l'interaction. On dispose alors d'un monde composé d'objets physiques sur lequel chaque protagoniste peut agir directement ou indirectement par l'intermédiaire de commandes. Les applications types correspondantes sont la commande de processus industriels, la manipulation d'environnements de multifenêtrage, la commande de systèmes d'exploitation tels que MSDOS ou UNIX. De telles conditions sont très propices à l'instauration d'un cadre de référence à l'action. Aussi est-ce l'hypothèse que nous avons retenue pour fixer notre cadre d'étude. Nous développerons les caractéristiques de cette classe d'applications dans la section 3.3.2.

Exploiter la subjectivité des connaissances. Dans cette seconde hypothèse, les connaissances impliquent nécessairement un sujet auxquelles elles sont relatives, en l'occurrence un protagoniste de la communication : l'homme (protagoniste naturel) ou la machine (protagoniste artificiel). Cela fait donc intervenir au niveau de la tâche un but supplémentaire qui pourra s'avérer plus ou moins prépondérant selon la spécificité de l'application : augmenter la part d'objectivité des connaissances par rapport à tel ou tel protagoniste. Par contraste avec les objets physiques, les connaissances peuvent être instanciées. Nous distinguerons donc deux niveaux, selon que l'on travaille sur les connaissances elles-mêmes ou sur des instanciations de ces connaissances.

Au niveau même des connaissances. L'actualisation de connaissances par l'acquisition de nouvelles connaissances constitue ce que l'on appelle l'apprentissage.

- Si c'est la machine qui acquiert de nouvelles connaissances, la tâche de l'utilisateur est alors d'*inculquer des connaissances*.
- Sinon, le résultat escompté par l'utilisateur est d'acquérir de nouvelles connaissances. Les applications correspondantes sont celles de l'enseignement intelligent assisté par ordinateur (EIAO).

Le modèle associé est dans ce cas un méta-modèle de connaissances [Pitrat, 1990].

Au niveau instanciation de connaissances. Une instanciation de connaissances peut s'appeler classiquement une *information* ou un *renseignement*. Nous emploierons ici indifféremment le terme de *donnée*.

- Si c'est la machine qui acquiert de nouvelles données, la tâche de l'utilisateur est alors de *fournir des données*. Les applications types correspondantes sont la réservation automatique de billets, la saisie orale de données, la dictée de texte avec reconnaissance automatique.
- Si c'est la machine qui fournit des données, la tâche de l'utilisateur est alors de *demande des renseignements*. Les conditions d'atteinte de ce but pour l'utilisateur sont donc la satisfaction complète de sa ou ses requêtes.

Dans les deux cas, le modèle associé sera un modèle de structures de données, indiquant comment elles sont organisées les unes par rapport aux autres (modèle relationnel).

1.1.3 Récapitulatif

Selon l'hypothèse que l'on effectue au niveau de la nature des objets composant l'environnement applicatif et la visibilité qu'en a chaque protagoniste de l'interaction, on y distingue :

- α Les applications de type commande,
avec un environnement physique complètement objectif (réel, virtuel ou mixte)
- β Les autres applications,
avec un environnement cognitif subjectif (demande de renseignements, saisie, EIAO, apprentissage à une machine)¹

Nous pourrions en outre mentionner des applications mixtes qui incorporent une partie de α et de β . Un bon exemple est celui de la commande et interrogation à distance de dispositifs réels. Cela peut avoir une application dans le cadre d'une maison robotisée lorsque ses habitants n'y vivent pas en permanence. Ce peut être pour allumer le chauffage, simuler une présence lors d'un départ en vacances, déclencher un arrosage.

Dans ce qui suit, nous nous restreindrons à une classe particulière d'applications qui est propice à la formulation de notre problème : celle des applications de type commande.

1.2 La classe des applications de type commande

1.2.1 Le but de l'utilisateur

La tâche de l'utilisateur consiste à *effectuer des modifications sur un environnement composé d'objets physiques*, afin de le faire évoluer. Le modèle de l'application sous-jacent sera donc un modèle de l'action physique. Au niveau des objets, cela correspondra à des actions de création, de modification de propriétés ou de relations, et de destruction. Ce qui concourt à la spécificité des applications de type commande est de supposer une objectivité des connaissances que chaque protagoniste a des objets de l'environnement. Nous allons voir quelles sont les conséquences de cette hypothèse au niveau des caractéristiques de l'interaction entre l'utilisateur et ce type d'application.

1.2.2 La visualisation de l'environnement

Pour que l'on puisse parler de connaissances objectives sur les objets composant l'environnement, il est important de faire partager à l'utilisateur et à l'application toutes les caractéristiques pouvant être mises en jeu dans l'accomplissement de la tâche. Il faut donc pouvoir à la fois assurer une perception visuelle (en ce qui concerne l'utilisateur) et artificielle (en ce qui concerne l'ordinateur) de l'environnement. On peut différencier plusieurs cas, selon que l'on fasse appel au niveau du dispositif informatique à une perception artificielle ou non.

1. Notons également la nécessité qu'il y ait une part d'objectivité des connaissances quant aux objets composant l'environnement applicatif, sans quoi il n'y aurait pas de communication envisageable. La tâche générale de β relève même de l'exploitation de la part de subjectivité inhérente à chaque protagoniste afin de la réduire, de la rendre objective. Ainsi α apparaît comme un cas extrême de β .

Le recours à la perception artificielle. La robotique intervient ici pleinement au niveau de l'interaction entre la machine et l'environnement physique, mettant en jeu des actionneurs et des capteurs pour, respectivement, agir sur l'environnement et en avoir une perception. On peut citer comme exemple de base le cas où l'ordinateur et son application sont incorporés dans un robot [Klein, 1990]. Cela fait typiquement intervenir une caméra CCD² comme capteur, permettant ainsi au robot de percevoir l'environnement dans lequel il évolue. Or, les ordres provenant de l'utilisateur qui commande le robot feront eux aussi référence à cet environnement. Afin de pouvoir effectuer le rapprochement nécessaire, le robot devra effectuer deux interprétations :

- l'une par rapport à l'environnement, en effectuant une analyse de la scène perçue ;
- l'autre par rapport à l'utilisateur, en comprenant les commandes qu'il reçoit.

Ces deux interprétations sont complémentaires, l'une guidant l'autre.

L'intérêt d'intégrer un système de vision à un robot est de pouvoir lui laisser une plus grande autonomie. Cela est notamment indispensable quand l'univers dans lequel il évolue est inaccessible à l'utilisateur. Citons comme exemple l'enceinte d'une centrale nucléaire où des robots commandés à distance peuvent intervenir dans des zones contaminées. Divers moyens peuvent alors permettre à l'utilisateur de percevoir l'environnement dans lequel évolue la machine qu'il contrôle : une observation directe, une visualisation par caméra de contrôle (non prise en compte par le système informatique), ou encore l'affichage de mesures physiques sur un écran de contrôle. Ce dernier cas concerne majoritairement les applications de type *supervision de processus continus* (par exemple, le recuit continu à SOLLAC dans la sidérurgie) où les caractéristiques principales des objets sont affichées en permanence à l'écran. Il se peut aussi qu'il n'y ait aucune interaction avec l'extérieur, comme c'est le cas lorsque la machine gère l'affichage de l'environnement. Dans ce cas, il n'y a plus besoin que le système recoure à une caméra CCD ou à un dispositif analogue.

La visualisation virtuelle à l'écran. Considérer des environnements virtuels revient à effectuer une métaphore de l'environnement réel en le plongeant à l'intérieur de la machine, imposant ainsi l'utilisation d'interfaces graphiques [Coutaz, 1990] [Helander, 1988] [Meinadier, 1991]. De manière générale, des métaphores sont à la base de la plupart des modèles d'interfaces. Elles transfèrent une compréhension ou une perception supposée existante chez les utilisateurs sur un nouveau contexte, tant au niveau des objets que des opérations ou des règles de manipulation. Souvent, l'interface graphique utilise la métaphore du bureau pour donner un cadre général à l'interface, avec les icônes comme métaphores réduites pour symboliser les objets du bureau et permettre leur manipulation directe, rappelant ainsi les gestes de la secrétaire à son bureau. C'est tout particulièrement le cas des intégrateurs qui permettent à l'utilisateur de manipuler, sous forme d'icônes, les objets du système, et notamment de créer leurs hiérarchies de fichiers à l'aide d'icônes représentant des dossiers ou des classeurs. À cela, s'ajoutent diverses métaphores secondaires : métaphore du formulaire papier, de la corbeille à papiers, du tableau de bord avec ses boutons. . . Un exemple bien connu est celui de l'interface du Macintosh, reprenant celle développée par Xerox Parc.

2. Charge Coupled Device (Dispositif à charges conjuguées).

Par ce biais, il devient possible d'agir directement au niveau des objets qui sont visualisés, ce que l'on appelle la *manipulation virtuelle d'objets* (Cf. figure 1.2 par rapport à la figure 1.1). Au niveau de l'action directe de l'utilisateur, cela pourra donc correspondre à

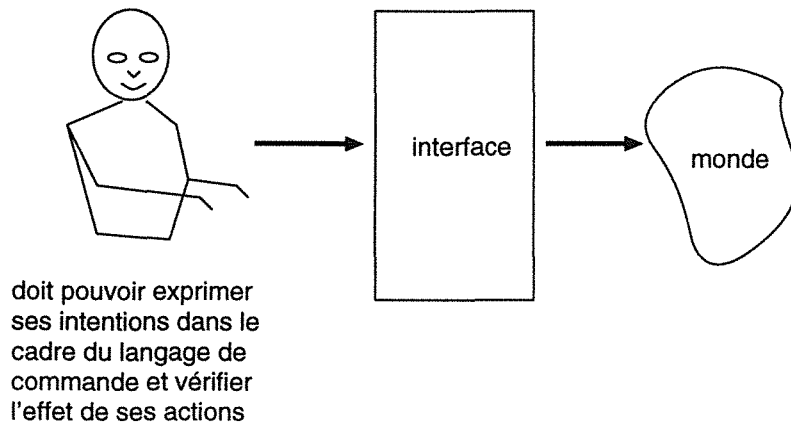


FIG. 1.1 - *Interaction traditionnelle*

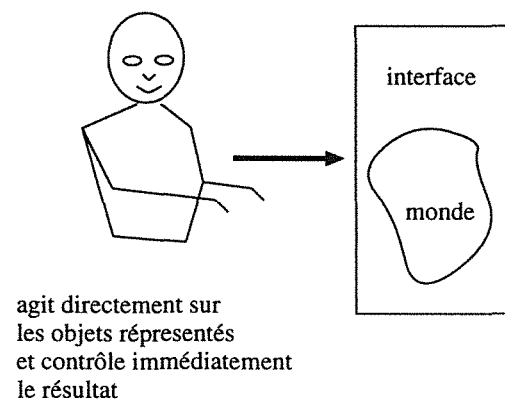


FIG. 1.2 - *Manipulation virtuelle d'objets*

une motricité virtuelle. De ce fait, l'utilisateur crée, modifie, manipule des objets à l'écran avec l'impression qu'il n'y a pas d'intermédiaire entre lui et ce qu'il voit à l'écran. Les opérations sont incrémentales à effet immédiat, ce qui permet le contrôle de l'action. Elles sont en outre réversibles puisqu'il est généralement possible de revenir à l'état précédent par la manipulation inverse. La distance entre action et évaluation se trouve de ce fait abolie. La manipulation directe présente ainsi de nombreux avantages : elle permet un apprentissage rapide tout en étant efficace pour les experts ; elle donne une visibilité totale de ce qui est pertinent ; elle limite les efforts de mémoire, l'écran étant commun à la mémoire humaine et à celle de l'ordinateur ; elle permet l'expérimentation grâce à la vérification immédiate et à la réversibilité des opérations.

Même sans offrir à l'utilisateur la possibilité de manipuler les objets à l'écran, la simple figuration des objets à l'écran présente un autre avantage : il devient dès lors envisageable de les désigner par le geste, en complément de l'expression vocale. Le résultat est un gain en confort au niveau de la communication.

1.2.3 La nécessité d'instaurer un dialogue

Imaginer un système informatique sans composante dialogue reviendrait à supposer qu'il lui soit possible de réaliser directement une tâche globale qu'aurait exprimée l'utilisateur. Autrement dit, cela correspondrait à l'hypothèse que l'utilisateur obtienne directement ce qu'il veut (la réponse à des questions, l'exécution d'une suite d'actions, ...). Une telle éventualité ne peut se rencontrer qu'à la condition que l'utilisateur puisse exprimer dans sa globalité tout ce qu'il veut, sans susciter d'erreurs. Ainsi, dans l'hypothèse où l'utilisateur exprimerait exactement ce qu'il veut pour tout ce qu'il entreprend, connaissant toutes les contraintes de l'application auxquelles il doit se plier, la question du dialogue ne se poserait pas.

Mais en règle générale, cela n'est pas envisageable. Pratiquement, l'utilisateur élabore son but au fur et à mesure du déroulement de son activité, selon une démarche qui n'est pas nécessairement linéaire. L'utilisateur n'a pas non plus à connaître toutes les contraintes liées à l'application auxquelles il doit se plier. Enfin, s'il utilise un langage proche du langage naturel, cela conduit inexorablement à une part d'implicite que la machine doit savoir interpréter [Hoc, 1988] (p. 53).

« Dans toute description de tâche, il y aura une part d'implicite. Toutes les conditions ne sont pas données et on pourra distinguer les conditions explicites qui sont fournies par la description et les conditions implicites qui sont passées sous silence mais doivent être prises en considération (...) Quelquefois, la tâche ne donne lieu à aucune prescription explicite et la prescription est donnée à un niveau très général et le sujet (l'ordinateur, en l'occurrence) doit définir lui-même les unités adéquates. »

Nous pourrions rajouter : ou bien en faire la demande à l'utilisateur. La gestion d'un dialogue entre l'utilisateur et l'application devient dès lors une nécessité. Il y aura d'une part les messages provenant de l'utilisateur et d'autre part ceux provenant de l'application. Or, il faut bien distinguer ce qui relève des intentions véhiculées par l'utilisateur et ce qui relève des possibilités d'action liées au système, chacune de ces composantes ayant une autonomie relative. Dans les systèmes de dialogue, on se place dans le cadre d'un dialogue coopératif, ce qui revient à restreindre la part d'autonomie entre les deux protagonistes du dialogue. La question est donc de savoir jusqu'où l'on peut prétendre à une coopération entre utilisateur et système.

La coopération en question n'est pas à confondre avec la convivialité, qui relève davantage du confort que peut apporter le système à l'utilisateur. Elle est souvent réduite à la prise en compte du principe de coopération et des maximes conversationnelles de Grice [Grice, 1979]. Le principe de coopération est le suivant (d'après la traduction française effectuée par Ducrot [Grice, 1979]) :

Que votre contribution conversationnelle corresponde à ce qui est exigé de vous, au stade atteint par celle-ci, par le but ou la direction acceptés de l'échange dans lequel vous êtes engagés.

De manière plus précise, Grice propose de distinguer à partir de ce principe de coopération des maximes conversationnelles, puisant leur classification dans les quatre catégories que

Kant a mis en évidence dans *la Critique de la Raison Pure*. Ce sont en fait des règles de rationalité qui correspondent à des « efforts de coopération », à savoir :

(M1): Maxime de quantité

- Que votre contribution contienne autant d'information qu'il n'est requis (ni moins ni plus).

(M2): Maximes de qualité

- N'affirmez pas ce que vous croyez être faux.
- N'affirmez pas ce pour quoi vous manquez de preuves.

(M3): Maxime de relation

- Parlez à propos (règle de pertinence).

(M4): Maximes de manière

- Evitez d'utiliser des expressions obscures.
- Soyez bref.
- Soyez méthodique (règle concernant l'ordre des informations à donner).

Il existe différentes façons de ne pas satisfaire ces règles: les violer sciemment en toute quiétude (et induire les autres en erreur), refuser de jouer le jeu (en indiquant son intention de ne pas coopérer), apporter une contribution qui fasse entrer en contradiction deux règles³, ou encore transgresser ouvertement une règle de manière à laisser présager à l'auditeur davantage que ce qui est explicitement dit⁴.

Néanmoins, dans le cadre limité du dialogue homme-machine qui est le nôtre, de tels effets rhétoriques sont inexistantes. On peut donc se placer dans l'hypothèse où l'utilisateur est idéalement coopérant, en respectant le principe de coopération et les maximes conversationnelles. On peut même en arriver à l'extrémité où le respect scrupuleux de ces principes est considéré comme une prémisse du processus de compréhension. Ainsi, la présupposition que l'utilisateur veuille uniquement ce que le système soit en mesure de réaliser a été effectuée dans un certain nombre de systèmes de dialogue afin de satisfaire l'un des deux critères suivants :

1. Simplifier la reconstitution du non-dit à partir de la reconstitution préalable d'un plan d'action.

3. Par exemple, si à la question « Où habite Bertrand? », je répons « Quelque part en Lorraine . . . », je ne respecte pas la maxime de quantité (la personne posant la question s'attend à une réponse plus précise). Mais, si j'en disais plus, je pourrais enfreindre la maxime de qualité (je ne sais plus le nom de la ville). De manière générale, les maximes n'ont pas toutes le même degré d'importance.

4. Cette dernière possibilité est à la base de la théorie des implicatures de Grice. Ainsi, en bafouant la maxime de manière « Soyez bref » dans l'énoncé *La Castafiore produisit une série de sons qui correspondait étroitement à L'air des Bijoux* (au lieu de *La Castafiore chanta L'air des Bijoux*), on implique qu'elle chanta effroyablement (exemple emprunté à [Moeshler et Reboul, 1994]).

1. La finalité de la communication : commander une application

2. Obliger l'utilisateur à suivre un protocole et lui éviter ainsi de commettre des erreurs⁵.

On peut dès lors considérer le dialogue comme la condition pour maintenir l'objectivité⁶ que nous supposons exister entre les deux protagonistes, par rapport aux connaissances qu'ils partagent sur les objets de l'environnement applicatif.

1.2.4 Un schéma général d'interaction

Nous sommes à présent en mesure de dresser un schéma général concernant l'interaction homme-machine avec une application de type commande. Par rapport à l'utilisateur, nous distinguons l'interaction avec cet environnement (production motrice et perception sensorielle) et l'interaction avec la machine (production et perception de messages), comme illustré dans la figure 1.3.

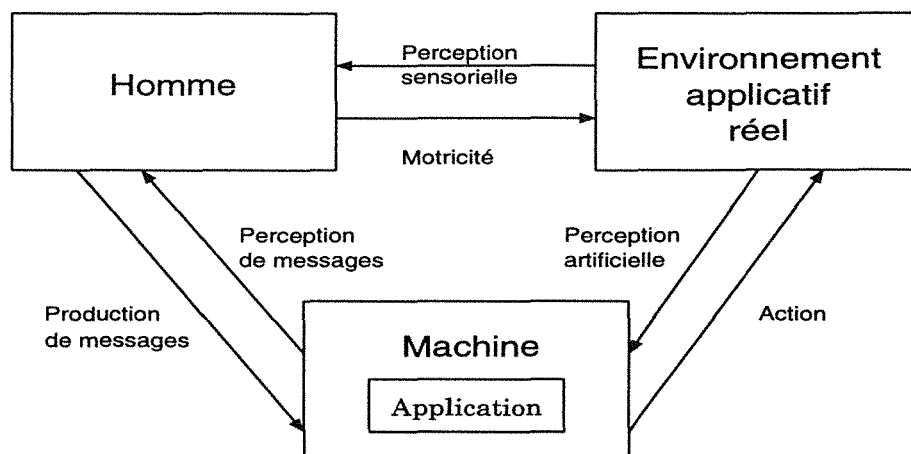


FIG. 1.3 - Communication homme-machine avec environnement physique réel

1.2.5 Le choix de la communication : l'acte de langage directif

Deux possibilités s'ouvrent à l'utilisateur : faire par lui-même ou faire faire. C'est dans cette dernière éventualité que l'on parle de commande. Nous laisserons de côté ce qui est relatif à la manipulation des objets (nous renvoyons le lecteur à [Coutaz, 1990] [Helander, 1988]), pour ne retenir que le deuxième cas, consistant à *faire faire*. On considèrera alors que les énoncés de l'utilisateur sont des énoncés de commande. On peut faire un rapprochement direct avec les actes de langage tels que les ont décrits [Austin, 1962] et [Searle, 1983]. Il s'agit en effet d'un acte de langage directif adressé à l'ordinateur.

Notons que les actes de langages directifs ne sont pas spécifiques aux applications de type commande. Considérons par exemple l'énoncé :

(1.1) Montre-moi les tableaux peints par Renoir.

5. Remplir ce critère a été l'objectif principal du système SATIC (Brunessaux, 89), destiné à assister un chirurgien pour son diagnostic.

6. Pour les autres types d'application, il s'agit d'augmenter la part d'objectivité des connaissances par rapport à tel ou tel protagoniste.

Un tel énoncé fait-il partie d'une application de type « commande » α ou « autre » β ? Une telle question est difficile à trancher. Il serait en effet possible d'associer un tel énoncé tant à une commande (DISPLAY(document)) qu'à une requête de type SQL⁷ (SELECT ... FROM ... WHERE ...). En fait, il n'y a pas de rupture entre α et β . On peut en effet faire le lien entre perception sensorielle et accès aux connaissances d'une part, action motrice et actualisation des connaissances d'autre part. La virtualisation de l'environnement permet de passer facilement de l'objet concret du monde physique à l'objet abstrait du monde cognitif qu'est la connaissance.

D'un certain point de vue, le paradigme du *faire faire* revient à considérer que l'utilisateur n'effectue pas lui-même la tâche mais la prescrit (totalement ou partiellement) à l'ordinateur. Nous parlerons de *délégation de tâche*. À partir de là, on peut considérer d'une part la tâche propre à l'utilisateur et d'autre part celle dévolue à l'ordinateur. Cela ouvre alors une nouvelle perspective pour envisager le dialogue, conçu comme une négociation par rapport à une compétence partagée pour réaliser une tâche d'ensemble.

2 L'intérêt du langage naturel dans une interface

2.1 L'adaptation à l'utilisateur

Nous partons du principe de base suivant (slogan d'apprenti ergonomes, ou vœu pieux), selon lequel c'est à la machine de s'adapter à l'homme, et non l'inverse. Cette seconde option consisterait en effet à contraindre l'utilisateur à s'adapter aux exigences de l'application informatique. Or, cela n'est pas dépourvu d'effets néfastes comme le montre [Davis et Hersh, 1986] :

« Une informatisation excessive ne peut produire qu'un mode d'existence fait d'actions purement formelles et dépourvues de sens. Car l'ordinateur fonctionne selon des langages déterminés, des instructions précises, des programmes abstraits et généraux où la signification profonde du travail effectué devient secondaire ; il engendre un formalisme sans âme.

« On décrit souvent les ordinateurs comme des esclaves neutres, mais complaisants. En fait, le danger ne vient pas de l'existence de tels robots, mais du fait que les êtres humains eux-mêmes risquent de se robotiser en s'adaptant à leur fonctionnement abstrait et rigide. »

À cette fin, nous souhaitons incorporer dans l'interface le nécessaire pour que l'adaptation à la machine se fasse avec un minimum d'efforts de la part de l'utilisateur. Il s'agit de permettre l'utilisation du mode de communication naturel de l'homme : la parole associée au geste. Il devient dès lors envisageable de rendre des dialogues de commande accessibles à un public plus large que l'utilisateur chevronné devant sa console spécifique. Dans cette perspective, on peut fortement espérer que le naturel de la communication aille de pair avec l'efficacité dans la réalisation de la tâche. Précisons toutefois qu'il s'agit

7. Standard Query Language (Langage d'interrogation standard)

là d'une direction de travail dans laquelle s'est engagée notre équipe, préjugant d'un certain nombre de critères qui ne prédominent pas forcément dans toutes les tâches de nature professionnelles, mais du moins dans toutes celles qui s'adressent à un public plus large (ne bénéficiant pas d'un entraînement préalable).

2.1.1 Une compétence déjà acquise

Selon un point de vue largement répandu, [Chomsky, 1965; Pottier, 1987] définissent une langue naturelle comme le résultat d'une lente élaboration non contrôlée. La langue est alors conçue comme une partie intégrante de l'utilisateur. Cela a pour effet de ne pas nécessiter d'apprentissage de la part de l'utilisateur, ni d'effort à effectuer lors de son utilisation.

2.1.2 Une bonne efficacité d'emploi

Un avantage du recours à une entrée vocale est de libérer les mains de l'utilisateur et de détacher son regard de l'écran. En outre, par rapport à la sélection dans un menu, l'usage de la parole présente l'avantage de ne pas masquer l'affichage comme cela est le cas lors du déroulement d'un menu [Souvay, 1992]. Ainsi, l'utilisateur peut mener à bien sa tâche sans être gêné par des contraintes de communication.

Par ailleurs, une caractéristique partagée tant par l'écrit que l'oral concerne le pouvoir d'expression de la langue, permettant d'exprimer ses intentions selon le degré d'abstraction nécessaire. Notamment, inclure des structures plus complexes empruntées aux langues naturelles se révèle nécessaire dès que la tâche elle-même se complexifie. Mais par rapport à l'écrit, un avantage de l'oral réside dans son efficacité : alors qu'un utilisateur entraîné frappe au clavier 20 mots à la minute et écrit à la vitesse de 25 mots à la minute, il peut parler en moyenne à 200 mots à la minute [Poirier et Lefebvre, 1993]. D'autre part, une désignation gestuelle, qu'elle s'effectue par sélection ou entourage [Bellalem et Romary, 1993; Bellalem, 1994] procure un net gain de temps par rapport à une description entièrement orale. De ce fait, alliée au geste, la modalité orale peut aussi se révéler très concise. Toutefois, il serait illusoire de s'imaginer que l'utilisateur puisse exprimer ses intentions selon un degré d'abstraction quelconque.

2.2 Les facteurs limitatifs de l'adaptation à l'utilisateur

Il est clair que l'on ne sait pas à l'heure actuelle traiter automatiquement toute la langue, ce qui introduit inexorablement des limitations. À cela, on peut avancer diverses raisons.

2.2.1 Une taille de lexique restreinte

Les impératifs liés à la reconnaissance lexicale imposent au lexique une taille restreinte. Le lexique comporte classiquement des mots d'usage général et des mots spécifiques à l'application. Or, il est difficile de prévoir tous les mots susceptibles d'être prononcés par l'utilisateur. On sait notamment qu'il est tout à fait possible de déborder de ce lexique en allant de proche en proche (par exemple, en utilisant des paraphrases). En outre,

l'application peut elle-même introduire des noms d'objets ne figurant pas dans le lexique sur lequel repose la reconnaissance. C'est par exemple le cas pour les noms arbitraires que peuvent avoir les fichiers ou répertoires. Cette contrainte impose le recours à une interface graphique pour permettre de sélectionner des objets que le système ne pourrait reconnaître autrement. Une autre solution, plus complexe à développer, serait de disposer d'une reconnaissance analytique (syllabe par syllabe) et de retranscrire la forme sonore sous forme graphique.

2.2.2 Un fonctionnement restreint de la langue

Nous nous limitons à un *sous-langage*, en raison à la fois de la taille restreinte du lexique et de la difficulté de modéliser une langue dans son ensemble. D'abord, on ne saurait modéliser qu'une fraction limitée des phénomènes linguistiques que l'on souhaite prendre en compte : ceux que l'on a pu observer et expliquer. Ensuite, les inférences issues de l'interprétation d'un énoncé ne mettent pas seulement en jeu des phénomènes linguistiques mais nécessitent le recours à des connaissances extra-linguistiques. Citons comme exemple particulièrement illustratif l'inférence que l'on peut effectuer sur le caractère répétitif des plongées, dans l'un des deux cas suivants (cité d'après [Pottier, 1987] §234 b)) :

(1.2) Pierre a plongé pendant deux heures.
= « il a fait plusieurs plongées » (s'il n'a pas d'appareil).

(1.3) Le sous-marin a plongé pendant deux heures.
= « il est resté en plongée ».

À ce niveau s'insère pleinement l'usage que l'on fait de la langue, ce que l'on appelle la pragmatique (cf. pour un état de l'art dans cette récente discipline [Moeshler et Reboul, 1994]). Or, là aussi il est nécessaire d'être conscient des limites à ne pas dépasser.

2.2.3 La limitation par les connaissances

Les limitations dues au langage ne sont pas les seules : il faut savoir modéliser les connaissances extra-linguistiques relatives aux domaines sur lesquels porte le discours [White, 1990]. Une solution est de se limiter au domaine de l'application. Nous pouvons en outre ajouter que la nécessité de se restreindre à un domaine limité par une application répond à une exigence sur le plan technique, dès que l'on devient ambitieux quant au naturel des conditions de communication envisagées, en cherchant à rivaliser (du point de vue des formes d'expression) avec le dialogue que pourraient se tenir deux individus⁸.

8. Dès le début des années 1970, Winograd a très bien illustré ce point avec son système SHRDLU [Winograd, 1972] [Winston, 1977], l'un des premiers systèmes de dialogue et de compréhension d'un sous-langage de l'anglais à être guidé par le souci de faire émerger de l'énoncé un sens plutôt qu'une structure formelle. Persuadé qu'un ordinateur ne peut traiter raisonnablement le langage à moins d'avoir une relative compréhension du sujet de conversation, Winograd a eu l'idée de restreindre fortement le domaine d'application au profit d'une très grande liberté d'expression. Il a choisi une application qui se prête facilement à la modélisation des connaissances et au raisonnement : la simulation du comportement d'un robot qui agit sur un monde clos constitué d'une table sur laquelle sont posés des objets de diverses formes, tailles et couleurs.

Les techniques d'Intelligence Artificielle donnent aux ordinateurs des capacités de raisonnement très limitées qui nécessitent d'explicitier un très grand nombre de connaissances relativement à la complexité des tâches qu'ils permettent d'accomplir. Il s'agit dans ces conditions de ne pas faire exploser sur le plan combinatoire le nombre de connaissances que doit utiliser le système informatique.

2.2.4 Inférences : les limitations des techniques d'IA

Dans cette optique, on se limite du point de vue sémantique à la partie opérative pertinente pour l'application [Memmi, 1990]. L'approche prônée par D. Memmi considère le sens d'une expression comme les inférences qui y amènent ou en découlent avec une précision arbitraire. Il s'agit d'une sémantique fonctionnelle, où le sens d'une expression symbolique n'est rien d'autre que sa fonction dans le système, c'est-à-dire la totalité des interactions entre cette expression et le reste du système. Ces interactions se font classiquement en Intelligence Artificielle par associations du même symbole se trouvant dans différentes formules déclaratives à l'aide de mécanismes standards très généraux (unification, schémas d'inférence avec *pattern matching*, stratégies de recherche). Toutefois, il est difficile d'accepter que dans une sémantique, la référence ne joue aucun rôle dans le sens. Dans le cadre d'une sémantique liée à un modèle du monde, D. Memmi suggère d'aller un peu plus loin en donnant une définition opérationnelle de certains symboles. Le sens d'un symbole pourrait alors s'identifier avec les processus par lesquels le système s'assure de la présence de l'objet désigné dans le monde extérieur ou bien encore avec les actions à effectuer sur cet objet. De cette façon, les processus référentiels participent au sens.

2.3 Entre langages artificiels et langues naturelles

On peut imaginer tout un continuum entre un langage artificiel et une langue naturelle. Il s'agit pour nous de trouver un compromis raisonnable, compte tenu des limites qu'offre un ordinateur d'une part, et ses possibilités d'utilisation d'autre part.

2.3.1 Les langages artificiels

Lorsque l'on considère des systèmes de dialogue destinés à être utilisés dans un milieu professionnel, on a typiquement affaire à un utilisateur spécialisé (un « opérateur ») face à un poste de travail tout autant spécialisé. En examinant la forme des énoncés de commandes qu'emploie l'utilisateur, on peut être frappé de constater qu'ils sont très éloignés de la manière dont on s'exprime tous les jours. C'est ce que l'on appelle un langage artificiel. On rencontre classiquement deux approches pour rendre compte de ce que recouvre un langage artificiel, présentant chacune un aspect essentiel.

Définition 1.

« Un langage artificiel est uniquement décrit par des règles formelles de construction de phrases [Souvay, 1992]. »

De telles règles sont issues de grammaires formelles que l'on peut classer selon différents degrés de sophistication :

- les grammaires régulières ;
- les grammaires récursives ;
- les grammaires à contexte libre.

L'informatique emploie classiquement de telles grammaires pour décrire des langages informatiques, comme cela se fait en compilation. De ce fait, les langages artificiels présentent un certain nombre d'avantages au niveau de la connexion avec la partie de l'interface chargée de la reconnaissance de la parole, comme nous le verrons dans le paragraphe §3.2.1.

Il est bien évident par ailleurs qu'appliquer une telle grammaire nécessite un apprentissage. D'où une seconde approche concevable pour envisager un langage artificiel.

Définition 2.

« Un langage artificiel est un langage inventé et difficile à acquérir, dans la mesure où il ne suit pas le mode de fonctionnement interne d'un sujet⁹ (d'après [Barton *et al.*, 198]). »

En reprenant l'approche de la définition 1, on peut très bien se contenter de la seule connaissance de l'ordre des constituants si cela s'avère suffisant pour comprendre et interpréter convenablement un message. [Pierrel, 1987] montre que c'est le cas pour certaines applications de dialogue oral homme-machine (DOHM) qui ne nécessitent et n'utilisent que des langages artificiels restreints (à consonance plus ou moins naturelle). Ceux-ci ont la caractéristique de pouvoir être entièrement décrits par leur syntaxe (commande de processus industriels, par exemple) et définis par une grammaire à contexte libre ou par un automate dans lequel les terminaux correspondent aux mots ou aux groupes de mots du langage. Ils ne réclament aucun traitement sémantique supplémentaire, dans la mesure où l'on peut définir une relation biunivoque entre une structure syntaxique (fournie sous la forme d'un arbre syntaxique, par exemple) et une interprétation de la requête ou de l'action correspondante.

Les symboles terminaux que l'on rencontre dans les langages artificiels ne sont pas purement arbitraires. À des fins mnémotechniques et pour réduire la difficulté d'apprentissage, on les emprunte à des langues naturelles.

Le cas des langages opératifs Des langages artificiels peuvent être envisageables pour des pupitreurs très entraînés. On peut alors parler dans ce cas de *langages opératifs* [Falzon, 1984]. Par rapport aux langues naturelles, les langages opératifs présentent la caractéristique de n'employer, à travers les divers énoncés qu'ils acceptent, que les seuls mots ou formes de mots nécessaires à une interprétation non ambiguë. Typiquement, ces

9. L'auteur parle de *lack of learnability*, expression pour laquelle il n'existe aucun équivalent direct en français.

2. L'intérêt du langage naturel dans une interface

langages n'emploient pas d'articles et ne conjuguent pas les verbes. L'hypothèse que fait Falzon est que l'univers du discours est tellement restreint que la syntaxe peut souvent être omise, pourvu qu'une connaissance suffisante du domaine de l'application soit à disposition. Il part de la constatation que la compétence linguistique n'est pas suffisante pour comprendre un langage technique. La démarche qu'il préconise est plutôt de voir comment un langage technique diffère du langage naturel, sans essayer de l'analyser à travers les méthodes propres au langage naturel. Il propose alors de construire de toutes pièces un langage spécifique, uniquement d'après les caractéristiques de la tâche et de ses objectifs. En outre, il prend en compte d'autres aspects tels que la charge de travail (avec la tendance à restreindre la longueur des messages et à en concaténer plusieurs en un seul) et l'absence d'ambiguïtés. Ainsi, dans le domaine du contrôle de trafic aérien, on dira par exemple :

(1.4) climb 330
au lieu de :

(1.5) climb to the fight level 330

Le critère prédominant est avant tout celui de l'efficacité. Mais jusqu'où peut-on s'écarter de la manière naturelle de s'exprimer sans pour autant nuire à l'efficacité? On peut alors se demander si l'on pourrait rendre plus naturelles les conditions de communication sans nuire à l'efficacité de l'ensemble.

Une caractéristique des langages opératifs tels que les conçoit Falzon est d'accorder une part négligeable à la structure de l'énoncé, se contentant de poser qu'un énoncé débute par un mot exprimant une commande. Le reste est alors obtenu en exploitant des connaissances sur les caractéristiques de la tâche. Mais une conséquence de ceci est que le langage obtenu n'est pas homogène :

« The language of the controllers is not homogeneous. Most of the time, the language they use is a technical dialect, which vocabulary and syntax are highly restricted. » ([Falzon, 1984], p. 31)

Il faut en effet savoir qu'une caractéristique importante des langues naturelles est celle de former un système homogène [Benveniste, 1974b]. Le problème que pose l'hétérogénéité d'un langage opératif lorsqu'il comporte des emprunts à des langues naturelles est le fait qu'elle rende son apprentissage et surtout son utilisation particulièrement difficiles. Il suffit pour s'en persuader de songer à la difficulté de jeux tels que le « ni-oui, ni-non »¹⁰. De la même façon, on peut s'interroger sur l'utilisation d'un langage dans lequel on ne pourrait pas utiliser, par exemple, de déterminants.

Un problème sérieux qui se pose lors de la conception de langages pour utilisateurs avertis ou simplement novices, est d'aboutir à un langage qui soit suffisamment puissant et expressif, facile à apprendre et dont on peut se souvenir après une assez longue période de non-utilisation. Une approche naïve consisterait à vouloir concevoir un langage de petite taille en se disant qu'un tel langage serait nécessairement plus facile à apprendre

10. Jeu illustrant la difficulté à répondre à des questions ouvertes par une réponse affirmative autre que *oui*, ou par une réponse négative autre que *non*.

qu'un langage plus gros, en dépit de son manque d'expressivité. [Green, 1979; Green et Payne, 1984] montrent au contraire qu'il est préférable au niveau de l'apprentissage de recourir à un langage plus imposant, présentant une cohérence explicite dans sa structure. Un novice qui apprend un nouveau langage recherche en effet des principes directeurs¹¹, parce qu'il lui est moins coûteux de découvrir une structure systématique dans le langage, que d'apprendre entièrement une collection arbitraire de règles. Ainsi, il vaut mieux un langage volumineux présentant une structure claire et cohérente qu'un langage plus petit mais dont la structure est impénétrable.

On en vient alors à conclure qu'un langage peut d'autant plus facilement être appris et utilisé qu'il calque, par sa structure même, la logique interne de l'utilisateur. Or, les langages artificiels ont justement cette caractéristique de ne pas préserver la dualité avec la pensée que décrit Guillaume, lorsqu'il affirme que chaque mot véhicule des instructions pour la diriger. Cela peut avoir des effets au niveau de l'activation de connaissances, linguistiques ou extra-linguistiques. Il s'avère par conséquent nécessaire de considérer le contenu instructionnel des mots, et pas seulement l'ordre selon lequel ils s'enchaînent. Ces arguments nous poussent à nous rapprocher des langues naturelles, en donnant non seulement un contenu aux mots formels, mais aussi en reprenant certaines autres caractéristiques des langues naturelles.

2.3.2 Vers des langages à consonance naturelle

Dans des systèmes utilisant un langage opératif contraint, il est souvent fait fi des déterminants et autres mots « inutiles » pour ne conserver que ce qui correspond à des « concepts », essentiellement noms, verbes et adjectifs. Dans ces conditions, il est difficile de parler de référence définie, sauf à l'assimiler en supposant un article défini implicite. La référence définie est en dépit de cela couramment traitée dans les systèmes reposant sur un langage artificiel.

D'une manière générale, le cadre dans lequel nous nous situons est celui des actes directifs. Cette classe recouvre tout acte dont le but est de rendre le monde conforme au contenu propositionnel. Nous mettons ainsi de côté tout ce qui concerne les actes assertifs, qu'ils soient positifs (affirmation, confirmation, description, explication, rectification, ...) ou bien négatifs (négation, contestation, critique, ...). Les énoncés de type commande s'expriment typiquement par des formes impératives, mais on peut aussi les rencontrer sous des formes infinitives dans certains langages artificiels de commande ou bien sous des formes interrogatives ou déclaratives comme formule de politesse. Nous allons passer en revue les diverses formes que peut recouvrir un énoncé de commande, en utilisant des résultats obtenus par une expérience de type magicien d'Oz [Mignot *et al.*, 1993]. L'expérience en question consistait à étudier l'utilisation spontanée et naturelle d'une interface multimodale dans des actions en situation à l'aide de dispositifs de communication simulés par un opérateur humain. En ne prenant que des énoncés visant à déplacer un meuble, on retrouve une grande diversité de formes linguistiques utilisées, que ce soit sur les plans

11. Parmi ces principes directeurs, on peut citer une consistance dans l'ordre selon lequel apparaissent les différents constituants de l'énoncé, une ressemblance à la langue que possède l'utilisateur, des indices perceptuels, l'usage de métaphores, des noms distinctifs, la congruence (présence d'un terme et de son antonyme).

intra-individuel ou inter-individuel.

– **Forme impérative classique**

Tournez la chaise, s'il vous plait.

– **Forme infinitive**

Tourner le lit.

Mettre la bibliothèque devant le mur de droite.

Déplacer la bibliothèque, pour la mettre ici.

Déplacer le fauteuil, pour la mettre en face du lit.

– **Forme modale**

Je voudrais déplacer la bibliothèque contre le mur à gauche en entrant.

Je voudrais déplacer le fauteuil près du lit. (Êtes-vous satisfait?) Non, je le veux comme ceci.

Nous allons¹² placer le fauteuil derrière le bureau.

Alors... on va déplacer le piano qu'on va mettre là.

– **Formes descriptives**

On prend la bibliothèque que l'on va porter ici.

Nous replaçons le bureau en oblique vers le centre de la pièce.

– **Formes interrogatives**

Peut-on orienter le piano un peu vers le bas?

– **Formes elliptiques**

Le bureau le long du mur.

La bibliothèque près du mur.

Le fauteuil devant le bureau.

Nous avons dans cette section pris parti pour une interface langagière à consonance naturelle pour commander une application. L'idée que nous avons défendue est un critère de régularité langagière qui peut être résumé ainsi : « Si une propriété linguistique est vérifiée sur certains énoncés acceptables, alors elle doit l'être pour tous les énoncés dans lesquels elle pourrait se manifester ».

Nous avons en outre préalablement montré que les énoncés reconnus, issus d'un dialogue entre l'utilisateur et le système, devaient être interprétés par le système afin d'aboutir à une activité. Tout cela nous conduit à mettre en évidence les diverses fonctionnalités d'une telle interface, avant de cerner celles qui retiendront notre attention.

12. Le verbe *aller* est ici considéré comme un auxiliaire modal.

3 Les fonctionnalités de l'interface

Il est primordial qu'une interface puisse comprendre les messages qu'elle fait transiter. Les signaux provenant de l'utilisateur, tels qu'ils sont reçus par les médias de perception de l'interface, ne doivent pas être seulement considérés comme des signaux physiques [Boe et Lienard, 1988, p. 81]. Il se cache derrière ces signaux toute l'intelligence de l'utilisateur, qui a su planifier une série d'actions, se les représenter sous forme de concepts pour finalement les exprimer. Pour que l'interface puisse prétendre retrouver les intentions de l'utilisateur, elle doit être en mesure de parcourir le chemin inverse. C'est en ces termes que l'on peut définir la compréhension dont doit faire preuve l'interface [Bastien et Colas, 1991]. Cela recouvre la reconnaissance, la conceptualisation (ou représentation du sens), et la représentation interne (sous une forme propre à alimenter des inférences). Enfin, pour qu'il y ait une communication possible, les représentations internes de l'interface doivent être extériorisées vers l'utilisateur de manière compréhensible.

Si l'on part de la finalité d'une interface langagière à une application de commande – aboutir à une activité de la part du système –, on peut mettre en évidence quatre grandes fonctionnalités que doit posséder ce système :

- prendre en charge l'activité (et réaliser ainsi la tâche à effectuer) ;
- interpréter ce qui est à faire à partir des messages de l'utilisateur (énoncés de commande) ;
- reconnaître les différentes portions du message à interpréter ;
- dialoguer avec l'utilisateur.

3.1 Gérer la tâche

Le rôle de cette fonctionnalité est de donner une compétence au système tant pour superviser le déroulement de la tâche, que pour assister l'utilisateur dans l'accomplissement de celle-ci. Nous nous plaçons en effet dans le cadre d'une compétence partagée dans la réalisation de la tâche. Dans ce cadre, on peut affirmer qu'un dialogue est *coopératif* lorsque la tâche de l'ordinateur obéit aux mêmes impératifs que celle de l'utilisateur. Nous pouvons ainsi préciser où se situe la synergie entre tâche et dialogue que nous avons mise en évidence dans la première section de ce chapitre : nous dirons que le dialogue naît face à la *nécessité de rendre compatibles les buts des tâches de chaque protagoniste*.

3.1.1 Un mécanisme de raisonnement

La mise en œuvre d'un tel dispositif de coopération n'est envisageable qu'à condition que le système dispose de mécanismes de raisonnement, ce qui impose le recours à certaines techniques d'Intelligence Artificielle. Cela correspond au concept d'*interface intelligente*.

« The idea of an intelligent interface presupposes that a human and machine will be working together to carry out a task . . . The intelligent interface serves

3. Les fonctionnalités de l'interface

to enhance the operation of the cognitive interface by supporting and simplifying the activities and selecting/executing operators. » [Chignell et Hancock, 1988]

Pour mettre les choses au point, précisons tout de suite qu'il n'existe pas d'interfaces vraiment intelligentes. Dans un article [Searle, 1990] qu'il a écrit à propos des limites de l'intelligence artificielle (la « chambre chinoise »), Searle argumente que les programmes informatiques sont purement formels et, de ce fait, ne sont ni constitutifs de la pensée (qui, elle, a un contenu), ni suffisants pour la produire. La raison en est que tout système capable d'engendrer des pensées devrait posséder un pouvoir causal équivalent à celui du cerveau¹³. Ainsi, un système artificiel ne produirait de phénomène mental¹⁴ que s'il reproduisait les pouvoirs déterminants spécifiques du cerveau. Or, tant qu'un système se limitera à l'exécution d'un programme formel, il ne pourra pas y parvenir. Tout au plus demandons-nous à une machine de donner un semblant d'intelligence. On se place ainsi dans une perspective de « trompe l'œil ». Et c'est bien là le sens d'*artificiel* qu'il faut percevoir à travers l'appellation *Intelligence artificielle*.

Il serait fastidieux de recenser ici toutes les caractéristiques de la coopération. Tout au plus pouvons nous en donner des exemples types pour les principaux modes de coopération, d'après les différents rôles que peuvent cumuler l'utilisateur et le système [Delouis, 1993; Monclar *et al.*, 1994].

3.1.2 Modes et rôles de coopération

Le rôle d'exécution consiste à exécuter les fonctions de l'application qui ont été sélectionnées. Le système possède toujours ce rôle. Dans le cadre du paradigme du *faire faire* où nous nous plaçons, nous excluons de fait ce rôle pour l'utilisateur.

Le rôle de décision consiste à sélectionner les fonctions de l'application à exécuter.

Une des questions principales concerne l'indépendance de la machine, c'est-à-dire la prise d'initiative spontanée au niveau d'une activité motrice. On peut rencontrer ce cas dans le domaine du contrôle de trafic aérien. Afin d'alléger la charge du contrôleur aérien, la machine peut prendre elle-même en charge certains plans de vols faciles à traiter [Gerlach et Onken, 1993] :

« The Automatic Flight Planner (AFP) provides a flight plan as basic guide line for the flight. An initial flight plan may be fed into the system or is generated autonomously by the AFP using problem solving algorithms considering constraints of the crew, of air traffic control (ATC), weather conditions and available navigation aids. The generated flight plan is presented to the crew as recommandation and, if not ajusted by the crew, serves as input for the module Piloting Expert (PE). »

13. Des événements neuronaux de base (de la sensation de soif à l'évocation d'un théorème mathématique ou d'un souvenir d'enfance) engendrent des phénomènes mentaux de niveau supérieur, selon une causalité « ascendante ».

14. Ce qui revient à dire qu'il serait un cerveau artificiel.

On peut en outre citer l'exemple des jeux de stratégies où l'initiative d'action est prise à tour de rôle pour modifier l'environnement que constitue le plateau de jeu.

Le rôle d'assistance englobe les étapes de conseil sur les choix et validations à effectuer. L'assistance peut se manifester par des confirmations de la part de l'utilisateur pour des décisions que le système ne peut pas prendre à lui tout seul.

Le rôle de critique permet de donner un conseil sur les choix et validations effectués. Cela peut se rencontrer dans le domaine du contrôle de trafic aérien, si un opérateur fait manifestement une erreur [Gerlach et Onken, 1993] :

« The module Pilot Intent and Error Recognition (PIER) compares the expected pilot behaviour as generated by the PE with the actual behaviour of the crew. The crew actions are deduced from an intensive situation assessment. In general, deviations from the expected behaviour are due to pilot error. In this case, a message is issued to make the crew aware of the situation. In some cases, the crew might deviate from the flight plan intentionality (e.g. a missed approach procedure). These events are recognized and serve as new inputs for all CASSY modules to activate appropriate support and to change the valid flight plan accordingly. »

3.2 Reconnaître les messages

L'interprétation d'un message de l'utilisateur suppose qu'il y ait une reconnaissance, partielle du moins, de celui-ci. Dans le cadre où nous nous situons, nous privilégions une communication à forte composante orale, présentant une consonance naturelle. La modalité liée à la voix pourra donc être considérée comme pivot du dialogue. Mais ce n'est pas la seule qui intervienne dans notre cadre. D'une manière générale, un *système multimodal* est constitué de N modalités en entrée, chaque modalité étant sujette à un traitement indépendant. Il doit donc intervenir diverses composantes liées à la prise en compte de telle ou telle modalité. Elles consistent à reconnaître les divers signaux physiques (acoustiques, gestuels, ...) tels qu'ils apparaissent à l'entrée du système. Mais cela n'est pas suffisant. Il faut en outre prévoir de donner suite aux diverses erreurs de reconnaissance pouvant intervenir et donc gérer le *canal de communication*. Nous allons ainsi présenter les différentes composantes qui interviennent lors de la reconnaissance d'un message multimodal.

3.2.1 La reconnaissance vocale

Dès lors que l'on considère la voix comme modalité primaire du message, la reconnaissance du message dans son ensemble repose entièrement sur les résultats de la reconnaissance vocale. Aussi est-il important d'intégrer le système de reconnaissance de sorte à avoir un maximum de performances.

Performances d'un système de reconnaissance vocale. Au niveau des performances de reconnaissance, il y a grand avantage à figer le langage en imposant une

structure d'énoncé à respecter, notamment en utilisant des grammaires formelles, qui contraignent l'enchaînement des mots les uns par rapport aux autres. Pour mesurer ces contraintes, on rend compte du coefficient de perplexité.

Le coefficient de perplexité se définit comme le nombre moyen de mots prévisibles à partir du modèle du langage, en tant que successeurs de chaque mot du texte. Cela est donc égal à l'inverse de la probabilité de trouver la séquence de mots du texte [Jardino et Adda, 1993]. Si en première approximation, on prend seulement en compte l'influence d'un mot sur le suivant, la perplexité PP d'un texte de L mots devient :

$$PP = \exp\left(-\frac{1}{L} \sum_{l=1}^L \ln \Pr(\omega_l | \omega_{l-1})\right)$$

où $\Pr(\omega_l | \omega_{l-1})$ représente la probabilité de trouver le mot ω_l après le mot ω_{l-1} .

On peut citer l'exemple de *Sundial* [Charpentier *et al.*, 1992] pour lequel un retour d'une composante de grammaire formelle fait évoluer de manière significative le taux de reconnaissance de phrases. L'emploi de telles grammaires, fortement contraintes par la syntaxe, a donc pour effet d'accroître la fiabilité et la robustesse du système. On peut même dire que plus le langage sera contraint syntaxiquement, plus grande sera la fiabilité du système [Calliope, 1989]. Parmi les facteurs importants à prendre en compte, nous pouvons citer :

- la faillibilité. Il faut accepter qu'il puisse y avoir des erreurs de reconnaissance ;
- la nécessité d'être prédictif pour améliorer le taux de reconnaissance.

Pour illustrer ce dernier point, nous pouvons reprendre l'exemple du contrôle de trafic aérien [Gerlach et Onken, 1993], avec des énoncés du type :

(1.6) Turn left to heading 180.

Le grand nombre de possibilités au niveau des paramètres conduit rapidement à des nombres astronomiques quant aux énoncés pouvant être engendrés par la grammaire. Lorsque le langage n'est pas restreint par rapport à la situation courante, Gerlach aboutit à une syntaxe pouvant être décrite par une perplexité de 16 avec 218 mots. En revanche, si l'on utilise une représentation des actions attendues de la part du contrôleur pour prédire la tâche, cela fait passer la perplexité à 3.

3.2.2 La reconnaissance gestuelle

Des gestes de désignation pourront très naturellement venir compléter ce qui a été désigné par le langage, par exemple par entourage d'une zone délimitant un ensemble d'objets [Bellalem et Romary, 1993; Bellalem, 1994]. L'utilisation d'un tel système de reconnaissance gestuelle présuppose un environnement graphique où soient représentés les objets susceptibles d'être désignés. On peut envisager de recourir à une souris, un gant de désignation, un écran tactile,...

Un intérêt majeur de la désignation gestuelle est de pouvoir désigner des objets que l'on ne préfère pas nommer, se conformant à des principes d'économie. Cela peut même

devenir une nécessité si le module de reconnaissance vocale ne connaît pas à l'avance le mot en question. On peut par exemple citer le cas de l'interface à un système d'exploitation, où les noms de fichier et de répertoire doivent être directement saisis au clavier. Nous voyons donc que ce qui prime est la complémentarité des différents modes, à travers ce qui constitue la spécificité de chacun.

Mais si l'on veut reconstituer le message dans son ensemble, une question qui s'ouvre est de savoir où l'on effectue l'articulation entre les différentes modalités.

3.2.3 La question de l'articulation entre modalités

Les figures 1.4 et 1.5 illustrent les deux grandes classes d'approches que l'on peut adopter pour « combiner les événements » identifiés séparément au niveau des différents médias [Azémard et Macchion, 1993].

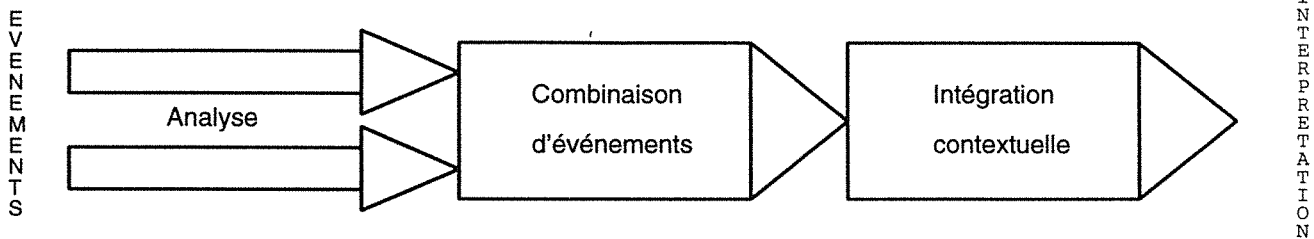


FIG. 1.4 - Approche « morpho-syntaxique » (1) de la multimodalité

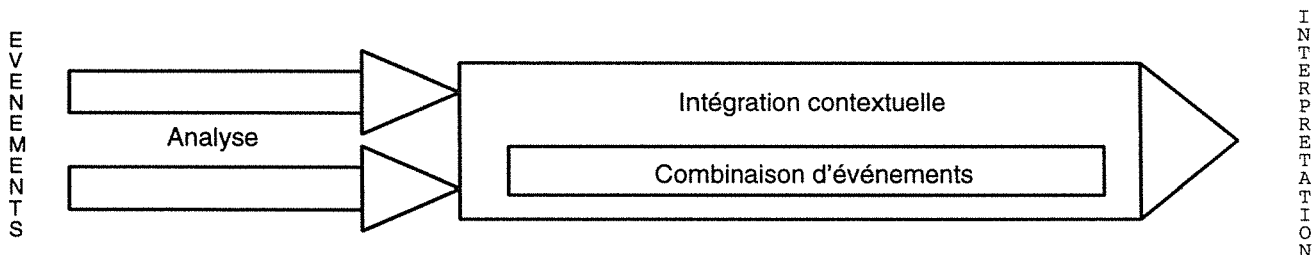


FIG. 1.5 - Approche « sémantico-pragmatique » (2) de la multimodalité

Dans la première approche, la combinaison des événements reconnus s'effectue à partir de leur représentation morpho-syntaxique, en se basant sur des critères purement temporels de co-occurrence. Différentes combinaisons sont alors essayées et c'est le contexte du dialogue qui valide telle ou telle combinaison. Des applications telles que ICPdraw [Caelen *et al.*, 1991], ICPplan [Bourguet, 1992] utilisent cette approche.

D'autres applications telles que MELODIA [Duermael et Pierrel, 1992; Pouteau *et al.*, 1993], MMI2 [Falzon *et al.*, 1991] utilisent une approche complètement différente. La gestion du dialogue a la tâche plus générale de guider l'interprétation sémantique de chaque modalité, en utilisant le contexte lié à la situation courante. Cette approche est également celle que nous prônons, le recouplement s'effectuant selon nous d'après des critères référentiels.

D'une manière générale, on ne saurait manipuler le langage comme on assemble des briques, d'un point de vue purement formel (même s'il faut bien partir de là) sans s'intéresser au contenu de ces briques. De plus, ce contenu est sensible au contexte, si bien que l'on ne saurait effectuer de séparation nette entre ce qui relève purement de l'identification et ce qui relève de l'interprétation : les deux fonctionnent en symbiose. Autrement dit, il faut pouvoir reconnaître un énoncé pour pouvoir espérer le comprendre ; mais si on en reconnaît seulement quelques bribes, une compréhension d'ensemble pourra aider à reconnaître le restant. Dans la section suivante, nous allons nous intéresser aux fonctionnalités que doit posséder une composante destinée à interpréter un énoncé de commande, en nous focalisant plus particulièrement sur un aspect lié à cette compréhension d'ensemble – la référence aux actions – qui constitue le sujet de ma thèse.

3.3 Interpréter l'énoncé en termes d'actions

Dans le cadre du dialogue de commande où nous nous situons, l'interprétation de l'énoncé a pour finalité une activité de la part du système. Pour interpréter alors ce qui est à faire, il est nécessaire de déterminer quels sont les objets visés par l'énoncé, et quelles actions leur appliquer.

3.3.1 Sur quels objets porte l'énoncé ?

Le problème de la détermination des objets qui sont désignés dans l'énoncé s'appelle classiquement la *référence aux objets*. Supposons par exemple un écran sur lequel sont affichées une fenêtre verte contenant un texte, une fenêtre rouge contenant une image et une icône bleue. Dans la série d'énoncés suivante (reprise de chez [Gaiffe, 1992]), les portions de l'énoncé référant à un objet sont représentées en italique.

- (1.7) Déplace *la fenêtre verte* pour *la* mettre ici (accompagné d'un geste).
- (1.8) Transforme *la fenêtre rouge* en icône et mets *cette icône* en bas à gauche.
- (1.9) Agrandis *l'icône bleue*.
- (1.10) Mets *la* à gauche de *la fenêtre verte*.
- (1.11) Enlève *l'ascenseur vertical* de *cette dernière*.
- (1.12) Crée *une fenêtre bleue*.

On devine facilement les portions de l'énoncé qui vont pouvoir correspondre à la désignation d'objets : il s'agit des *groupes nominaux*. Nous pouvons distinguer différents types de référence, suivant la nature des groupes nominaux qui réfèrent :

- le pronom défini (ex : *la*). De manière générale, le pronom défini est incapable de référer directement à un objet mais reprend un élément linguistique à l'intérieur du discours qui lui fournit des contraintes pour sélectionner un référent [Milner, 1982]. Classiquement, ce phénomène se nomme une anaphore.

- le groupe nominal défini (ex : la fenêtre verte). Au premier abord, il est destiné à filtrer un référent dans un environnement. Toutefois, il peut également être anaphorique, de même que le pronom défini.
- le groupe nominal indéfini (ex : une fenêtre bleue)
- le groupe nominal démonstratif (ex : cette dernière, cette icône). Le groupe nominal démonstratif peut avoir un fonctionnement anaphorique en reprenant un élément saillant du discours mais il est avant tout dédié à la co-référence avec le geste.

Toutefois, il existe des groupes nominaux qui ne réfèrent pas. Dans la série d'énoncés ci-dessus, on peut relever par exemple *icône* dans l'énoncé 1.8, groupe nominal sans déterminant. Même en employant un déterminant, la désignation d'un objet (directe ou anaphorique) n'est pas obligatoire, comme l'illustre l'exemple suivant :

(1.13) Fais le ménage.

Il n'existe pas en effet d'objets visibles figurant dans l'environnement applicatif qui puisse être désigné par le groupe nominal *le ménage*. Dans ce dernier cas, il apparaît clairement que le groupe nominal joue un rôle prépondérant dans la détermination de l'action à exécuter. Nous verrons que c'est également le cas pour tous les groupes nominaux en général. Nous effleurons déjà l'autre volet de l'interprétation d'un énoncé de commande – la référence aux actions – qui constitue le problème abordé dans ce mémoire.

3.3.2 Quelles actions appliquer ?

Le problème de la référence aux actions consiste à *identifier* les actions à effectuer, telles qu'elles sont véhiculées par les commandes de l'utilisateur. Ce problème est directement lié au but de l'utilisateur : aboutir à une activité du système à partir d'énoncés de commande. En définitive, il s'agit de mettre en correspondance une séquence d'énoncés et une activité du système, dans le sens d'une séquence d'exécution de fonctions de l'application.

Au cœur du problème, la notion d'action : La notion d'action étant très intuitive et source à nombreuses interprétations, nous allons tenter d'en donner une définition par rapport au cadre que nous nous sommes fixé, tout en préservant un certain niveau de généralité. Nous retiendrons la définition d'une action comme *un passage contrôlé entre une situation et une autre situation*. Cette définition comporte divers aspects interdépendants qu'il faut éclaircir : la dépendance par rapport à la situation et le contrôle qui lui est lié.

La situation : Nous définissons la *situation* comme *un ensemble de relations entre objets de l'application*. Une situation peut donc être obtenue à partir de l'état de l'application vu par l'interface, en exhibant les relations entre les objets affectés par l'action. Nous voyons donc à ce niveau que la notion d'objet est étroitement liée à celle d'action.

3. Les fonctionnalités de l'interface

Un passage entre situations : Nous employons le terme de *passage* pour désigner toute l'étendue du continuum allant de la modification brutale (ex : *Eteindre la lumière*) à la lente évolution (ex : *Tondre la pelouse*).

Comme dans toute échelle continue s'étendant sur un domaine conceptuel, la langue implique une catégorisation (issue d'un jugement), indiquant si, compte tenu de la situation et du contexte linguistique, on doit plutôt considérer que le passage est discret ou qu'il est continu¹⁵, comme dans *fermer la porte*. Par ailleurs, on remarquera que n'importe quel passage entre situations ne saurait constituer une action : pour constituer une action, le passage doit être contrôlé.

Le contrôle de l'action Dans la définition de l'action que nous avons donnée, l'aspect de contrôlabilité joue un rôle primordial. Une action ne saurait se concevoir en dehors de ce qui la contrôle. La notion de contrôle est très spécifique et n'existe que dans certaines classes de procès. Par exemple, elle n'intervient pas dans les phrases suivantes :

Ce morceau de fer a rouillé.

Le vent souffle de plus en plus fort¹⁶. (*force incontrôlée*)

Jean aime les gateaux. (*verbes de sentiment*)

Jean connaît la liste des départements par cœur. (*verbes d'état*)

Jean ne voit rien à plus de trois mètres. (*verbes de perception*)

Jean a reçu un coup de poing¹⁷. (*verbes à sens passif*)

Agent et agentivité : Aucune des phrases ci-dessus ne dénote une action car aucune ne fait référence à une entité contrôlante (ou plus exactement jugée comme telle), que l'on nomme généralement *agent*. On peut d'ailleurs remarquer qu'aucune de ces phrases ne pourrait se mettre à l'impératif. Les agents et activités disposant d'un champ illimité de variabilité l'un par rapport à l'autre, cela impose tout un continuum de *contrôle*, que l'on appelle généralement en linguistique *l'échelle d'agentivité*. Par exemple, [Desclès, 1990] introduit trois primitives d'agentivité :

1. FAIRE : avoir la capacité d'effectuer la modification ;
2. CONTRÔLE : avoir la maîtrise du changement pour le déclencher ou pour l'interrompre ;

15. Nous effleurons ici la question de l'aspect verbal (la manière dont le déroulement de l'action est envisagé), notion-clé sur laquelle nous reviendrons.

16. Dans ces deux phrases, il y a un changement de situation qui est exprimé, mais on ne l'impute à aucune entité qui pourrait le contrôler. Par contre, il peut arriver que l'on prête métaphoriquement un contrôle à des entités qui en sont théoriquement dépourvues. Par exemple, dans des phrases telles "La mer recule." ou "La mer a englouti un pêcheur.", l'emploi de verbes tel que *reculer* ou *engloutir* incite à personnifier le sujet et donc à établir un contrôle. Si l'on veut trancher la question : "Cela correspond-il à des actions ?", il faut préalablement effectuer le jugement : "Jusqu'à quelle mesure peut-on considérer que *la mer* ne puisse exercer de contrôle, ou bien puisse être personnifiée ?" C'est avec de tels exemples que l'on peut percevoir et surtout prendre plaisir au flou lié à la sémantique d'une langue naturelle.

17. Dans cette phrase, on peut inférer l'existence d'une entité qui contrôle l'action de *donner un coup de poing* mais ceci ne fait pas de *recevoir* un verbe d'action à notre sens.

3. TELEO¹⁸ : viser la situation finale et orienter le changement pour qu'elle soit atteinte.

On ne saurait concevoir d'entité contrôlante sans entité contrôlée, même si les deux peuvent parfois coïncider¹⁹. C'est par exemple le cas dans *Jean marche*. On définit les notions purement sémantiques d'*agent* comme entité contrôlante et de *patient* comme entité contrôlée. Dans le cadre que nous nous sommes donné, l'agent sera toujours l'ordinateur auquel l'utilisateur s'adresse²⁰ et les patients potentiels figureront toujours parmi les objets du monde de l'application.

3.4 Dialoguer avec l'utilisateur

Le système ne dispose pas nécessairement de tous les éléments qui lui assureraient une autonomie dans la réalisation des trois grandes fonctionnalités que nous avons développées ci-dessus. L'utilisateur intervient à différents niveaux, dans les cas où le système ne peut pas par lui-même reconnaître les divers éléments qui composent un énoncé, le comprendre entièrement ou gérer la tâche à effectuer. Chacune des composantes qui rend compte de ces fonctionnalités doit ainsi permettre de maintenir une interaction avec l'utilisateur.

3.4.1 Gestion du canal de communication

Il est important de pouvoir reposer sur un message correctement identifié. Or, c'est au niveau de la reconnaissance vocale qu'il y aura le plus de sources d'erreurs. Si l'on compare le traitement du langage écrit avec celui du langage oral, on constate dans une majorité d'algorithmes concernant l'écrit, qu'une phase de vérification suit une phase de construction. Cela est dû au fait qu'il n'y a aucune incertitude sur l'énoncé écrit (caractères ASCII). Il en va tout autrement pour le langage oral, de nature évanescence qui comporte des ambiguïtés au niveau de l'énoncé émis. Cela est d'autant plus problématique que la voix constitue le support du message et que le processus de reconnaissance est situé en amont de la chaîne de traitements.

Dans les relances effectuées à l'utilisateur, il faut en outre éviter de tomber dans un puits sans fonds, où par exemple l'utilisateur s'obstine dans une mauvaise direction et la machine répète sempiternellement :

« Je ne peux pas prendre en compte votre message... »

Dans le cas d'une erreur globale de reconnaissance, il serait préférable, selon les contextes, d'effectuer les relances suivantes :

– « Votre message est inaudible, veuillez parler plus fort. »

ou si le niveau sonore est suffisant :

18. du grec *telos* : finalité.

19. On parle alors de *self-contrôle*. Il s'agit là d'une notion purement sémantique, se situant sur un plan radicalement différent de la réflexivité, qui ne constitue qu'un simple indice grammatical (pour reprendre le vocabulaire de [Langacker, 1991])

20. Cela sous-entend qu'au cours de la communication, l'utilisateur personnifie l'ordinateur.

3. Les fonctionnalités de l'interface

- « Votre message est mal formé, veuillez le reformuler autrement. »

De cette manière, l'utilisateur peut en déduire que le problème n'est pas un problème de compréhension ou d'exécution d'une commande, mais bien un problème de reconnaissance.

Lorsque la reconnaissance est partielle sur certaines portions de l'énoncé, il est important de ne pas se lancer dans un dialogue interminable. [Pierrel, 1987] en donne un exemple :

- U: Je voudrais parler à M. Durand.
- S: Vous voulez parler à M. Dupont?
- U: Non, je veux parler à M. Durand.
- S: Vous avez bien dit M. Dupont?
- U: ...

Une solution est d'opter pour une possibilité et de considérer que c'est la bonne du moment que l'utilisateur ne conteste pas ce qu'a compris ou ce qu'exécute la machine. Il s'agit là d'une manière de reposer sur une couche de dialogue située à un niveau plus haut (interprétation, gestion de la tâche). Enfin, dans le cas où le système ne peut trancher, et qu'un retour vers l'utilisateur s'avère nécessaire, il faut veiller à ce que l'utilisateur ne dévie pas du domaine de compétence linguistique du système [Junqua et Morin, 1993]. La manière dont est orienté le dialogue joue à cet égard un rôle fondamental.

3.4.2 Communication motivée par des difficultés de compréhension

Lorsque les différents mots qui composent un énoncé ont pu être reconnus, on est pas assuré pour autant d'aboutir à un énoncé compréhensible, que la reconnaissance utilise ou non des règles syntaxiques²¹. Par exemple, même si l'énoncé suivant est bien formé, il est asémantique :

(1.14) * Imprime cette image en deux colonnes.

En effet, alors qu'on peut définir les colonnes d'un texte, on ne peut pas définir les colonnes d'une image. Dans ce cas, la compréhension de l'énoncé échoue à cause d'une incompatibilité sémantique de *en deux colonnes* par rapport à l'impression d'une image. Si l'on veut réaliser des systèmes efficaces d'aide à l'utilisateur, il est nécessaire de retourner à l'utilisateur pourquoi son message n'a pas été compris. Dans l'exemple précédent, ce pourrait être :

(1.15) Une image ne peut pas être imprimée avec des colonnes. . .

21. La reconnaissance peut en effet supposer des contraintes syntaxiques (fournies par des grammaires formelles ou des modèles stochastiques concernant l'enchaînement des mots)

3.4.3 Communication motivée par la tâche à réaliser

La prise d'initiative au niveau de la conduite du dialogue peut jouer pleinement lorsqu'il n'y a aucune tâche en cours de réalisation de part et d'autre. On peut alors envisager différents cas, selon l'aptitude de la machine à conduire le dialogue. Ou bien elle est passive et attend que l'utilisateur fasse preuve d'une activité comportementale (langagière ou motrice). Ou bien elle est active et suscite une activité comportementale en posant une question du type « Que dois-je faire? ». On peut ainsi considérer, de façon orthogonale à la typologie des applications, l'aptitude à conduire le dialogue. La machine active propose à l'utilisateur de choisir parmi un ensemble de tâches qu'elle ne peut (ou qu'elle ne veut) réaliser elle-même. Mais, dans la grande majorité des systèmes de dialogue de commande, la prise d'initiative au niveau du dialogue reviendra à l'utilisateur. L'idéal serait que les deux protagonistes guident autant l'un que l'autre le dialogue. Pratiquement, c'est celui qui est le moins passif qui effectue spontanément la prise d'initiative pour lancer le dialogue.

Mais il existe aussi des cas où la prise d'initiative est suscitée, notamment lorsque les moyens pour remplir une condition ne sont pas tous réunis, ce qui provoque un blocage dans l'élaboration de la tâche. Cela peut être dû à différentes raisons :

- il manque une donnée que la machine ne peut pas inventer ;
- la machine n'a pas la compétence pour accomplir une sous-tâche ;
- la machine n'a pas « carte blanche » pour accomplir certaines sous-tâches.

Dans tous les cas, il faut donc une intervention extérieure pour pouvoir relancer l'activité d'ensemble. Pour ce faire, les raisons du blocage doivent être extériorisées sous la forme d'un message, ce qui correspond à une activité de dialogue. Selon la nature de la condition à vérifier, le message peut revêtir deux formes différentes, requête ou prescription de tâche. Une requête sera typiquement une question de type confirmation, par exemple, « Êtes-vous bien sûr de vouloir détruire ce fichier ? ». Mais cela pourra également concerner la donnée d'un paramètre manquant. Une prescription de tâche pourra être suscitée pour diverses raisons, suivant les applications. Cela pourra être dû à l'absence d'accès à l'environnement applicatif, à une délégation de tâche pour éviter la surcharge. Dans tous les cas, le but sera de réaliser la condition que la machine ne peut pas (ou ne veut pas) réaliser par elle-même.

Enfin, il est important que le système effectue un retour vers l'utilisateur pour lui assurer que ses messages ont bien été pris en compte. S'il s'agit d'une commande correspondant à une action visible, l'exécution de cette action peut constituer à elle-seule un retour vers l'utilisateur. Cela pourra correspondre à une resynchronisation avec l'utilisateur lorsqu'interviennent des problèmes de communication, ou encore à une aide contextuelle.

4 Conclusion : formulation du problème

Dans les différentes fonctionnalités que nous venons de décrire, c'est l'interprétation de l'énoncé par rapport à ce qui est à faire qui va retenir le plus notre attention. Selon

4. Conclusion : formulation du problème

les aspects que l'on privilégie dans la définition de l'action, on est amené à formuler le problème de façons différentes. Si l'utilisateur vise une activité du système à travers les énoncés qu'il prononce, c'est la manifestation de cette activité qui lui importera. L'action est toutefois davantage que la manifestation d'une activité. Elle inclut la notion de motivation telle qu'elle est présente dans la notion de tâche [Searle, 1983]. Plus qu'une vision duale, il s'agit d'une vision intercomplémentaire de l'action, puisqu'à la fois la tâche guide l'activité, et l'activité réalise la tâche. Selon l'aspect que l'on vise en premier lieu, on peut formuler le problème de deux manières différentes.

La formulation traditionnelle : La formulation traditionnelle privilégie l'activité, en allant droit au but. Dès lors, la question est celle de l'identification de l'outil (une fonction de l'application) à appeler. Si l'on fait le lien avec les données du problème, la question devient :

Étant donné un énoncé de commande, à quel outil réfère-t-il et sur quels objets doit-on l'appliquer ?

Cette approche constitue ce que l'on pourrait appeler le modèle élémentaire de référence aux actions, que nous décrirons dans le chapitre 2. Nous montrerons que cette approche se révèle insuffisante dès que l'on veut inclure un minimum de naturel dans l'expression des commandes.

Une autre formulation possible : L'autre formulation s'analyse en termes de prospective, se montrant ainsi compatible avec la notion de tâche. La question qu'elle pose est alors la suivante :

Quelle(s) modification(s) des objets mentionnés cherche à provoquer l'énoncé de commande ?

Sous cet angle, la correspondance entre séquences d'énoncés et outils de l'application s'effectuerait à un niveau encore plus abstrait. C'est l'approche que nous prônerons et que nous défendrons dans le chapitre 3.

Chapitre 2

L'identification des actions dans les systèmes classiques de dialogue

Sommaire

1	Le modèle de l'application sous-jacent	56
1.1	La nécessité de recourir à des préconditions d'utilisation	56
1.2	Représentation de l'état de l'application	56
1.3	Le modèle de l'application : définitions de base	58
2	Description du modèle « classique » de référence aux actions	59
2.1	Présentation générale	59
2.2	L'analyse structurale de l'énoncé	62
2.3	La structure prédicative de l'énoncé	69
2.4	La mise en correspondance avec une fonction	76
2.5	Récapitulatif	84
3	Les limites concernant l'adjonction de naturel	85
3.1	Un objectif à atteindre : une certaine régularité langagière	85
3.2	Une solution (trop) complexe	86
4	Conclusion	90

Après avoir exposé le modèle de l'application sur lequel s'appuie toute conception de la référence aux actions, nous présenterons l'approche classique de référence aux actions et établirons en quoi les systèmes existants de dialogue naturel de commande s'y ramènent effectivement. Nous mettrons alors en évidence l'ensemble des limites inhérentes à ce type d'approche.

1 Le modèle de l'application sous-jacent

Plaçons-nous dans l'hypothèse où un énoncé de commande a été reconnu et où l'on a pu déterminer par un moyen quelconque la fonction à appliquer. Même dans ce cas, il n'est pas question de déclencher aussitôt cette fonction : il faut recourir à des préconditions.

1.1 La nécessité de recourir à des préconditions d'utilisation

La première raison, inhérente aux systèmes de reconnaissance vocale, est qu'il est possible qu'un autre énoncé ait été reconnu à la place de celui que le locuteur transmet. Il faut donc pouvoir envisager de confirmer les commandes que l'on prononce. Dans une perspective de fiabilité différente, on ne peut se permettre de provoquer une erreur d'exécution de l'application lors de l'appel d'une fonction. Il s'avère donc nécessaire de disposer de garde-fous pour s'assurer de l'applicabilité des fonctions. Cela correspond à ce que l'on appelle classiquement des préconditions. Les préconditions d'utilisation des fonctions s'appliquent aux objets de l'application tels qu'ils sont observés. De ce fait, il n'y a pas d'accès direct aux objets mais seulement un accès à travers la représentation que s'en construit l'interface. Par conséquent, il faut pouvoir disposer d'une modèle de l'état de l'application.

1.2 Représentation de l'état de l'application

L'état de l'application peut être considéré comme un arrêt sur le déroulement du monde de référence que constitue l'environnement. La question est alors de savoir comment est perçu ce monde. Nous pouvons illustrer ceci par une métaphore de notre propre perception du monde. Lorsqu'on perçoit, par combinaison de nos cinq sens, le monde qui nous entoure, on pourrait croire que l'on en saisit une portion dans sa totalité. Pourtant, certains animaux en ont une perception bien différente : pour peu que l'on essaie de se mettre dans la peau d'une abeille, voilà qu'il faut reconsidérer tous ses sens, voire s'en imaginer d'autres ! Il faudrait ainsi utiliser la courbure de la polarisation de la lumière pour s'orienter par rapport au soleil même lorsque celui-ci est caché, ou encore se servir de l'orientation du champ magnétique terrestre pour confectionner des alvéoles de ruche bien alignées les unes par rapport aux autres. Ainsi, dans l'impossibilité d'atteindre une vision objective du monde, nous ne pouvons nous en constituer que des représentations partielles.

Si maintenant l'on considère le monde informatique qui est celui de l'application, on peut se poser la question des modalités de *perception artificielle* relativement à l'interface. Cette perception a alors pour motivation la mise à jour du modèle de l'état de l'application. On peut envisager d'observer l'application suivant deux approches, selon le rôle que l'on fait jouer à la perception. La première approche consiste à observer directement l'application alors que la seconde revient à en simuler le fonctionnement, sans en avoir de perception directe.

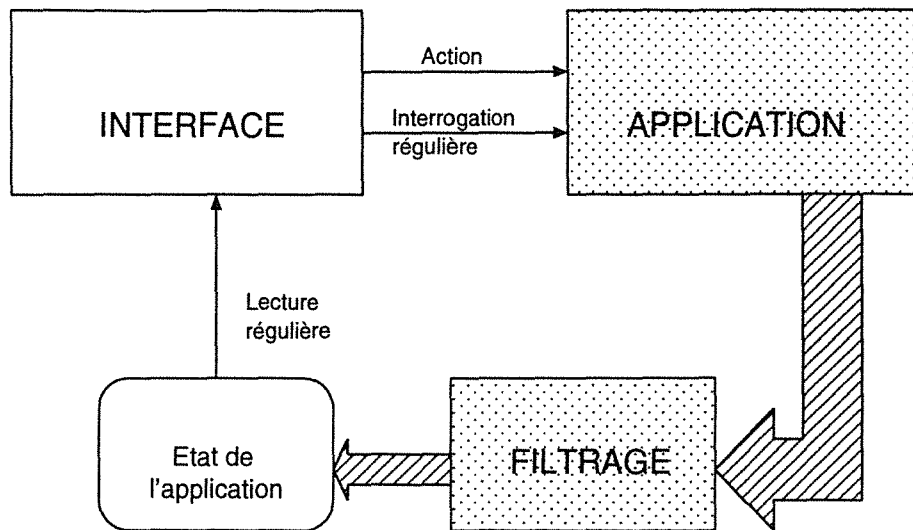


FIG. 2.1 - Approche 1 : observation de l'application

1.2.1 Une observation directe

En quelque sorte, l'observation directe revient à définir les *sens artificiels* minimaux que devra posséder l'interface. Pour pouvoir interpréter les résultats d'une observation, il est nécessaire de procéder à une discrétisation de ce qui a été observé. On passe ainsi du domaine du continu à celui du discret, pour en arriver à ce que des valeurs qualitatives représentent des valeurs quantitatives [Sandewall, 1992]. Concrètement, cela se traduit par des interprétations partielles (valeur de vérité pour des relations) et un ensemble de formules logiques (caractérisation partielle des états de l'application). La fonction d'observation introduit donc une perte d'information, ce qui correspond à un « filtrage ». [Romary *et al.*, 1991] ont montré que cette approche présente l'avantage d'être exacte mais est extrêmement dépendante de l'application et nécessite un filtrage important.

Cette approche est problématique, voire impossible à surmonter, lorsqu'il s'agit de concevoir une interface sur une application déjà existante. C'est pourquoi il est le plus souvent nécessaire d'utiliser une seconde approche, consistant à simuler l'application.

1.2.2 Une simulation de l'application

Cette approche suppose que l'on se comporte en aveugle, en s'interdisant toute perception. Dans la figure 2.2, le filtrage que produisait la fonction d'observation est remplacé par un simulateur d'effet se situant directement au niveau de l'interface. Toutefois, une simulation de l'application présente l'inconvénient de ne guère offrir de précision. Pour y pallier, il faut prévoir une boucle de régulation, de sorte qu'il n'y ait pas de divergence entre l'état réel de l'application et l'état tel qu'il est vu par l'interface. De cette manière, il devient possible de gérer des inattendus. En revanche, l'interface possède une autonomie et une indépendance par rapport à l'application. Evaluons à présent les conséquences de

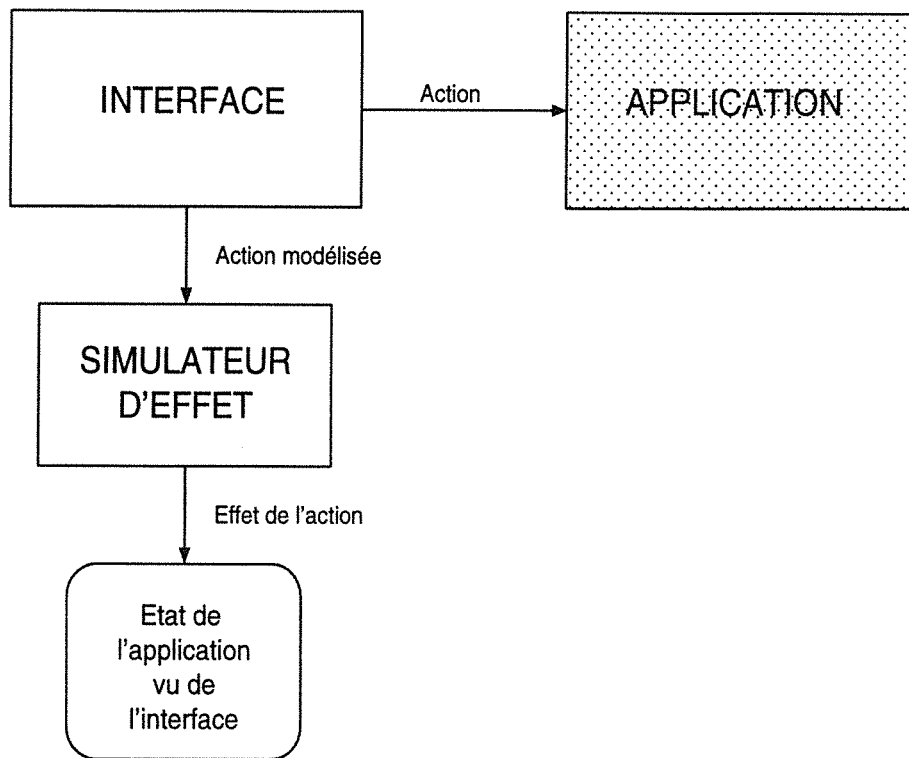


FIG. 2.2 - Approche 2 : simulation de l'application

cette approche au niveau de l'interface :

- i. La fonction n'étant pas encore exécutée, il faut pouvoir en trouver une représentation qui la simule. Il faut donc pouvoir traduire au niveau de la représentation les modifications que subiraient les objets de l'application. Il s'agit classiquement des postconditions liées à la représentation de la fonction.
- ii. Il est nécessaire de disposer de connaissances statiques (ou décrites de manière plus générale) pour pouvoir faire évoluer une représentation de l'état de l'application.

Comme nous allons le voir, ces deux aspects sont à la base de la définition du modèle de l'application.

1.3 Le modèle de l'application : définitions de base

Tout en restant situé à l'intérieur de l'interface, il faut prévoir des points d'entrée dans l'application. Ces points d'entrées correspondent à des *fonctions*¹ que peut effectuer l'application et qui sont mises à disposition *via* l'interface. Cela nous conduit à introduire un modèle de l'application, composé de deux parties :

1. un modèle dynamique de l'application, représentant l'état courant des objet ;

1. Nous considérons les procédures comme des fonctions.

2. Description du modèle "classique" de référence aux actions

2. un modèle statique de l'application, représentant l'ensemble des fonctions que peut effectuer l'application (cf figure 2.3).

Pour différencier la fonction et sa représentation, nous emploierons le terme d'*opérateur* pour leur représentation. Un opérateur est donc un modèle de fonction.

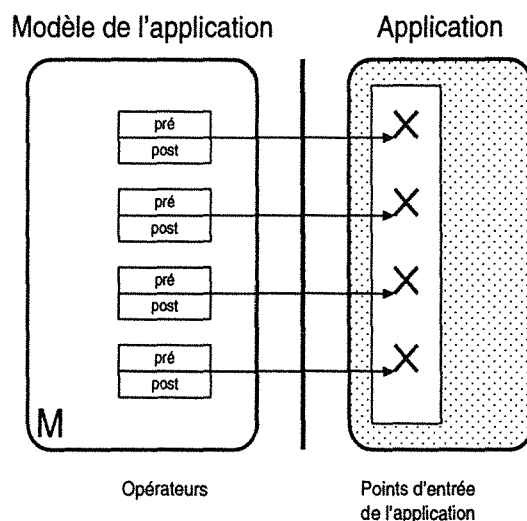


FIG. 2.3 - *Modèle statique de l'application*

De manière générale, un opérateur sera constitué de *préconditions*, de *postconditions* et d'un *corps*. Ce corps correspondra à la désignation d'une fonction de l'application. On retrouve ces divers composants à travers la règle d'utilisation que [Hoc, 1988] nous donne d'un opérateur :

« Si les conditions de validité pour appliquer une opération (une fonction, une séquence de fonctions) sont présentes et que cette opération est lancée, alors tel effet est produit. » (p.27)

C'est donc à travers les opérateurs constituant le modèle de l'application que l'on pourra accéder aux fonctions qu'ils renferment. Il faut voir là un impératif lié à la nécessité de représenter les fonctions que peut effectuer une application, mais ne présupposant en rien de la démarche de référence aux actions que l'on adopte. Maintenant que nous avons établi que l'on fera référence à des opérateurs et non à des fonctions, nous allons examiner comment l'on procède classiquement pour identifier les opérateurs (avec leurs arguments) rendant compte de l'action à exécuter.

2 Description du modèle « classique » de référence aux actions

2.1 Présentation générale

La démarche classique de référence aux actions cherche à assimiler directement une action véhiculée par un énoncé de commande à un opérateur du modèle statique des

fonctions de l'application. Le procédé général consiste alors à caractériser les éléments majeurs communs à une classe d'énoncés dans le but de mettre en évidence une structure prédicative devant correspondre avec l'opérateur à sélectionner. Une conséquence de cette approche est de dissocier complètement la référence aux actions de la référence aux objets. Par exemple, plaçons-nous dans le contexte d'une application constituée par un *shell* de système d'exploitation (UNIX dans notre cas), et considérons une interface acceptant des énoncés à consonance naturelle tels que :

(2.1) Imprime ce fichier sur la laserjet.

dans un contexte représenté par l'écran suivant (figure 2.4) :

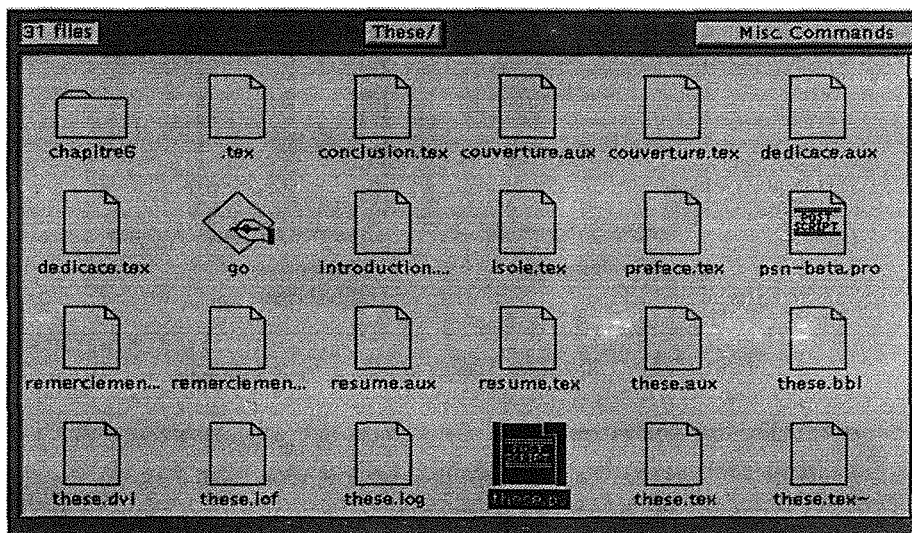


FIG. 2.4 - Ecran où s'effectue la désignation du fichier à imprimer

Une analyse classique de l'énoncé de commande va dissocier :

- le verbe « Imprime » auquel la référence aux actions associera la fonction `lpr`.
- le complément d'objet direct « ce fichier »
La référence aux objets intervient pour lui associer le nom `these.ps` (dans la situation de la figure 2.4).
- le complément prépositionnel « sur la laserjet »
La référence aux objets intervient pour lui associer le paramètre `-Plj1`.

De là, on forme l'appel de fonction à exécuter : `lpr -Plj1 these.ps`.

2.1.1 La sélection de l'action

La sélection de l'action est décisive au niveau du traitement de la référence à l'action, étant donné que la détermination de la fonction à exécuter s'ensuivra directement. Certains constituants de l'énoncé vont alors jouer un rôle déterminant au niveau de sa sélection.

Le poids du verbe. La première idée venant à l'esprit pour déterminer une action est de considérer le verbe. Ainsi, il existe dans bon nombre de systèmes une association directe entre un verbe et une action. Par exemple, [Caelen, 1991] écrit :

« La forme logique d'une commande est :

$$\text{Action}(\langle \text{arg}_1 \rangle \langle \text{arg}_2 \rangle \dots \langle \text{arg}_n \rangle)$$

Action représente une tâche élémentaire. Elle est dénotée par le verbe de la phrase. »

On retrouve le même phénomène avec les grammaires lexicales fonctionnelles où le prédicat est directement rattaché au verbe au sein de l'entrée lexicale. Mais le verbe n'est pas en général suffisant pour déterminer l'action et il est nécessaire d'effectuer un filtrage sur d'autres éléments de la phrase, en particulier l'objet direct, comme nous allons le voir dans le paragraphe suivant.

L'utilisation d'un filtrage. L'identification de la classe à laquelle appartient un énoncé s'effectue en testant le type d'éléments qui le constitue (verbe, objet direct, ...). Considérons les exemples suivants :

(2.2)

- (a) Efface ce fichier
`rm nom-de-fichier`
- (b) Efface ce répertoire
`rmdir nom-de-repertoire`

On a à chaque fois le même verbe « effacer », mais des objets de type différent, fichier dans un cas et répertoire dans l'autre. C'est ici le type du référent qui intervient dans le mécanisme de filtrage. Mais d'autres éléments de l'énoncé peuvent également intervenir dans la sélection de l'action, tels que ceux contenus dans les éléments adverbiaux (préposition, type du référent du complément nominal, ...) On aboutira donc dans ce cas général à des règles de filtrage du type :

$$\text{verbe} + \text{type(Objet-Direct)} + \text{type(Elt-Adverbial)} \rightarrow \text{Nom-d-action}$$

2.1.2 Bilan

En partant de l'hypothèse que l'action spécifiée dans l'énoncé de commande correspond à un opérateur du modèle de l'application, on assimile le problème de la référence à l'action à un problème de reconnaissance élémentaire, à savoir *reconnaître à travers l'énoncé un élément particulier dans un ensemble à N éléments (en l'occurrence 1 opérateur parmi N opérateurs possibles)*. Si l'on appelle L l'ensemble des énoncés acceptés par le système, cette conception nous amène à effectuer une partition de L en N classes (modulo une relation \mathcal{R}), chaque classe d'énoncés correspondant à un opérateur de l'application. En

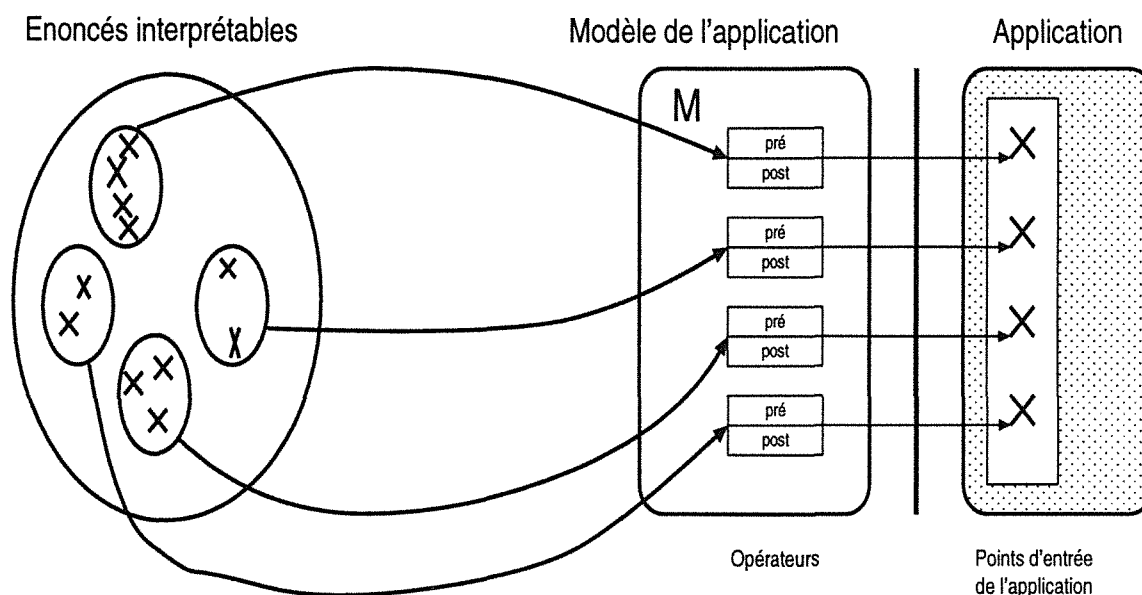


FIG. 2.5 - *Bijection classique*

termes purement algébriques, il s'agit d'une bijection de l'ensemble quotient des énoncés L/\mathcal{R} sur le modèle de l'application (figure 2.5).

La démarche traditionnelle de référence aux actions a pour caractéristique générale de faire correspondre à un opérateur (i.e. une représentation d'une fonction de l'application) une classe d'énoncés. Cela nous conduit donc à poser une relation \mathcal{R} d'équivalence d'appartenance à une classe. Au niveau du flux de données que cela introduit, cela peut se traduire par le schéma général suivant (cf. figure 2.6). Cette démarche suppose que l'on puisse distinguer d'une part le nom de la fonction et d'autre part ses arguments, qui correspondent à des objets ou à des paramètres. Pour ce faire, il est nécessaire d'analyser la structure de l'énoncé et isoler ses divers constituants.

2.2 L'analyse structurale de l'énoncé

2.2.1 Les données en jeu

Il semble naturel de considérer qu'un énoncé soit constitué d'une suite de mots (cf. figure 2.7). La notion de mot est assez vague. Nous retiendrons celle d'une entrée de dictionnaire, ou encore celle d'une chaîne délimitée par deux espaces blancs en transcription écrite. Mais cette vue simpliste de l'énoncé n'est pas satisfaisante pour la simple raison que l'on ne saurait substituer n'importe quel mot à un autre et obtenir de la sorte un énoncé grammatical.

Plutôt que de se situer au niveau du mot, on peut alors se situer au niveau du groupe de mots faisant une unité. Nous avons en effet établi dans le §2.1.1 qu'il était nécessaire d'isoler des groupes de mots (le verbe, l'objet direct, les éléments adverbiaux) et de les

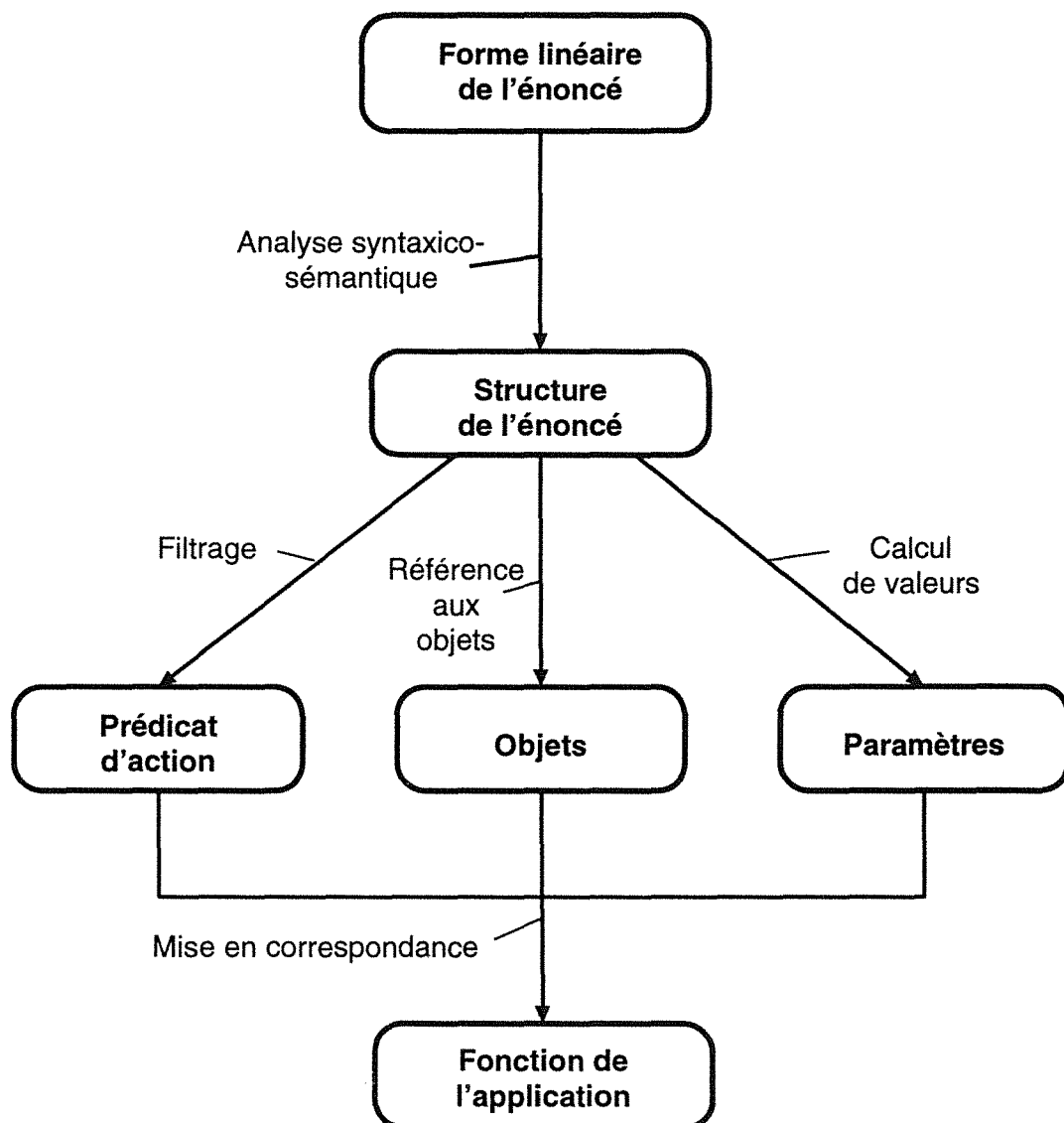


FIG. 2.6 - Flux général de données dans les approches classiques de référence aux actions

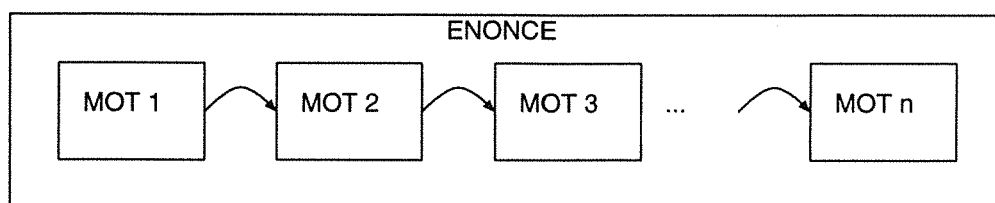


FIG. 2.7 - Une approche simpliste de la constitution d'un énoncé

caractériser par leur *fonction grammaticale* dans l'énoncé. En linguistique, cela correspond à la notion de *syntagme*. La notion de syntagme est étroitement liée à la celle de substitution : en remplaçant un syntagme par un autre de même nature, quel que soit le contexte, on est censé obtenir un énoncé grammatical. La notion de la grammaticalité telle qu'elle est présentée ici peut d'une certaine manière se concevoir comme une relation d'équivalence qui aboutit à des classes grammaticales de groupes de mots faisant unité.

Dans le cadre du dialogue homme-machine dans lequel nous nous situons, il n'est pas évident que l'on puisse considérer l'énoncé comme un syntagme *phrase* qui constituerait la racine d'un arbre syntagmatique comme le présupposent les théories structurales syntagmatiques. En effet, on ne peut définir ce qui opposerait une phrase à une autre, comme le souligne [Benveniste, 1974b] :

« Du fait que la phrase ne constitue pas une classe d'unités distinctives, qui seraient membres virtuels d'unités supérieures, comme le sont les phonèmes ou les morphèmes, elle se distingue foncièrement des autres entités linguistiques. Le fondement de cette différence est que la phrase contient des signes (unités linguistiques) mais n'est pas elle-même un signe. Une fois ceci reconnu, le contraste apparaît clairement entre les ensembles de signes que nous avons rencontrés aux niveaux inférieurs et les entités du présent niveau.

« Les phonèmes, les morphèmes, les mots (lexèmes) peuvent être comptés ; ils sont en nombre fini. Les phrases, non².

« Les phonèmes, les morphèmes, les mots (lexèmes) ont une distribution à leur niveau respectif, un emploi au niveau supérieur. Les phrases n'ont ni distribution ni emploi.

« Un inventaire des emplois d'un mot pourrait ne pas finir ; un inventaire des emplois d'une phrase ne pourrait même pas commencer.

« La phrase, création indéfinie, variété sans limite est la vie même du langage en action. Nous en concluons qu'avec la phrase on quitte le domaine de la langue comme système de signes, et l'on entre dans un autre univers, celui de la langue comme instrument de communication, dont l'expression est le discours. »

Cette envolée de Benveniste s'inscrit donc en opposition avec l'approche selon laquelle on identifierait l'emploi d'un énoncé à partir de la distribution de ses constituants. Pré-

2. Benveniste dénie de ce fait à la langue le statut de langage régulier sur un vocabulaire fini.

2. Description du modèle "classique" de référence aux actions

cisons que dans le cadre finalisé où nous nous situons, nous pouvons effectuer quelques hypothèses simplificatrices quant à la la forme que peut prendre une phrase. Si, dans le cas général, on ne peut attribuer de distribution à un énoncé, on a souvent envisagé de faire référence à un énoncé théorique « bien-formé » auquel on peut attribuer un ensemble fini de distributions. Cela ouvre une voie pour se représenter une structure de l'énoncé reposant sur des syntagmes.

La structure d'un énoncé telle que se la représente l'analyse classique correspond à une imbrication de syntagmes les uns dans les autres (syntagmes verbaux, syntagmes nominaux, propositions relatives, ...). Au bout du compte, on aboutit à des mots individuels de diverses catégories (articles, noms, verbes, adjectifs, prépositions, ...), constituant les feuilles d'un arbre dit *syntagmatique*. C'est ce que l'on appelle la structure de surface de l'énoncé. En effet, les feuilles des arbres permettent de reconstituer directement la forme linéaire de l'énoncé. C'est pour cette raison que l'on rencontre également pour cette structure l'appellation de structure de précedence linéaire [Sabah, 1988]. La structure standard d'un énoncé correspond de ce fait à une arborescence de syntagmes (figure 2.8). Nous

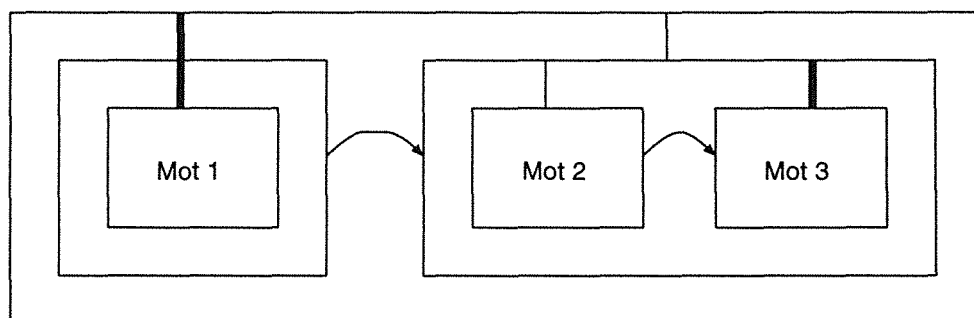


FIG. 2.8 - Une approche arborescente de la constitution d'un énoncé

allons à présent développer les différents types d'analyses qui utilisent une telle structure.

2.2.2 L'analyse syntaxique standard

L'analyse syntaxique standard d'un énoncé a pour objectif de décomposer cet énoncé en constituants plus petits imbriqués les uns dans les autres. Les différentes manières de construire des phrases à partir de mots sont décrites par les règles d'une *grammaire*. Celles-ci s'interfacent avec un *lexique* spécifiant les mots élémentaires du langage (avec leurs propriétés syntaxiques) et un ensemble de *règles morphologiques* décrivant la façon dont les mots peuvent être dérivés les uns des autres. La morphologie et la grammaire peuvent occuper des proportions très variables d'une langue à une autre.

La manière la plus simple de décrire les structures de surface est d'utiliser des *règles de réécriture à contexte libre*, comme on le fait pour décrire les langages de programmation. On peut donner un exemple simple d'une telle grammaire³ pour une application de type commande :

$$S \rightarrow \text{Verbe.SN.SP}^*$$

3. Les conventions de notation sont les suivantes : \rightarrow est la relation de production, le $.$ indique une séquence et $*$ désigne l'étoile de Kleene (indiquant que le symbole peut figurer 0 fois ou plus).

SN → Det.Adj*.Nom.Adj*.SP*

SP → Prep.SN

Ci-dessous est représentée la grammaire d'une application de dessin (ICP-Draw) [Caelen, 1991]. Les conventions de notation sont les suivantes : → est la relation de production, les " " indiquent des terminaux, le . indique une séquence et le | indique une disjonction de règles.

Action → Pred.NP.Location

Reiter → NP.Location

Rectif → "no".NP.Location | "more".AdjT

NP → Det1.AdjT.N.AdjC

Pred → "draw" | "move" | "erase" | "change" | "undo" | "select" | "duplicate" | "quit"

Det1 → "the" | "one" | "two" | "this" | "these" | etc...

AdjT → "large" | "small"

N → "square" | "circle" | "triangle"

AdjC → "white" | "black" | "blue" | etc...

Location → PP | LocP | LocA

PP → Prep.Det2.N.AdjC

Prep → "under" | "on" | "on this"

Det2 → "the" | "this"

LocP → "to the right" | "to the left" | "above" | "under" | "in the center" | etc...

LocA → "here" | "there" | "over there" | "towards here" | "around here" | etc...

Pour construire la structure de l'énoncé, il faut bien sûr recourir à un analyseur destiné à traiter ces règles. Les systèmes existants ont trouvé des raffinements et des variations par rapport à ces règles de réécriture syntagmatiques, tout en conservant les mêmes principes de base. En voici quelques exemples.

2.2.3 Les grammaires transformationnelles

Elles ont été proposées par [Chomsky, 1957] [Chomsky, 1965]. Elles utilisent une grammaire syntagmatique pour définir les *structures profondes* canoniques des phrases en langage naturel. La *structure de surface* effective est dérivée en déplaçant les constituants d'après leur position canonique par application d'une séquence de transformations sur ces structures profondes. Pratiquement, on utilise un algorithme à deux étapes. La première analyse la structure de surface de la phrase d'entrée en se basant sur une grammaire à contexte libre trop large (admettant des formes de phrases qui ne sont pas autorisées). Elle est suivie par une composante transformationnelle qui transforme les structures de surface en leurs structures profondes sous-jacentes. Cette étape peut ainsi être assimilée à un filtre. La structure profonde ainsi obtenue est considérée comme l'analyse de la phrase. Grossièrement, il s'agit d'une structure prédicative linguistique.

Les grammaires ATN (Augmented Transition Network) Elles ont été développées par Woods [Woods, 1970] dans le but de rendre efficaces les grammaires transformationnelles sur le plan informatique. Une grammaire ATN revient à spécifier à l'aide d'un automate un algorithme d'analyse non-déterministe. On définit les analyses correctes en donnant les différents chemins à travers un réseau de transitions récursif (RTN pour *Recursive Transition Network*). Ces réseaux consistent eux-mêmes en une série d'états (représentés par des cercles), connectés entre eux par des arcs. Des étiquettes associées aux arcs peuvent correspondre à des terminaux ou des non-terminaux de la grammaire. Dans le premier cas, on s'en tient là. Dans ce dernier cas, on réfère à un sous-réseau de la grammaire. La figure 2.9 montre un exemple simple d'une grammaire de commande. On

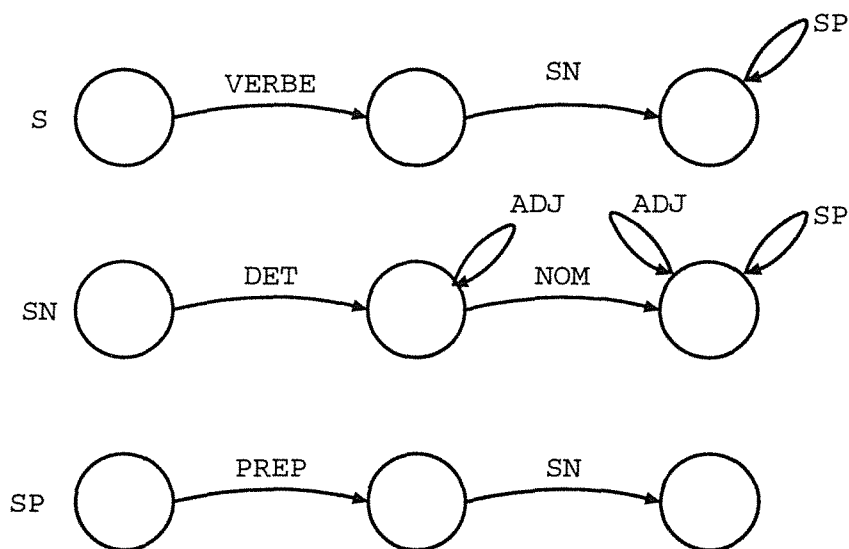


FIG. 2.9 - Exemple de grammaire RTN. Terminaux : VERBE, DET, ADJ, NOM, PREP. Non-terminaux : SN, SP, S.

peut enrichir cette grammaire en augmentant les arcs de conditions et d'actions sur des registres qui stockent des informations auxiliaires. On arrive alors aux ATN. Ces actions confèrent à la grammaire un aspect procédural qui a été exploité dans de nombreux systèmes. Dans notre cas, cela pourrait même aller jusqu'au calcul de l'opérateur à appeler.

2.2.4 Les grammaires sémantiques

Les grammaires sémantiques ont été introduites par [Burton, 1976] dans le but d'intégrer des contraintes sémantiques et pragmatiques à l'intérieur de la grammaire. Les symboles non-terminaux de la grammaire correspondent à des classes sémantiques (regroupements d'objets ayant un sens commun), qui sont définies par les concepteurs à partir du domaine à traiter. Les grammaires sémantiques utilisent la technique des règles de réécriture. Cette technique est bien maîtrisée car elle repose sur les mêmes travaux que ceux effectués pour les compilateurs. La syntaxe est implicite, étant donné qu'elle correspond à l'ordre dans lequel les constituants sémantiques sont susceptibles d'apparaître. Pour reprendre les termes de G. Sabah, il s'agit d'une façon pragmatique de construire

des programmes efficaces.

Bien souvent, la connaissance de l'ordre des constituants est suffisante pour comprendre et interpréter convenablement un message. D'après [Pierrel, 1987], certaines applications de dialogues oraux homme-machine nécessitent et n'utilisent que des langages artificiels restreints (à consonance plus ou moins naturelle) qui peuvent être entièrement décrits par leur syntaxe (commande de processus industriels, par exemple) et définis par une grammaire à contexte libre ou par un automate dans lequel les terminaux correspondent aux mots ou aux groupes de mots du langage. Ils ne réclament aucun traitement sémantique particulier, dans la mesure où l'on peut définir une relation biunivoque entre une structure syntaxique (fournie sous la forme d'un arbre syntaxique, par exemple) et une interprétation de la requête ou de l'action correspondante.

2.2.5 Les grammaires à base de caractéristiques

Le besoin de rajouter des caractéristiques aux constituants s'est vite fait sentir pour rendre compte de phénomènes d'accord et de dépendance à distance. Les grammaires à base de caractéristiques utilisent toutes une représentation syntaxique des énoncés, sous la forme d'ensembles imbriqués de structures attribut-valeur, chacune de ces structures devant être considérée comme l'étiquette d'un syntagme. Le traitement en lui-même se résume à un mécanisme d'unification de caractéristiques couplé à un mécanisme d'analyse en constituants. Aussi rencontre-t-on l'appellation de *grammaires d'unification*.

2.2.6 Grammaires lexicales fonctionnelles

La grammaire lexicale et fonctionnelle de Bresnan et Kaplan [Bresnan et Kaplan, 1981] est une théorie linguistique se prêtant à l'utilisation de formalismes à base de caractéristiques (*attribut valeur*), ce qui, du point de vue de la formalisation, place cette théorie parmi les grammaires d'unification. L'une des caractéristiques majeures des théories lexicales fonctionnelles est de privilégier la représentation lexicale des différentes catégories de mots. Une entrée lexicale sera donc identifiée par un attribut *Lex* et par la catégorie *Cat* auquel elle appartient. Outre cette fonction d'identification, on peut répertorier deux usages principaux concernant les attributs.

Un premier usage est de marquer les caractéristiques liées à telle entité lexicale. Cela peut constituer en des traits morphosyntaxiques (marques de nombre, de genre, de personne grammaticale, de mode, de temps grammatical) mais aussi en des formes logiques. Dans ce dernier cas, l'attribut *Predicate* indique le noyau prédicatif et *Xadjunct*⁴ peut optionnellement en indiquer la partie périphérique qui lui est associée. Si l'on compare les structures attribut-valeur à la représentation de forme prédicative dont il est question dans ce chapitre, on fait immédiatement le rapprochement entre d'une part *Predicate* et l'ensemble *prédicat + complément*, d'autre part *Xadjoint* et les circonstants. Par exemple, l'entrée lexicale (simplifiée) correspondant au verbe *déplacer* serait la suivante :

$$\left[\begin{array}{l} \text{Lex} = \text{Déplacer} \\ \text{Cat} = \text{Verb} \\ \text{Predicate} = \text{MOVE}(\leftarrow \text{Sujet}, \leftarrow \text{Objet}, \leftarrow \text{Oblx}) \end{array} \right]$$

4. En français, un adjectif

Un deuxième usage est de préciser la fonction (rôle) du syntagme associé (le gouverneur) par rapport à des syntagmes se trouvant à une hiérarchie inférieure (les gouvernés). Dans l'exemple ci-dessus, il s'agit des fonctions grammaticales *Sujet*, *Objet* (complément direct) et *Obj* (complément indirect). Nous discuterons dans la deuxième partie sur le procédé par lequel s'attribuent ces fonctions en cours d'analyse.

Pour assurer le lien avec le modèle de l'application⁵, des entrées lexicales spécifient une correspondance directe entre les arguments sémantiques et les configurations des fonctions grammaticales de surface. C'est ainsi à partir d'une fonction prédicative que l'on recherche en priorité les constituants qui peuvent s'y rattacher.

À partir d'une telle structure arborescente, il devient possible de caractériser les éléments d'un énoncé dans le processus de référence aux actions en passant par la structure prédicative de l'énoncé.

2.3 La structure prédicative de l'énoncé

Dans les approches syntaxiques exposées ci-dessus, l'analyse effectuée a pour but d'isoler le verbe et les éléments composant sa suite. Pour pouvoir caractériser ces éléments, il est nécessaire de déterminer le rôle de chacun d'eux par rapport au verbe. On peut alors envisager de représenter l'énoncé sous la forme d'un prédicat logique du premier ordre (nous en avons déjà eu un exemple avec les grammaires lexicales fonctionnelles) : le prédicat est issu du verbe et les arguments sont issus de la suite verbale. Un nouveau problème se pose alors : quelle est l'arité du prédicat ? Ce problème est classiquement désigné comme celui de la valence. Précisons que nous nous situons toujours à un niveau linguistique et que le prédicat en question n'a aucune raison *a priori* de correspondre au profil d'un opérateur ou de la fonction que ce dernier représente.

2.3.1 La notion de valence

La valence d'un verbe correspond au nombre de compléments qu'il nécessite [Tesnière, 1969]. En comparant les deux phrases :

(2.3) Il a posé ici son sac de ciment le moins lourd.

(2.4) Il a posé ta truelle dans l'autre pièce.

on pourrait être tenté de dire que la valence est 3 : une place pour le sujet, une place pour le complément d'objet et une autre pour le complément de lieu. Cependant, les choses s'avèrent plus compliquées. À l'opposé de verbes tels que *mettre* où le complément de lieu est obligatoire, le lieu est facultatif pour le prédicat *poser*. On peut en effet dire :

(2.5) Pose ça !

On a aussi des expressions telles que *poser des rideaux*, *poser une serrure*, *poser une voie de chemin de fer* ou *poser un principe*. On rencontre la même difficulté avec le verbe

5. Ou, de manière générale, avec une sémantique référentielle.

mettre, classiquement considéré comme trivalent (quelqu'un qui met quelque chose quelque part). Mais ne dit-on pas *mettre un vêtement* ?

(2.6) À la soirée d'adieu, j'ai mis mon plus beau costume.

On pourrait alors envisager le problème par rapport au sens du verbe, altéré qu'il est par la suite verbale qui interagit avec lui. Les liens plus ou moins tenus qu'un élément de la suite verbale peut avoir avec le verbe obéissent à des facteurs sémantiques, ce que reflètent les problèmes sous-jacents à la classification de compléments.

2.3.2 Problèmes sous-jacents de classification des compléments

Comme on le voit, il est important de classer les différents cas de surface selon leur distance par rapport au noyau verbal. L'approche que nous avons eue jusqu'à présent est de considérer comme complément ce qui est lié à la valence du verbe.

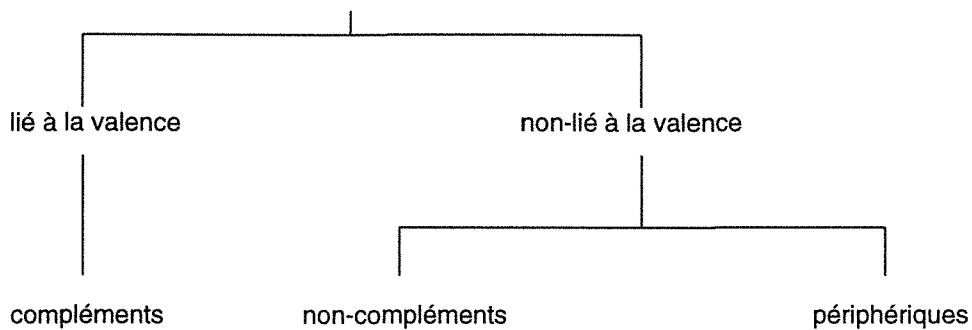


FIG. 2.10 - Première approche de la valence

Une approche complémentaire est de lier ce qui est périphérique à ce qui ne l'est pas ; on obtient alors de manière plus complète :

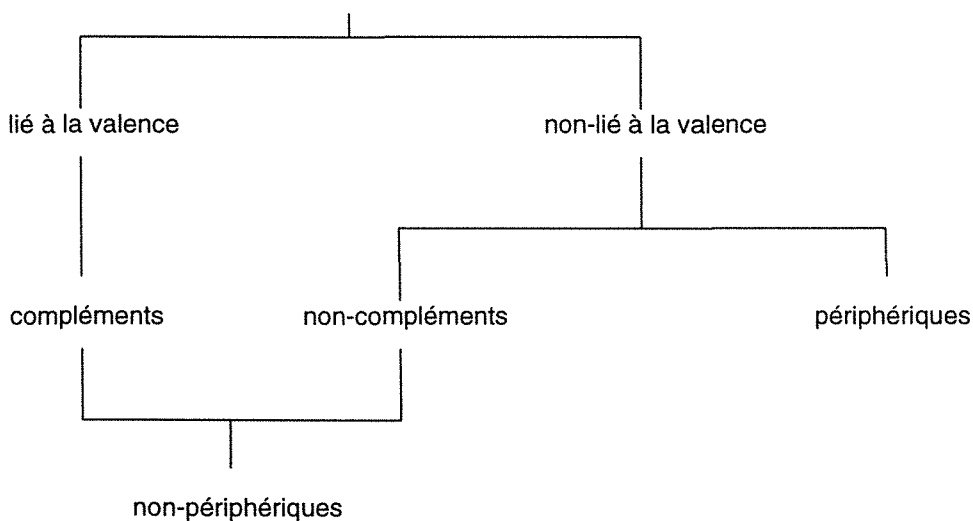


FIG. 2.11 - Approche complémentaire de la valence

2. Description du modèle "classique" de référence aux actions

Ces schémas mettent en évidence les cas de surface qui ne sont pas liés à la valence, sans pour autant être périphériques : c'est ce que nous appellerons les intermédiaires. Nous distinguons ainsi paramètres, intermédiaires et circonstants situationnels.

Une classification plus fine est proposée par Somers (*op. cit.* p. 27) qui distingue six degrés de liaison avec le noyau. Le premier type est collé au noyau, les deux suivants sont des sous-catégories du prédicat, alors que les autres lui sont externes.

1° Compléments intégraux

Ils sont déterminés lexicalement, se trouvant ainsi collés avec le noyau prédicatif. Ils ne peuvent pas être omis ni pronominalisés, et sont très exceptionnellement modifiables par un complément (par exemple *casser sa pipe*, *claquer des doigts* ou à la limite *passer le balai* auquel on peut substituer *balayer*).

2° Compléments « obligatoires »

Le cas du sujet est toujours obligatoire. Dans le cas de l'impératif, il est fourni par la situation de communication. De manière générale (mais il y a un nombre non négligeable d'exceptions, cf. [Grevisse, 1986]), les autres compléments obligatoires correspondent aux objets et objets seconds, comme par exemple :

(2.7) Imprime ce fichier.

(2.8) Envoie ce message à Laurent.

On rencontre par ailleurs des compléments d'objets obligatoires dans le cas de certains verbes (dits transitifs indirects) qui, dans un sens donné, régissent une préposition donnée pour un syntagme prépositionnel (SP), comme dans :

(2.9) Va dans le répertoire public.

3° Compléments optionnels

Il s'agit d'éléments qui font partie des arguments du prédicat, mais qui correspondent généralement à une valeur par défaut elle-même introduite par le prédicat. Par exemple, ce serait un pléonasme que de peindre un objet avec un pinceau (ou l'on aurait du moins un fort effet thématique sur le pinceau, si on l'accompagne d'un élément permettant de le déterminer). Par contre, on peut très bien peindre un objet avec un rouleau. C'est bien illustré dans le système de compréhension par frames de [Charniak, 1978]. Cela est également traité par Wilks [Wilks, 1975a; Wilks, 1975b; Wilks, 1978] qui enferme dans la formule d'un prédicat les types d'objets concernés par la définition, ce qui lui permet par exemple d'"expliquer" des verbes pouvant être intransitifs (ou que des verbes bitransitifs puissent être transitifs). Il explique alors l'optionnalité en terme de préférences. Néanmoins, il nous semble qu'il prenne le problème à l'envers. De façon similaire aux compléments « obligatoires », les compléments optionnels sont strictement subordonnés au prédicat. Les contraintes peuvent varier selon les prédicats (et les places argumentales qui leur sont liées) et se situer à différents niveaux, c'est-à-dire dans une hiérarchie décroissante :

-les cas sémantiques (instrument, destination, ...)

- les fonctions syntaxiques (nominatif, accusatif, cas obliques à la GLF)
- les formes syntaxiques (syntagme nominal, proposition infinitive, subordonnée introduite par 'que')

Exemples :

(2.10) Imprime ce document en recto-verso.

(2.11) Imprime ce document en deux colonnes.

(2.12) Imprime ce document sur la laserjet de la salle console.

4° Eléments intermédiaires

Ces éléments ne sont pas prévisibles pour un prédicat donné, mais seulement de la classe générique dont ce prédicat est une instance particulière. Ils ont ainsi à la fois des propriétés de (3) et (5). D'où, pour les syntacticiens, le caractère capricieux des éléments compris entre (3) et (5).

Exemples :

(2.13) Imprime ce document en deux exemplaires.

(2.14) Imprime ce document dans la salle console.

5° Eléments accessoires (ou périphériques)

Ces éléments sont moins restreints que les précédents en ce qui concerne leur place (i.e. l'ordre des syntagmes) dans la proposition. Ce sont des périphériques : ils ne sont pas subordonnés au prédicat. Leur présence est (plus ou moins) complètement imprévisible à partir du prédicat ou de leur classe générique. Ils correspondent à des circonstants d'évaluation de la situation (typiquement à des circonstants de lieu et de temps, mais ce ne sont pas les seuls). Ils peuvent de plus se rencontrer avec des compléments du verbe du même type sémantique.

Exemples :

(2.15) Imprime ce document dans cinq heures.

(2.16) Imprime ce document [dans la nuit/cette nuit].

6° Eléments extra-périphériques

Ils ne sont pas du tout contraints dans la proposition, tant du point de vue de la place où ils peuvent apparaître que du point de vue sémantique. Les seules contraintes sont celles du discours. Ils modifient la proposition entière, éléments accessoires compris, et sont typiquement des modifieurs logiques ou discursifs (*notamment, d'habitude, ainsi, comme vous savez,...*). Nous ne traiterons pas ce type d'élément, étant donné que les effets rhétoriques sont rarissimes dans le cadre du dialogue homme-machine.

Si l'on veut s'en tenir à une arité fixe, il faut trouver des valeurs par défaut associées à chaque argument de prédicat d'un rang donné. Dans le cas d'*imprimer*, il peut s'agir des valeurs suivantes : en recto, sur une colonne, en un seul exemplaire, sur l'imprimante par défaut. On s'aperçoit cependant qu'il est difficile de fixer un nombre donné d'arguments et de les spécifier tous. L'important est de spécifier ce qui est donné par l'énoncé.

Charniak et MacDermott [Wilensky, 1991] ont proposé une stratégie pour pallier ce problème. Il s'agit d'un style de représentation appelé *slot-assertion notation*, qui s'apparente directement aux *frames*. Elle s'appuie sur une approche fondée sur la situation comme celle introduite par [Reichenbach, 1947] et [Davidson, 1967]. Elle consiste à introduire un argument représentant un événement (e), et d'autre part des prédicats pour typer et exprimer des relations.

(2.17) Imprime ce fichier en recto-verso.

2.17 serait ainsi analysé par la formule suivante :

$$\exists e(\text{IMPRIMER}(e) \wedge \text{Objet}(\langle \text{fichier83} \rangle, e) \wedge \text{En}(\text{recto} - \text{verso}, e))$$

Cette formule logique peut se mettre directement sous la forme suivante, où chaque slot du schéma correspond à une relation, obligatoire ou facultative :

$$\left[\begin{array}{l} \text{predicat : Imprimer} \\ \text{objet : fichier83} \\ \text{en : recto-verso} \end{array} \right]$$

On met ainsi en évidence le prédicat et ses relations casuelles.

2.3.3 Prédication et relations casuelles

Une prédication est une structure constituée d'un prédicat et d'un nombre adéquat de termes fonctionnant comme arguments de ce prédicat. Ces termes sont spécifiés au moyen de fonctions sémantiques, ou *cas*.

Les grammaires de cas⁶ considèrent qu'une phrase est constituée d'une modalité s'appliquant à une proposition de la forme (*Predicat Cas₁ Cas₂ ... Cas_N*) où le prédicat est typiquement un verbe. La modalité est un terme qui est employé ici abusivement pour rendre compte à la fois :

- de la prise en charge de l'énoncé par l'utilisateur (avec les notions de négation, de mode, d'aspect et de temps grammatical) ;
- de la localisation de l'énoncé dans une situation particulière (avec les notions de temps cognitif, de lieu, de circonstants d'évaluation).

Les cas qui sont contenus dans la proposition sont attachés *a priori* au prédicat. Ils indiquent pour ses différents arguments le type de lien qui les lie au prédicat. Les cas que

6. Les grammaires de cas ont été la résurgence formelle de théories linguistiques (par exemple [Tesnière, 1969]) qui prônaient l'organisation fonctionnelle et dynamique d'actants autour d'un procès dans le cadre d'une conception scénique de l'énoncé. Elles ont été propulsées à la suite de l'article célèbre de Fillmore *The case for case* [Fillmore, 1968]. Depuis, de nombreux autres modèles de grammaires casuelles ont été proposés.

l'on pourrait qualifier de standard, dans le sens qu'ils sont classiquement rencontrés dans les différentes grammaires dites *de cas*, sont les suivants :

- **Agent** : qui fait l'action?
Ex: **Jean** pose le livre sur la table.

- **Patient** : qui subit l'action?
Ex: Pose le **livre** sur la table.

- **Destination** : vers où se dirige l'action?
Ex: Pose le livre **sur la table**.

- **Source** : d'où provient l'action?
Ex: Sors le livre **du tiroir**.

- **Bénéficiaire** : qui bénéficie de l'action?
Ex: Donne le livre **à Jean**.

- **Manière** : comment s'effectue l'action?
Ex: Pose **délicatement** le livre sur la table.

Cela suscite un certain nombre d'interrogations. D'abord, il n'y a pas d'universalité d'un tel ensemble de types de liens [Rastier, 1991] et la question de leur choix demeure ouverte. Ensuite, pour un verbe donné, le nombre de cas qu'on peut lui associer n'est pas fixé.

La question du choix des relations casuelles. La définition des cas linguistiques fait appel à des notions purement sémantiques directement relatives à une théorie de l'action. Nous avons déjà vu que la notion d'action était très spécifique à certaines classes de procès, impliquant le contrôle d'une entité par rapport à l'évolution d'une situation. C'est seulement dans un tel cadre que l'on peut parler d'*agent* en tant qu'entité contrôlante et de *patient* en tant qu'entité contrôlée. Or, c'est là pure affaire de jugement, où interviennent quantité de connaissances extralinguistiques sur les entités elles-mêmes. Dans une phrase telle que *La mer se recule lentement*, il y a évolution de situation. Peut-on considérer pour autant qu'il y ait contrôle de la mer? Nos connaissances sur le monde nous incitent à répondre par la négative, mais on ne saurait véritablement trancher, car la langue peut nous conduire à réexaminer les entités sous un autre angle, comme dans :

(2.18) La mer a englouti un pêcheur.

2. Description du modèle “classique” de référence aux actions

Le verbe *engloutir*, de par ses propriétés sémantiques, incite à personnifier le sujet. Si l'on accepte de personnifier la mer, lui conférant ainsi un contrôle sur l'action d'engloutir quelqu'un, cela revient à considérer la phrase comme dénotant une action.

Il n'y a donc pas universalité des cas linguistiques car leur définition doit prendre en compte deux facteurs interdépendants :

1. se délimiter par rapport à une classe sémantique de procès, dans notre cas les classes d'action ;
2. mettre en jeu des connaissances extra-linguistiques⁷ sur les entités impliquées lors d'un procès afin de jauger leur agentivité (ou, ce qui revient au même dans notre cas, leur intentionnalité).

À cet égard, il est révélateur de considérer des applications de commande. Winston, dans l'analyse qu'il fait de SHDRLU, concède que l'analyse du rôle de chaque actant dans la configuration exprimée par l'énoncé se trouve extrêmement guidée par la tâche⁸ :

- Agent : SHDRLU.
- Bénéficiaire : le maître qui commande le système.
- Instrument : le bras manipulateur de SHDRLU.
- Objet : Un objet du monde. des blocs (cube, pyramide,...) auquel un groupe nominal (sans préposition) réfère.
- Lieu de destination : Implicite ou explicité par un objet (ou plutôt un sous-espace...) du monde des blocs (table + cube, pyramide,...) auquel un groupe nominal (précédé d'une préposition) réfère.

Cela n'est pas un reproche en soi dans la mesure où les méthodologies mises en œuvre s'avèrent suffisantes pour traiter correctement le sous-ensemble du langage naturel qui se trouve être impliqué. Mais le problème se pose dès lors que l'on définit une application un tant soit peu plus complexe, les mêmes méthodologies pouvant se révéler insuffisantes et mettre en évidence les limites du système. Ce qui fait dire à Winston :

« Indeed there is so much constraints in the block world, the problem of sorting the noun groups into cases is not a significant task for most sentences. »

En ce qui concerne les rôles sémantiques (Agent, Instrument, qui se trouvent être ici des valeurs par défaut), nous ne trouvons pas de différence majeure avec d'autres systèmes de commande orale d'une plus grande portée applicative. Mais il en va tout autrement pour les autres rôles sémantiques. Au niveau de l'objet, nous avons affaire à des types différents qui se plient différemment à l'exécution d'une action donnée. En outre, les seules actions qui sont ici modélisées concernent le changement de place et le mouvement. Dans

7. Qui peuvent être remodelées en partie dans des romans de science fiction ou des univers imaginaires.

8. Lorsqu'on fixe une limite à un domaine, cela se traduit par de nouvelles propriétés : ou bien on les utilise (monde fermé) ou bien on ne les utilise pas (monde ouvert).

des applications plus complexes, il faudrait faire face à des transformations, des créations ou des suppressions d'objets, ce qui amènerait à revoir cette modélisation des actions. Reposer sur un domaine d'application très réduit peut ainsi se révéler une arme à double tranchant.

2.4 La mise en correspondance avec une fonction

2.4.1 L'arbitraire de l'en-tête de fonction

On pourrait être tenté de rapprocher un en-tête de fonction à la structure prédicative issue d'un énoncé. Tous deux font appel à un nom de prédicat et à des arguments. Klein [Klein, 1990] en note le fort appariement :

« La sémantique est définie à l'aide d'une grammaire de cas dont la structure (verbe + cas) s'apparente fortement à la structure logique d'une commande (prédicat + arguments). »

Mais il existe toutefois une différence de taille. La structure prédicative d'un énoncé reflète la structure de la langue à travers la sémantique naturelle de l'action (rôles sémantiques), alors que l'en-tête d'une fonction reflète la sémantique formelle de l'application (arguments de fonction). Il n'y a donc aucune raison *a priori* pour que l'on puisse effectuer une superposition entre deux données aussi indépendantes, la langue à consonance naturelle d'un côté et l'application informatique de l'autre.

Nous avons pu voir que les fonctions ne font pas de différence entre objets et paramètres : les uns et les autres correspondent à des arguments sans aucune distinction. Un rôle sémantique peut ainsi correspondre soit à :

- un argument de la fonction seulement ;
- un critère de sélection du nom de la fonction seulement ;
- les deux.

Par exemple, à l'énoncé

(2.19) Imprime ce fichier en deux colonnes.

peut aussi bien correspondre l'en-tête de fonction

```
PRINT(Fichier, NbCol) avec NbCol=2
```

que l'en-tête

```
PRINT-IN-2-COLUMNS(Fichier).
```

On constate ainsi que, pour une fonctionnalité donnée, les en-têtes de fonction sont assez arbitraires⁹. Dans le premier cas, c'est seulement le verbe qui intervient, alors que dans l'autre, c'est aussi le paramètre "en deux colonnes". Une fois que l'on a pu associer une fonction à une structure prédicative, il va falloir vérifier si elle est bien formée au niveau de ses arguments : sont-ils tous-là et avec des valeurs convenables ?

9. En tous cas, le choix de ces en-têtes relève de l'analyse et la programmation de l'application et non de la langue naturelle.

2.4.2 La complétude des arguments dans l'en-tête de fonction

Avant de pouvoir exécuter une fonction, il faut avoir à disposition tous les arguments obligatoires dont elle a besoin. Or, il n'est pas forcément prévu (et cela ne serait même pas souhaitable) que l'utilisateur puisse accéder à tous les arguments (obligatoires ou optionnels) d'une fonction. On peut citer le cas d'une interface langagière UNIX qui n'aurait pas prévu qu'un utilisateur puisse faire référence à la caractéristique « bordure » au niveau des options d'impression. Pourtant, la fonction que l'on associe à "Imprime ce fichier en deux colonnes" :

a2ps -2 -ns < nom - fichier > | lpr

contient comme argument -ns qui indique que l'impression ne doit pas comporter de bord (par défaut, a2ps en met un). Le cas intéressant intervient lorsqu'un argument obligatoire (et libre) d'une fonction n'a pas été spécifié dans l'énoncé. Considérons par exemple la commande UNIX :

mkdir < nom - repertoire >

consistant à créer un répertoire dont on mentionne le nom. Si maintenant on a l'énoncé :

(2.20) Crée un répertoire.

on constate que le nom du répertoire n'est pas mentionné. Il y a alors deux possibilités pour traiter la commande. Ou bien le système trouve le nom lui-même, ou bien il le demande à l'utilisateur.

Trouver par soi-même. Il y a deux méthodes principales, consistant pour le système à utiliser une valeur par défaut ou bien à rechercher dans l'historique du dialogue.

A l'aide d'une valeur par défaut. Dans notre exemple, cela pourrait-être un nom comme *Dossier Vide*, à l'image de ce qui se fait dans le monde du MacIntosh. Libre alors à l'utilisateur de continuer son énoncé et de dire :

(2.21) Appelle-le Jeux¹⁰.

D'une manière générale, une valeur par défaut n'est pas donnée mais doit être inférée à partir de la situation. J. Klein [Klein, 1990] effectue une distinction entre cas obligatoire (présence explicite nécessaire dans l'énoncé) et cas semi-obligatoire (la valeur est nécessaire, et peut être calculée à partir du contexte linguistique ou de la situation de communication). En guise d'illustration, reprenons son contexte applicatif du robot évoluant dans un environnement de type bureau, pour comparer les deux énoncés suivants :

(2.23) Pose la chaise.

10. Cela revient à décomposer en deux étapes un énoncé :

(2.22) Crée un répertoire Jeux/de nom Jeux/s'appelant Jeux/qui s'appelle Jeux.

la valeur par défaut de la destination est *par-terre*, alors que dans

(2.24) Pose le cahier.

la valeur par défaut de la destination est *sur un bureau*.

J. Klein remarque (p. 137) que les valeurs par défaut utilisées dépendent d'une part du type du prédicat et d'autre part de l'objet sur lequel celui-ci porte (lorsqu'il y a un cas objet associé au prédicat). Une méthode est invoquée lors de la résolution de l'ellipse correspondant au traitement du cas semi-obligatoire (en l'occurrence : destination) pour les verbes de changement de lieu : rechercher la position normale de l'objet. Tout ceci va donc dans le sens des valeurs par défaut telle que les conçoit Falzon [Falzon, 1984] (p. 13-14), pour qui elles sont dépendantes du contexte et utilisées pour spécifier un slot manquant dans un schéma. De manière plus générale, nous affirmons que la valeur par défaut (correspondant dans notre cas à la destination de l'objet) dépend de la tâche à effectuer et donc du but qui lui est relatif.

Toujours dans le cadre applicatif d'un robot évoluant dans un environnement de type bureau, plaçons nous dans le contexte où il tient une poubelle entre ses pinces et que l'utilisateur lui prescrit la tâche de *passer l'aspirateur*. Dans ce cas, le lieu où le robot devra poser la poubelle ne sera plus *par-terre* comme dans le cas de 2.23 mais *un emplacement libre en hauteur* : le robot doit en effet maximaliser la surface sur laquelle il passe l'aspirateur. On pourrait alors envisager que la dépendance par rapport aux types du prédicat et de l'objet se situe au niveau d'un but exprimé par l'utilisateur.

Mais le recours systématique à des valeurs par défaut peut parfois poser problème : l'utilisateur peut ne pas mentionner un terme simplement parce qu'il s'agit du thème du dialogue. Plutôt que d'inférer une valeur par défaut, il faut alors envisager de recourir à un historique du dialogue.

Rechercher dans l'historique. Remarquons tout d'abord que, de même que l'inférence d'une valeur par défaut, la recherche dans l'historique est conditionnée par des données d'ordre pragmatique. Dans notre exemple de l'interface à UNIX, face à l'absence de nom spécifié dans l'énoncé 2.20, il ne saurait être question d'effectuer une recherche dans l'historique. Cela procède d'une donnée d'ordre purement pragmatique, à savoir qu'il n'est pas pertinent d'appeler deux objets distincts du même nom, à plus forte raison s'ils sont situés au même endroit.

De manière générale, on pose une question à l'interlocuteur seulement une fois qu'on a épuisé tous les moyens de trouver la réponse par soi-même.

Demander à l'interlocuteur. Dans notre exemple de création de répertoire, il s'agirait de demander :

(2.25) Comment voulez-vous l'appeler ?

question à laquelle l'utilisateur pourrait répondre : « Jeux ». Même dans ce cas, il est à noter que cette réponse constitue un énoncé à elle-seule, et doit être traité en tant que tel au niveau de l'analyse linguistique (tout en s'aidant du contexte). Il faut en effet bien être conscient que l'énoncé constitue l'unité du discours. Notamment, on peut très bien

2. Description du modèle "classique" de référence aux actions

prononcer des énoncés (délimités par un silence de début et un silence de fin) qui sortent du cadre de la langue. Imaginons un scénario où une personne commanderait la fabrication d'un meuble à un menuisier :

(2.26) Je voudrais un buffet avec des portes rabattantes, haut d'environ 80cm, comportant des tiroirs fermant à clef, ...

On peut aussi très bien imaginer un dialogue entre le client et le menuisier :

(2.27) Client : Je voudrais un buffet avec des portes rabattantes.

Menuisier : De quelle hauteur ?

Client : Environ 80cm.

Menuisier : Avec des tiroirs ?

Client : Oui, fermant à clef.

Dans les deux cas, on a affaire à un inventaire de caractéristiques, qui renferment la spécification de ce à quoi on doit aboutir. On commence alors à deviner l'obstacle que rencontrent les conceptions classiques de la référence aux actions. Rien ne garantit que l'ordre dans lequel interviennent les spécifications correspond à celui dans lequel les actions correspondantes seront effectuées.

On peut trouver des exemples simples empruntés à des situations de dialogue de commande dans lesquels interviendrait un tel chassé-croisé. On peut citer :

(2.28) Crée un répertoire public.

Appelle le Pot-Commun.

qui correspondrait à :

```
mkdir Pot-Commun
chmod a+rw Pot-Commun
```

On voit donc que le problème de l'optionnalité des paramètres est relatif à des prérequis imposés par la tâche (telle qu'elle est prescrite par l'utilisateur) à effectuer. Ce problème est d'autant plus important qu'il peut y avoir dans certains cas tendance à restreindre la longueur des messages et à concaténer plusieurs messages en un seul, comme le fait remarquer [Falzon, 1984], suivant un principe de minimisation des efforts. C'est dans des cas de ce genre que pourront apparaître des propositions relatives ou des groupes prépositionnels précisant l'objectif à atteindre.

2.4.3 Quelques procédés représentatifs d'association structure d'énoncé-fonction

Le système Partner. Le système Partner [Morin et Pierrel, 1987] est un système de dialogue qui se veut entièrement indépendant de la langue et de l'application utilisées. Par rapport à notre problématique, le point intéressant de ce système est que le module (*Cassis*) chargé de l'analyse des énoncés reconnus n'active aucune procédure externe mais range simplement le résultat de l'analyse dans l'historique du dialogue, le mettant ainsi à la disposition des autres modules. Un module (*Cassis*) calcule à partir du résultat de la reconnaissance globale un arbre syntaxico-sémantique binaire. Cet arbre est obtenu par

un analyseur utilisant une grammaire sémantique. La branche de gauche correspond à une étiquette (syntagme, classe grammaticale, classe sémantique) et celle de droite correspond à la valeur (réalisation dans l'énoncé). À partir de l'étiquette, on accède à cette valeur par l'intermédiaire d'une fonction *Value*. Les étiquettes peuvent avoir une extension, qui revient en fait à un typage, auquel on accède par l'intermédiaire de la fonction *Extend*. C'est à un type associé à un verbe que l'on fait coïncider une action dans le modèle de la tâche, qui est en fait une passerelle par rapport à l'application. La mise en correspondance avec un opérateur s'effectue alors par des règles externes qui sont activées par le module chargé de la gestion de la tâche.

Une commande est représentée dans le modèle de la tâche par un automate d'états, où chaque état correspond au plus à un besoin élémentaire. C'est le modèle de la tâche qui gère le tout, exprimant au module de dialogue ses besoins en paramètres lorsque ceux-ci sont manquants, sans préjuger de la manière dont ceux-ci seront remplis. Au niveau de la gestion du dialogue, la satisfaction d'un besoin exprimé par un état s'effectue selon l'ordre suivant :

- 1° une recherche dans l'historique du dialogue (contenant notamment l'énoncé de commande qui vient d'être prononcé) ;
- 2° la recherche d'une valeur par défaut ;
- 3° une requête vers l'utilisateur.

Non seulement le dialogue est entièrement guidé par la tâche mais il se voit également très restreint par cette même tâche. À partir d'un état de la tâche, on ne peut effectuer qu'un nombre limité de commandes. Si la commande ne convient pas, il y a retour arrière et re-synchronisation par rapport à un nouvel état.

Un des points positifs de ce système est que le déclenchement d'un schéma d'action n'est pas activé par un mécanisme d'extraction des composantes de l'énoncé qui entrent en jeu dans le calcul de la référence aux actions, mais par des règles externes. C'est là un premier pas vers une adaptation du système au naturel d'expression de l'utilisateur. Toutefois, ce système pêche par l'excès inverse, en étant rigidifié par la structure de la tâche.

Le cas de SHRDLU. La démarche de Winograd a été, comme nous l'avons esquissé précédemment, de structurer son programme sur les différents choix à effectuer pour véhiculer un sens, plutôt que sur la forme exacte que doivent revêtir les différents constituants linguistiques. Ainsi, il construit une abstraction des caractéristiques pertinentes des structures importantes pour interpréter leur sens. Dans son fonctionnement, SHRDLU repose sur trois modules séparés physiquement : syntaxe, sémantique et résolution de problème (un planificateur). En cela, il s'inscrit dans la lignée de quantité de systèmes qui l'ont précédé ou qui l'ont suivi. Mais ce qui constitue sa particularité, c'est de rompre avec la sérialité alors consensuelle des traitements¹¹ étant donné que les modules peuvent interagir entre eux à n'importe quel moment. La question que se posait à l'époque Winograd

11. À savoir traitement syntaxique, dont le résultat sert d'entrée pour le traitement sémantique dont la sortie sert à son tour pour la résolution de problèmes.

2. Description du modèle “classique” de référence aux actions

(d’après [Sabah, 1988] p. 255) est toujours pertinente :

« Comment concevoir un formalisme gardant les avantages de la décomposabilité (modularité) sans sacrifier l’expression des interactions? »

L’un des mérites de Winograd a été de mettre en interaction permanente les différents modules de son système (syntaxe, sémantique et raisonnement) et par là-même de leur ôter un aveuglement inévitable dès lors que l’on émet des hypothèses invérifiables sur le champ. Le facteur-clef responsable de cette forte intégration tient au fait que le savoir-faire procédural (ou instructionnel) est largement inscrit au niveau du lexique et de la grammaire, ce qui permet de rassembler conditions et actions sous forme de procédures quelconques, en l’occurrence en LISP.

Nous allons illustrer plus particulièrement ce fonctionnement au niveau de la référence verbale. Dans SHRDLU comme dans la majorité des autres systèmes, la fonction à appeler est déjà préaffectée au niveau de l’entrée lexicale correspondant au verbe. Toutefois, cette fonction peut dans certains cas être évaluée en cours d’exécution, par la prise en compte de l’environnement linguistique du verbe. Qu’il y ait nécessité ou non d’une évaluation (sous forme de macro-expanders d’expressions LISP), la fonction en question prend la forme :

```
(#NOM-DE-LA-FONCTION #i1 #i2 ... #iN *time*).
```

où les i_k correspondent aux arguments de la fonction, et peuvent avoir deux types d’origine :

- un rang au niveau du module casuel considéré d’un point de vue exclusivement sémantique. Ainsi, le rang numéro 1, correspondant dans une théorie de l’action à l’agent, n’est jamais utilisé. Nous pouvons noter que Winograd fait preuve d’un certain bon sens, en ne spécifiant pas – comme c’est notamment le cas pour les théories lexico-fonctionnelles – la forme syntaxique que doit prendre l’énoncé.
- une évaluation de paramètres à partir de constituants de l’énoncé.

Intéressons-nous plus particulièrement à une entrée lexicale représentative de ce système de compréhension, ne serait-ce qu’au niveau de l’application retenue : celle du verbe PUT. Outre les caractéristiques morphologiques du verbe, l’entrée lexicale renferme son cadre sémantique :

```
(((#animate)) ((#physob)) (smobl12 (#place)))
```

ainsi que le code LISP pour aiguiller l’appel de fonction (soit #putin, soit #puton) :

```
(mapcar  
  \#' (lambda (%plnrphrase)  
    (cond ((eq (car %plnrphrase) '\#on)  
          (list '\#puton '\#2 (cadr %plnrphrase) '*time))
```

12. smobl est une variable globale qui pointe sur la définition sémantique correspondant à un groupe nominal locatif.

```
((eq (car %plnrphrase) '\#in)
  (list '\#putin '\#2 (cadr %plnrphrase) '*time))
(T (error put definition)))
(relations? smobl))
```

Dans cet exemple, nous voyons apparaître la fonction `(relations? x)` qui renvoie le code destiné au planificateur dans le but de trouver le référent correspondant à la définition sémantique de `x`, en l'occurrence le groupe nominal locatif. Selon la préposition accompagnant le complément du verbe PUT, le nom de la fonction associée peut être soit PUTON soit PUTIN. Si l'on désirait raffiner la méthodologie de détermination de

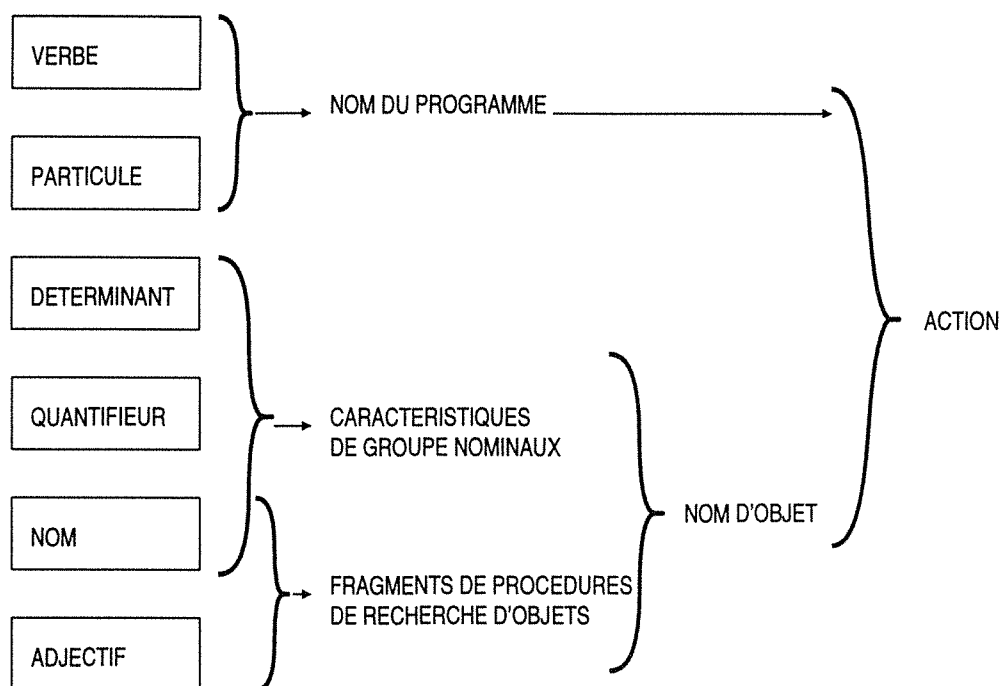


FIG. 2.12 - Procédé de calcul de la référence aux actions dans SHRDLU

la référence verbale sans la remettre en cause dans son ensemble, il faudrait compliquer un peu la portion de programme LISP relative à chaque entrée lexicale des verbes qui le nécessitent : rajouter un test de plus au niveau des marqueurs sémantiques relatifs aux objets afin de déterminer le nom du programme à appeler¹³ (figure 2.12).

Un grand apport de Winograd a été de montrer que l'on peut comprendre des énoncés très complexes en langage naturel à condition de se restreindre à un cadre limité. Il faut également noter une approche commune en ce qui concerne les énoncés de requête, que ce soient des commandes ou des demandes d'information. Dans tous les cas, la réponse correspond à l'exécution d'un programme lui-même synthétisé. La différence se situe au niveau de l'objectif mais pas au niveau de sa construction. Dans cette optique, le programme ainsi écrit constitue la référence de l'énoncé. Cela ne va pas non plus sans prendre de parti pris au niveau de la référence aux actions, ce qui fait dire à [Winston, 1977] (p.

13. C'est ce que l'on pourrait appeler une démarche *ad hoc*

176) :

« There is no need for intermediate computation in which descriptions become search programs and search programs which then find names. Program names come directly from the verb that specify them. »

Winston reconnaît de ce fait l’analogie directe qu’il effectue entre les énoncés et les fonctions produites. Toutefois, il ne pose pas cette analogie comme un principe général, mais il la présente comme une commodité liée à la simplicité du monde des blocs.

Le système Nalige. Le système Nalige [Manaris et Dominick, 1993; Manaris *et al.*, 1994] est destiné à offrir une interface en langage à consonance naturelle pour un système d’exploitation. Il fait appel à des spécifications décrivant les attendus linguistiques de la part de l’utilisateur de l’application. On peut distinguer trois composantes dans ce système :

1. NLIOS : domaine sémantique identifié en termes d’actions et entités conceptuelles ;
2. le sous-ensemble des commandes langagières qui réalise le domaine ;
3. les structures de surface appropriées correspondant à ce domaine sémantique.

La composante (1) sert en quelque sorte de pivot entre les composantes (2) et (3). Ci-dessous figure un extrait de (1) représentant les actions portant sur les fichiers et les répertoires :

File actions:

```
[copy-file] <file-spec-1> <file-spec-2>
[create-file] <file-spec>
[delete-file] <file-spec>
[print-file-contents] <file-spec>
[rename-file] <file-spec-1> <file-spec-2>
[show-file-contents] <file-spec>
[spell-check-file-contents] <file-spec>
```

Directory actions:

```
[change-working-directory] <directory-spec>
[create-directory] <file-spec>
[delete-directory] <directory-spec>
[print-directory-contents] <directory-spec>
[rename-directory] <directory-spec-1> <directory-spec-2>
[show-directory-contents] <directory-spec>
```

La correspondance s’effectue de manière globale en plaquant un gabarit de structure d’énoncés à trous associé à un élément de (1) :

```
action create-directory-act(directory-spec : dirn)
```

```
begin
  surface_description : "create a new directory called ",dirn;
  OSCL_realization : "mkdir ",dirn;
  pragmatic_constraint : not_exists_directory(dirn);

  ... (traitement des erreurs)
end
```

Une grammaire sémantique étendue possédant des actions associées à chaque règle de production¹⁴ permet d'extraire les paramètres utiles au niveau de l'énoncé (par l'intermédiaire de la fonction d'invocation V et le numéro du mot, x en l'occurrence, dans l'énoncé), comme illustré ci-dessous.

```
<CREATE-DIRECTORY> : [CREATE-V] <CREATE-DIR-SPEC>
|| '(create-directory-act : directory-spec ,(V 2));
```

```
<CREATE-DIR-SPEC> : [*FILENAME*] || (V 1)
| [DIR-N] [*FILENAME*] || (V 2)
| [DIR-N] [REF-V] [*FILENAME*] || (V 3)
| [NEW-ADJ] [DIR-N] [*FILENAME*] || (V 3)
| [NEW-ADJ] [DIR-N] [REF-V] [*FILENAME*] || (V 4)
| [DEF] [DIR-N] [*FILENAME*] || (V 3)
| [INDEF] [DIR-N] [REF-V] [*FILENAME*] || (V 4)
| [INDEF] [NEW-ADJ] [DIR-N] [*FILENAME*] || (V 4)
| [INDEF] [NEW-ADJ] [DIR-N] [REF-V] [*FILENAME*] || (V 5);
```

Cet extrait de grammaire permet de représenter la classe d'énoncés correspondant à la création d'un répertoire: *create x*, *make directory x*, *new directory x*, *create directory named x*, *create directory named x*, *create new directory x*, *make a new directory x*, *make a new directory named x*, ... On voit donc ici pleinement illustrée la bijection de l'ensemble quotient des énoncés L/\mathcal{R} sur le modèle de l'application, comme nous l'avions précédemment introduite.

2.5 Récapitulatif

Nous avons mis en évidence trois grandes variantes d'identification d'une action, reposant toutes sur la même hypothèse qui consiste à confondre l'action spécifiée dans l'énoncé avec la représentation d'une fonction de l'application. Ainsi, la mise en correspondance avec les fonctions de l'application peut s'effectuer à différents niveaux :

1. Au niveau des entrées lexicales. C'est le verbe qui joue un rôle central (Shrldu [Winograd, 1972], Shiva [Teil et Bellik, 1991]);

14. Le séparateur utilisé est une double-barre.

3. Les limites concernant l'adjonction de naturel

2. Au niveau de la description des schémas d'action. Le schéma d'action contient le type de l'objet sur lequel une fonction s'applique, les verbes auxquels il correspond. (ICP-Draw [Caelen, 1991; Caelen et Coutaz, 1991¹, Nalige [Manaris et Dominick, 1993; Manaris *et al.*, 1994]);
3. De manière externe, sous forme de règles de filtrage (Partner [Morin et Pierrel, 1987]).

Quelle que soit la variante utilisée, on établit un isomorphisme entre la structure de la langue et celle de l'application. Cela n'est pas sans imposer des limites fâcheuses aux systèmes de dialogue de conception classique. Nous allons nous centrer sur celles qui ont trait à l'adjonction de naturel.

3 Les limites concernant l'adjonction de naturel

L'intérêt de rajouter du naturel dans la composante langagière d'une interface réside dans le fait que le fonctionnement de la langue rentre dans les compétences de n'importe quel utilisateur. Les déterminants, les prépositions, les conjonctions, tout ce que l'on a coutume d'appeler les « mots-outils » sont autant de caractéristiques locales qui contribuent au naturel d'un langage. Mais il faut en outre se situer par rapport à une perspective globale, au niveau de la régularité de la langue. Nous allons développer ce dernier aspect sur un point précis : la régularité des groupes nominaux et prépositionnels.

3.1 Un objectif à atteindre : une certaine régularité langagière

Les énoncés qui se trouvent être regroupés en classes (cf. figure 2.5 p. 62) ne sauraient être considérés isolément les uns des autres : ils forment en effet un tout, indépendamment de la classe à laquelle ils appartiennent. Dès que l'on envisage une interaction avec un langage un peu plus naturel, on peut souhaiter que le sous-langage employé conserve cette propriété essentielle des langues naturelles : être un système cohérent [Benveniste, 1974b]. Plus concrètement, les propriétés linguistiques mises en évidence dans tel ou tel énoncé ne sont pas spécifiques à cet énoncé mais pourraient se retrouver dans n'importe quel énoncé. Ainsi, si un utilisateur fait référence à un cercle en précisant sa couleur comme dans l'énoncé 2.29, il pourrait être fort dérouté de se voir refuser l'énoncé 2.30, sous prétexte que le concepteur n'a pas prévu dans son modèle de l'application de dessiner un cercle d'une couleur donnée en une seule étape (et donc qu'il n'existe pas de fonction de l'application qui le fasse).

(2.29) Efface le cercle rouge.

(2.30) Dessine un cercle rouge.

Ce serait notamment le cas si la structure-même de l'application imposait de choisir la couleur du pinceau avant de tracer le cercle. Cette analyse se trouve confirmée par les

travaux des psycho-ergonomes Carroll et Thomas [Carroll et Thomas, 1980], qui sont parvenus à la conclusion suivante :

« People learn structures and not isolated pairings (fallacy of composition).
A system interface must be a coherent structure. The learning of any paired
associate reciprocally affects the learning of all other paired associates. »

Un concept essentiel rendant compte de la cohérence de la structure du langage est celle d'orthogonalité de la syntaxe. Dans [Green et Payne, 1984], on trouve une bonne illustration de cette notion sous forme d'une métaphore. Considérons le diagramme suivant (figure 2.13) : la question posée est de compléter la case contenant un point d'interroga-

+ + +	○	* *
○ ○ ○	* * *	+
*	?	○ ○

FIG. 2.13 - Une métaphore de l'orthogonalité de la syntaxe

tion. Toute autre réponse que deux signes « plus » semblerait pour le moins audacieuse, pour ne pas dire farfelue. Pour ne pas rester à un niveau aussi intuitif, on peut donner une définition plus précise de l'orthogonalité :

Definition : Un système \mathcal{S} est dit orthogonal pour un ensemble de relations de similarités si pour tout couple d'objets de \mathcal{S} présentant des similarités, il est possible d'utiliser autant l'un que l'autre dans tous les contextes logiquement acceptables.

Nous allons examiner ce qu'impose aux démarches traditionnelles la prise en compte de ce type de régularité.

3.2 Une solution (trop) complexe

Nous avons vu que si l'application comptait un certain nombre de points d'entrée correspondant aux outils qu'elle met à la disposition de l'interface, cela mettait en jeu autant d'opérateurs. Il est néanmoins tout à fait envisageable de concevoir des opérateurs (dits complexes dans le sens où ils ne sont plus élémentaires) pour enrichir cette application.

3.2.1 L'incorporation d'opérateurs complexes

Dans notre exemple 2.30, on pourrait introduire dans le modèle de l'application un opérateur complexe *dessiner-cercle-avec-couleur* en utilisant les opérateurs élémentaires *choisir-couleur* et *tracer-cercle*. Toutefois, du point de vue du modèle de l'application, on

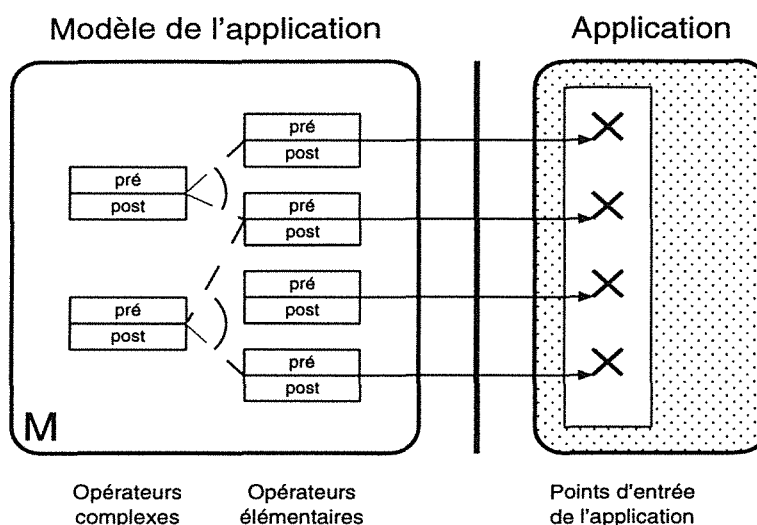


FIG. 2.14 - *Modèle structuré de l'application*

en vient à distinguer des opérateurs élémentaires (tels que nous les avons décrits en 1) et des opérateurs complexes (cf. figure 2.14). Du point de vue de la composante linguistique de l'interface, ces deux types d'actions, élémentaire ou complexe, sont indistingables. On peut alors considérer de manière équivalente que l'on a affaire à une application comportant davantage de fonctionnalités directement accessibles.

Un opérateur complexe se différencie d'un opérateur élémentaire par le fait que son corps peut correspondre à un regroupement figé d'opérateurs. On peut à ce niveau effectuer un rapprochement avec la notion de *script* introduite par Schank. D'après [Schank et Abelson, 1977], l'homme obéit dans sa vie quotidienne à une grande quantité d'événements stéréotypés pour lesquels les actions s'enchaînent les unes aux autres sans que les divers choix, à un moment donné, ne soient envisagés. C'est ce qui correspond à nos réflexes acquis, comme c'est le cas pour la conduite automobile. Ainsi, dès lors que le conducteur perçoit que le moteur de l'automobile est en sur- ou sous-régime, il débraye machinalement en même temps qu'il change de vitesse. Considérons une application imaginaire¹⁵ de commande d'une automobile, pour laquelle on disposerait entre autres des opérateurs élémentaires suivants :

- *debrayer* ;
- *embrayer* ;
- *passer-vitesse-superieure* ;
- *passer vitesse-inferieure*.

15. Il existe pour cela des boîtes de vitesse automatiques qui font très bien leur travail !

On aurait alors typiquement les scripts suivants :

- $\left[\begin{array}{l} \text{debrayer} \\ \text{passer-vitesse-superieure} \\ \text{embrayer} \end{array} \right]$ (associé à un sur-régime)
- $\left[\begin{array}{l} \text{debrayer} \\ \text{passer-vitesse-inferieure} \\ \text{embrayer} \end{array} \right]$ (associé à un sous-régime)

La bijection entre l'ensemble quotient des énoncés et le modèle de l'application devient donc la suivante (figure 2.15) :

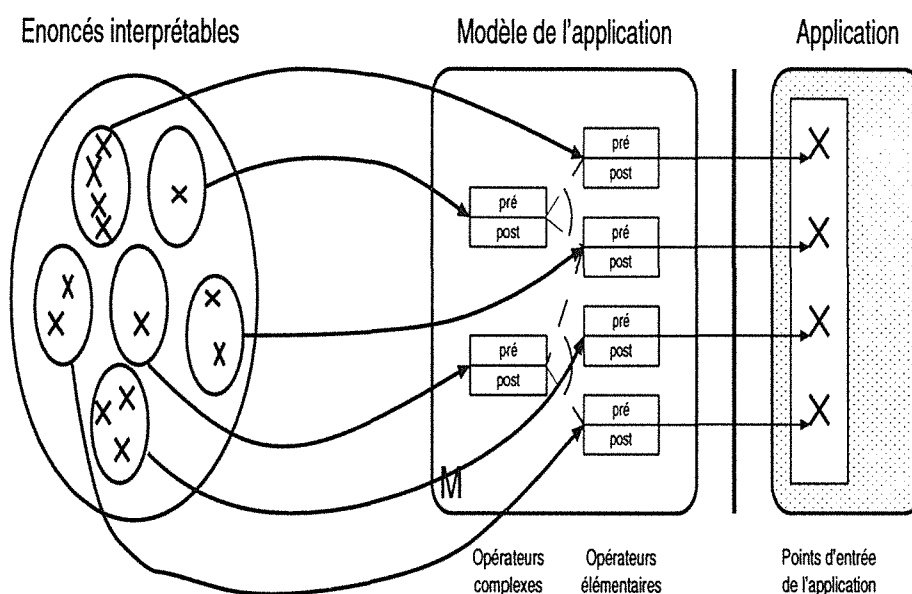


FIG. 2.15 - *Bijection classique (2)*

3.2.2 Conséquences sur la prise en compte de la régularité des groupes nominaux

Partant d'une fonction à déclencher, on considère la classe de tous les énoncés qui lui font référence (cf. figure 2.5). L'analyse linguistique revient alors à délimiter la classe à laquelle appartient un énoncé. Les énoncés 2.29 et 2.30 ont la même forme de surface. Pourtant, la différence de déterminant fait que l'adjectif *rouge*, qui n'intervenait dans 2.29 que pour l'identification de l'objet, entre en jeu dans 2.30 pour déterminer l'opérateur concerné. Le critère de régularité impose toutefois que, si l'on accepte un adjectif qualificatif dans un groupe nominal relatif à un énoncé de commande particulier, on doit l'accepter dans tous les groupes nominaux où il aurait un sens. C'est à ce niveau que l'on rentre en conflit avec l'exhaustivité du modèle de l'application. Tenir compte de la régularité de construction des groupes nominaux impose de rendre cet ajout d'opérateurs systématique.

3. Les limites concernant l'adjonction de naturel

Concrètement, supposons qu'à l'énoncé *dessine un cercle de rayon (RAYON)* on associe l'opérateur élémentaire *dessiner-cercle(RAYON)*. A partir du moment où l'on autorise les adjectifs de couleur dans les groupes nominaux, il faut prévoir d'introduire un nouvel opérateur, complexe celui-ci, *dessiner-cercle-avec-couleur(RAYON, COULEUR)* à tous les énoncés de la forme *dessine un cercle (Adj-Couleur) de rayon R* pour ne citer que cette classe d'énoncés.

Le seul ajout de la possibilité d'identifier un cercle par sa couleur conduit donc à :

1. Considérer *tous les opérateurs* pour lesquels une modification de la couleur d'un objet aurait un sens ;
2. Pour chaque opérateur \mathcal{O} considéré, rajouter un opérateur complexe faisant appel à *choisir-couleur* et \mathcal{O} .

D'un point de vue général, l'ajout d'une possibilité de qualification a donc un effet multiplicatif sur le nombre d'opérateurs complexes à ajouter.

3.2.3 Conséquences sur la prise en compte de la régularité des groupes prépositionnels

Supposons maintenant que, dans l'objectif d'obtenir davantage de naturel, on accepte l'utilisation des prépositions, en se donnant ainsi la possibilité de former des groupes prépositionnels ($G.Prep \rightarrow Prep + G.Nom$). De même qu'un adjectif, un groupe prépositionnel peut participer à l'identification d'un objet (ex : *Supprime la fenêtre à droite de l'icône*) ou spécifier un paramètre requis pour un opérateur (ex : *Peins ce rectangle en vert*). Pour les mêmes raisons de régularité que celles évoquées ci-dessus, on doit envisager d'accepter des énoncés tels que *Déplace la fenêtre à droite de l'icône*. Or, *à droite de l'icône* est-il à considérer comme spécificateur de la fenêtre, ou comme complément (de lieu) du verbe *déplacer*? Si le complément de lieu est obligatoire pour le verbe, il n'y a pas d'ambiguïté. Mais ce n'est pas toujours le cas. Ainsi, aucun critère régulier ne ressort pour décider du caractère optionnel ou obligatoire d'un complément. Dans le cadre informatique où nous nous situons, il est néanmoins possible de trouver diverses issues de secours.

Deux solutions sont classiquement proposées pour traiter le cas des compléments dits optionnels.

- i La première consiste à introduire un nouvel opérateur dans lequel le complément optionnel se comporte comme un paramètre obligatoire. Lorsqu'il n'est pas présent, on utilise l'ancien opérateur.
- ii La seconde consiste à définir dans le modèle de l'application des valeurs par défaut lorsque les paramètres optionnels correspondants ne sont pas présents. De façon corollaire, un complément sera considéré comme optionnel lorsqu'il ne sera pas utilisé pour identifier le référent à une action.

Comme nous le faisons remarquer dans [Duermael et Gaiffe, 1993], cela revient à introduire des opérateurs pour empêcher la formation de groupes nominaux référentiels indésirables :

« The reference problem we put forward here is strongly related to the so-called mandatory and optional cases of case-based grammars. The mandatory cases are partially intended to forbid the links between some prepositional noun phrases and other noun phrases. The point is that subordinating some of the noun phrases directly to the verb prevents links between these ones, thus avoiding ill-formed referential noun phrases rather than performing a correct predicative interpretation. »

Or, tout laisse à présager que les différents groupes ainsi subordonnés interagissent sémantiquement entre eux. Juste pour citer un exemple, s'il s'avérait que le document auquel réfère l'énoncé 2.11 (*Imprime ce document en deux colonnes*) correspond à une image, celui-ci deviendrait asémantique. Le syntagme prépositionnel *en deux colonnes* spécialise donc le document en question.

Pour résumer, la prise en compte de la régularité langagière au niveau des compléments indirects amène à s'interroger sur le caractère obligatoire ou optionnel des compléments en question. Cela a pour effet de dissocier les interactions entre les compléments et donc d'accroître aveuglément la combinatoire des compléments possibles. On retrouve ainsi un nouvel effet multiplicatif sur le nombre d'opérateurs complexes à rajouter dans le modèle de l'application. À ces deux effets s'ajoutent les effets multiplicatifs que nous avons évoqués à propos du groupe nominal, puisqu'un groupe prépositionnel inclut un groupe nominal. Il s'avère par conséquent que cette solution est globalement de complexité exponentielle.

4 Conclusion

Lorsqu'une démarche qui a le mérite d'être simple présente malgré tout un certain nombre de limites potentielles ou avérées, il est important pour le concepteur de connaître les éléments de son système qu'il devrait remettre en cause si la tâche venait à se complexifier ou (les deux choses n'étant pas indépendantes) s'il s'avérait souhaitable de rajouter un peu de naturel dans l'expression des commandes.

En définitive, les méthodes traditionnelles ont cherché le plus court chemin entre l'énoncé de commande et la fonction à appeler. Partant de la constatation que l'appel d'une fonction est constitué de son nom et d'arguments typés, le problème est alors devenu le suivant :

Quels sont les éléments de l'énoncé qui interviennent pour la détermination des arguments et quels sont ceux qui interviennent pour la détermination du nom de la fonction, c'est-à-dire l'outil de l'application à appliquer aux objets précédemment déterminés ?

La nature des exemples employés (de type création d'objets) et des arguments avancés quant au rajout d'opérateurs (de type sémantique relativement à la modification de caractéristiques d'objets) nous incite à penser que le problème est d'ordre temporel.

Chapitre 3

Une approche exploitant la dynamique temporelle de l'énoncé

Sommaire

1	L'association « énoncé-but »	92
1.1	La prise en compte des effets des fonctions	92
1.2	Une approche adaptée à une régularité langagière	95
1.3	La validité du point de vue psychologique	96
1.4	Un partage d'activité envisageable	97
2	Les moyens à développer	98
2.1	Le recours à un planificateur	98
2.2	Une analyse linguistique en termes de but à atteindre	104
3	Conclusion	105

Dans notre présentation des systèmes de dialogue à composante orale, nous avons mis en évidence le caractère finalisé de la communication par rapport à une tâche à accomplir, limitée à un domaine d'application. Cela nous a permis d'isoler une fonctionnalité du dialogue relative à certains types d'applications, consistant à prescrire une tâche à la machine. Notre thèse est qu'un énoncé de commande contient les éléments nécessaires participant à la description de cette tâche : un but à atteindre par rapport à la situation courante.

Dans les approches classiques de référence aux actions, ce sont les postconditions qui rendent compte du but à atteindre. Or, le fait de les reléguer au second plan a complètement occulté la dynamique temporelle que peut véhiculer un énoncé de commande. Ce que nous proposons est justement d'analyser les énoncés en terme des changements qu'ils font intervenir par rapport aux objets auxquels ils réfèrent, faisant ainsi jouer d'emblée un rôle majeur au temps. Avant de nous focaliser sur les différentes composantes requises, nous allons mettre en évidence les divers avantages que cela procure par rapport aux méthodes classiques.

1 L'association « énoncé-but »

Jusqu'à présent, il a été peu question de tâche, alors même que le dialogue est censé être orienté par la tâche. Aussi allons-nous commencer par nous intéresser à ce qui caractérise une tâche : un but à atteindre, matérialisé par les effets provoqués par des fonctions.

1.1 La prise en compte des effets des fonctions

Dans les approches que nous venons d'exposer, nous avons montré que les effets (ou postconditions) d'un opérateur avaient pour principale raison d'être la mise à jour de l'état de l'application après appel de la fonction associée. C'est dans ces conditions la conception suivante de l'opérateur qui se trouve privilégiée :

« Si les conditions de validité pour appliquer une opération (une fonction ou une séquence de fonctions) sont présentes et que cette opération est lancée, alors tel effet est produit. » ([Hoc, 1988], p.27)

Les postconditions, qui ne jouent alors aucun rôle dans l'application d'une opération, se voient par conséquent reléguées à l'arrière-plan.

Or, nous allons exhiber différents cas où les méthodes classiques sont amenées à emprunter une voie détournée pour ne pas faire jouer aux postconditions le rôle primordial qui leur reviendrait de droit.

1.1.1 La dissimulation des postconditions dans les approches classiques

L'anticipation des effets. Il n'est pas toujours souhaitable d'appeler directement une fonction si celle-ci peut avoir des effets préjudiciables. Il faut alors anticiper l'effet d'une action. Plaçons nous dans le cadre d'un dialogue avec un interpréteur de commandes en ligne, par exemple UNIX. Si l'utilisateur tape :

(3.1)

(a) `rm /tmp/*`
ou bien

(b) `rm ~/**`

consistant respectivement à effacer tous les fichiers temporaires ou toutes les anciennes versions des fichiers situés dans son répertoire personnel, il n'y aura pas besoin de lui demander de confirmation. Par contre, il en irait tout à fait autrement s'il tapait :

(3.2) `rm /These/Version-definitive/*`

Dans le premier cas (3.1 (a) ou (b)), on aurait comme postconditions : *des fichiers temporaires effacés* (ou encore *des anciens fichiers effacés*), alors que dans le second (3.2), ce serait *la version définitive de la thèse effacée*. Il va de soi que cela n'a pas la même gravité dans un cas ou dans l'autre. Nous voyons donc que ce qui détermine un retour vers l'utilisateur ne se situe pas au niveau de l'action en elle-même mais au niveau de

l'effet hypothétique qu'elle produit sur les objets. On peut retrouver un tel travers dans le système d'aide à l'utilisation d'Unix de [Danlos *et al.*, 1985], qui représente une commande en termes d'opérations (par exemple, la destruction de répertoire) et non en termes de changements d'états (par exemple, le passage de l'état existant à l'état non-existant d'un répertoire). Toutefois, Danlos remarque que ce système, s'il est adapté à l'aide en contexte sur le fonctionnement des commandes, ne conviendrait pas pour effectuer de la résolution de problèmes (à laquelle s'apparente plus notre exemple).

Les systèmes classiques, en s'appuyant uniquement sur les conditions de validité pour appliquer des opérateurs, se voient contraints de déguiser des postconditions hypothétiques en préconditions. Mais en posant d'emblée des postconditions à vérifier, nous rentrerions au sens strict dans le domaine hypothétique, ce qui est le cas tant que la fonction n'a pas été lancée. Il existe un autre argument qui fait jouer aux postconditions un rôle majeur.

Les référents « évoluant dans le temps ». Reprenons notre exemple :

(3.3) Crée un répertoire public.

et interrogeons-nous sur le rôle que l'on peut faire jouer à l'adjectif *public* dans l'identification de la (ou des) fonction(s) à exécuter.

Dans la démarche classique, nous avons vu que *public* devait servir directement à l'identification d'un opérateur associé au script. Une manière d'en rendre compte est d'incorporer (sous une forme ou une autre) cette caractéristique dans le nom de l'opérateur : *creer-repertoire-public* [nom-de-repertoire] aux côtés du verbe *crée* et du type *répertoire*. Nous avons en outre remarqué qu'il était possible d'effectuer des regroupements acceptables en introduisant des opérateurs admettant des arguments qui joueraient alors le rôle de paramètres d'action, par exemple *creer-repertoire* [nom-de-repertoire] [statut]. Cela sous-entend que ces paramètres correspondent à une classe sémantique bien définie, par exemple l'opposition *public vs. privé*, exprimant le *statut* d'un fichier (droit d'accès). Dans notre cas, on associerait la valeur "g+rxw o+rxw" à l'expression *public* et "g-rwx o-rwx" à l'expression *privé*.

Mais ce n'est pas là le seul rôle de *public*. Il intervient aussi naturellement au niveau des postconditions. Ainsi, si l'on simule l'action correspondant à *creer-repertoire-public* et que l'on appelle ?x l'objet correspondant au répertoire créé, on aura comme postcondition : *public(?x)*. Dans le cas où l'on introduit un paramètre ?y correspondant au statut, les postconditions seront conditionnelles :

privé(?x) si ?y ≠ *umask* (valeur par défaut)
public(?x) si ?y = "g+rxw o+rxw"

Ce qui est important de noter ici, c'est que ?x n'est pas un objet de l'application dans la situation initiale, mais le devient après l'exécution de l'action, ce qui est indiqué par les postconditions. L'énoncé met ainsi en évidence une propriété qui n'est relative qu'à un état final. Nous voyons dès lors qu'il est possible d'envisager qu'un énoncé puisse spécifier un état résultant. Ainsi, le prédicat va jouer systématiquement le rôle d'une propriété d'objets. Ces deux critiques nous orientent directement vers une autre démarche, où les postconditions jouent un rôle majeur pour concevoir la référence aux actions.

1.1.2 Une approche alternative de la référence aux actions

Pouvoir faire référence à des postconditions renverse complètement la perspective dans laquelle on envisage la référence aux actions, comme le montre la figure 3.1. La démarche

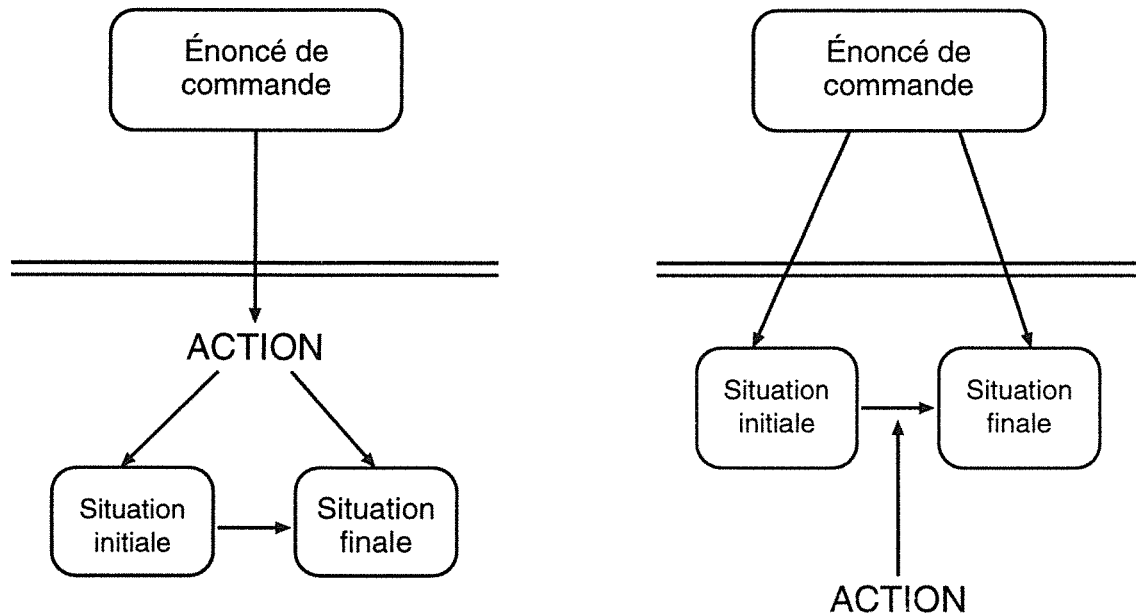


FIG. 3.1 - Comparatif de deux approches de référence aux actions

classique conçoit l'action comme un tout, sans chercher à la décomposer temporellement. On peut ainsi dire que l'action est envisagée selon un point de vue global. Notre approche vise au contraire à exploiter la temporalité de l'action. Cette temporalité transparaît d'ailleurs directement à travers la définition que nous en avons donnée : un passage contrôlé entre une situation initiale et une situation finale. La temporalité émerge d'elle-même lorsque l'on projette (temporellement) le but à atteindre. Nous prétendons alors que l'énoncé spécifie une situation finale à atteindre, compte tenu de la situation courante. Ainsi, dans notre exemple, on veut qu'un objet de type répertoire, ayant la propriété d'être public, soit créé.

La différence entre les deux approches est importante quant à la manière dont les postconditions des opérateurs sont prises en compte. Pour faire le parallèle avec la conception classique de l'opérateur, nous faisons émerger la règle suivante d'utilisation [Hoc, 1988, p. 28] :

« Si le but est d'obtenir un certain résultat à partir d'un état donné, alors exécuter telle et telle opération (pour l'atteindre). »

Les opérations (fonctions) ne sont plus données *a priori* mais sont désormais déduites avec un ensemble de règles contextuelles. Cette vision des choses est directement compatible avec les notions de planification et de résolution de problèmes telles qu'on les rencontre en Intelligence Artificielle. Ainsi, on voit apparaître la nécessité de recourir à un planificateur et de se démarquer de cette façon de l'aspect langagier.

1.2 Une approche adaptée à une régularité langagière

Si l'on vise une régularité langagière, un minimum à satisfaire est une régularité au niveau de la composition des groupes nominaux acceptables. Notamment, si l'on souhaite incorporer des adjectifs en tant que compléments de détermination, il faudra aussi prévoir de traiter des groupes prépositionnels qui peuvent avoir la même fonction. Prenons un exemple comprenant des variations élémentaires de paramètres autour de la commande d'impression d'un document. Ainsi, si l'on accepte :

(3.4) Imprime [S_N le document « Rapport.ps »]

on devra dans ce cas accepter aussi :

(3.5) Imprime [S_N le document sur deux colonnes]

(pour peu que le nombre de colonnes d'un document soit une caractéristique des documents définie dans le modèle de l'application), mais aussi n'importe quelle suite verbale faisant sens, sous forme d'un groupe prépositionnel.

(3.6)

- (a) Imprime ce document en recto-verso.
- (b) Imprime ce document sur deux colonnes.
- (c) Imprime ce document en deux exemplaires.
- (d) Imprime ce document sur la laserjet de la salle console.
- (e) Imprime ce document dans la salle console.
- (f) Imprime ce document dans cinq heures.
- (g) Imprime ce document dans la nuit.
- (h) Imprime ce document cette nuit.

Mais il faut aussi prévoir des combinaisons, comme par exemple :

(3.7) Imprime ce document en recto-verso sur deux colonnes pendant la nuit.

Au niveau linguistique, cela revient à caractériser un nouvel état à atteindre dans le monde de référence que constitue l'environnement applicatif [Falzon, 1984] (p. 5). Concrètement, cela nécessite de faire référence à des propriétés et à des relations entre objets composant l'environnement. Une propriété ne concerne qu'un seul objet (supposant éventuellement des dépendances à d'autres objets) alors qu'une relation en concerne deux. Ainsi la couleur, le poids, la vitesse sont des propriétés alors que l'adjacence, le support, la visibilité mutuelle sont des relations. La spécification d'un état final pourrait s'effectuer ainsi (en s'inspirant du formalisme introduit par Davidson) :

$$\exists e, x(instance(x) = doc_{124} \wedge imprime(e, x) \wedge type(x) = document \wedge en(recto - verso, x) \wedge sur(colonnes, x) \wedge card(colonnes) = 2$$

On remarque que, comme il se doit, *pendant(e, NUIT)* n'apparaît pas dans cette formule, puisqu'il s'agit d'une contrainte sur le déroulement.

En fait, le calcul de la référence aux actions n'est pas plus compliqué que celui de la référence aux objets. On peut même avancer que si un système possède les composantes nécessaires au calcul de la référence aux objets, il pourra effectuer le calcul de la référence à l'état initial et à l'état final. L'approche que nous proposons évite donc d'introduire de nouveaux opérateurs suivant une complexité exponentielle pour viser une régularité langagière. Ce que l'on peut dire avant tout, c'est qu'elle n'introduit pas de descripteurs supplémentaires. En partant de l'hypothèse que l'on peut accéder aux référents par l'intermédiaire de leur description, il n'est pas plus coûteux de décrire ces référents que de décrire un état final auquel on doit parvenir. Par exemple, si l'on se donne les moyens lexicaux¹ pour référer à *la fenêtre avec un ascenseur* ou *la fenêtre bleue*, on peut tout aussi bien décrire l'état final résultant de *crée une fenêtre avec un ascenseur* ou *crée une fenêtre verte*, ou encore *crée une fenêtre verte avec un ascenseur*.

1.3 La validité du point de vue psychologique

Concevoir un énoncé comme la spécification d'une situation finale à partir d'une situation initiale revient implicitement à le considérer comme la prescription d'une tâche à accomplir. Cela est à mettre directement en lien avec les notions d'implicite et d'explicite au niveau de la prescription d'une tâche [Hoc, 1980, p. 53].

« Dans toute description de tâche, il y aura une part d'implicite. Toutes les conditions ne sont pas données et on pourra distinguer les conditions explicites qui sont fournies par la description et les conditions implicites qui sont passées sous silence mais doivent être prises en considération (...) Quelquefois, la tâche ne donne lieu à aucune prescription explicite et la prescription est donnée à un niveau très général et le sujet doit définir lui-même les unités adéquates. »

Ainsi les éléments nécessaires à la description d'une tâche se retrouvent dans l'énoncé. On va même jusqu'à trouver dans l'énoncé le but rattaché à la tâche :

« Broadly speaking, understanding is a goal-directed activity, which consists of constructing a coherent representation of a set of elements. The goal corresponds to what individuals aim at accomplishing with the representation they have constructed. (...) Coherence is obtained by integrating all the information relevant to the goal into a single structure. » ([Hoc, 1988] p. 169-170)

Rasmussen [Rasmussen, 1986] va aussi directement dans ce sens. À partir de l'observation des informations et des données, l'utilisateur identifie l'état présent du système, ce qui lui permet d'évaluer l'état cible auquel il désire parvenir. Sa tâche ainsi définie, il peut se construire un plan de séquences d'actions, qui conduira à l'exécution de procédures.

1. Avec une sémantique appropriée, permettant de différencier *les fenêtres des fenêtres sans ascenseur*, ou encore *les fenêtres vertes des fenêtres d'une autre couleur*.

1.4 Un partage d'activité envisageable

Dès lors que le système recourt à un planificateur, l'utilisateur n'a plus à prendre en charge toute l'activité du système et peut se situer à un niveau supérieur à celui des outils mis à sa disposition par l'application. Il en résulte un partage de l'activité, puisque le système prend en charge une partie qui n'est spécifiée que par la tâche prescrite par l'utilisateur.

Une conséquence pour l'utilisateur va être de pouvoir prescrire des tâches correspondant à des buts de divers niveaux d'abstraction, comme cela est illustré dans la figure 3.2. On obtient ainsi une coopération entre l'utilisateur et le système dans la réalisation

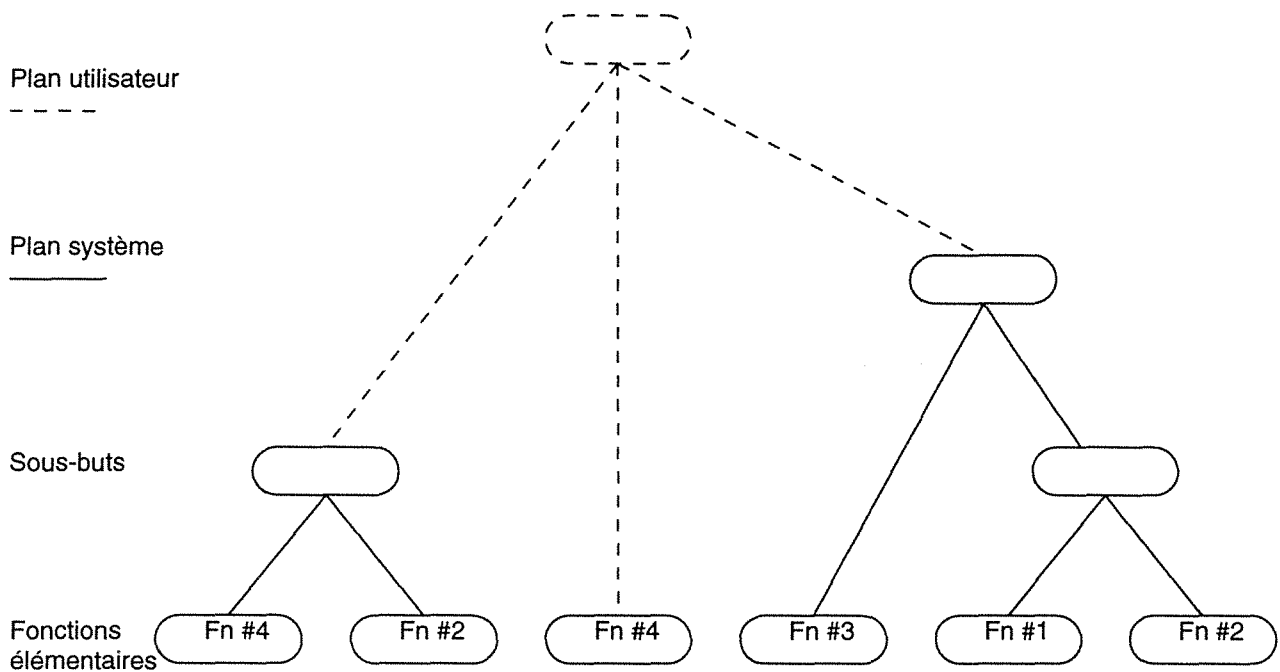


FIG. 3.2 - Partage d'activité entre utilisateur et système

d'une tâche partagée, comme nous l'avons vu au chapitre 1. Il serait toutefois illusoire de croire que cette coopération puisse s'effectuer à n'importe quel niveau. Dans n'importe quel logiciel de dessin pourvu d'une interface langagière, des commandes telles *Dessine-moi un mouton* ne peuvent pas être envisagées. On se heurte à la fois (a) à un problème de représentation des connaissances rattachées à des entrées lexicales, et (b) à des choix laissés à l'entière initiative du système.

Partant des fonctions élémentaires mises à la disposition de l'interface, qui doivent être toutes accessibles à l'utilisateur, la question concerne donc le niveau jusqu'où l'utilisateur peut remonter. Le cadre de régularité que nous nous sommes fixé va permettre d'y répondre. Par exemple, s'il est prévu une fonction élémentaire correspondant à *crée un répertoire* et une autre à *rends ce répertoire public*, il doit falloir se placer à un niveau supérieur pour accomplir la tâche prescrite par l'énoncé *crée un répertoire public*. En fait, c'est la régularité langagière respectée qui fixe le degré d'initiative du système.

Nous voyons donc qu'une telle approche présente des avantages non négligeables par

rapport aux démarches que l'on a coutume de rencontrer. Aussi allons-nous nous intéresser aux moyens à développer pour la mettre en œuvre.

2 Les moyens à développer

2.1 Le recours à un planificateur

Le but d'un planificateur est de trouver dans un graphe d'états un chemin entre un état initial et un état final en respectant les contraintes relatives aux opérations à effectuer. Ce chemin correspond à un plan indiquant l'ordre dans lequel on doit appliquer les opérations.

On définit classiquement un plan comme un couple $(T, <)$ où T est un ensemble de tâches et $<$ une relation d'ordre strict sur T (éventuellement partiel). Une tâche est une instance d'opérateur représentant une classe d'actions dans le monde réel. Les opérateurs sont eux-mêmes définis par leurs préconditions et postconditions, où les préconditions doivent vérifier que l'instance de l'opérateur est applicable et les postconditions seront vraies après l'exécution. On admet alors qu'un plan contient au moins les deux tâches factices *début* et *fin*, où *début* est la tâche la plus petite pour $<$ et *fin* la plus grande dans le plan. Les postconditions de *début* sont une description de l'état initial, c'est-à-dire la situation à partir de laquelle débute l'exécution du plan ; les préconditions de *fin* sont une description (partielle) de l'ensemble des caractéristiques qui seront vraies dans la situation désirée.

L'activité du système découle alors de l'exécution du plan qui a été élaboré lors de la phase de planification. Nous pouvons illustrer cela en reprenant notre exemple du système UNIX.

2.1.1 Un exemple de décomposition d'action

Nous nous plaçons toujours dans le cadre de l'application du shell d'UNIX avec les commandes (ou opérations dans notre terminologie) modélisées par des opérateurs du modèle de l'application. Supposons que nous disposions d'un moyen quelconque qui, à partir d'un énoncé "Crée un répertoire public" puisse nous faire aboutir à la spécification² suivante d'un état final :

```
(:and (type D3 repertoire)
      (prop D3 PUBLIC a-pour-droit))
```

Parmi les différentes opérations que peut effectuer le shell de UNIX, nous modélisons `chmod`, destiné à changer les droits d'un fichier ou d'un répertoire, par l'opérateur `change-droits` ci-dessous :

```
(define (operator change-droits
  :parameters
  (?X ?Y ?Z)
```

2. Bien que la syntaxe corresponde à du LISP, nous pensons que les identificateurs sont suffisamment explicites pour qu'il n'y ait pas lieu d'alourdir le fil de l'argumentation en précisant leur définition. Nous définirons ce formalisme au chapitre 7 (§1.5).

2. Les moyens à développer

```
:preconditions
  (:and (nom ?X ?Y)
        (:or (type ?X fichier)
              (type ?X repertoire)))
:effects
  ((:effect (prop ?X ?Z a-pour-droit))))
```

Le formalisme utilisé pour définir ces opérateurs est décrit au chapitre 7 (§1.5). L'opérateur ci-dessus admet trois paramètres, ?X indiquant l'objet (fichier ou repertoire), ?Y son nom tel qu'il est utilisé dans la commande³, ?Z ses droits. Nous modélisons également `mkdir`, destiné à créer un repertoire, par l'opérateur `cree-repertoire` ci-dessous :

```
(define (operator cree-repertoire
  :parameters
    (?X ?Y)
  :preconditions
    (:not (:exists ?Z (:and (type ?Z repertoire)
                             (nom ?Z ?Y))))
  :effects
    ((:effect (nom ?X ?Y))
     (:effect (type ?X repertoire))))))
```

Les paramètres qu'utilise l'opérateur sont l'objet et son nom (pour la même raison que précédemment). La précondition vise à vérifier qu'il n'y a pas de repertoire existant portant le même nom. Il s'agit donc d'une condition de validité sur le nom à donner au repertoire.

Le plan issu de la planification indiquerait d'appliquer `cree-repertoire` avant `change-droits`. On arrive ainsi au même résultat que si l'on utilisait un script complexe comme dans les démarches classiques. La différence est que l'on ne situe plus au niveau d'une opération complexe `cree-repertoire-avec-droit` considérée en bloc, qui serait composée de `cree-repertoire` et `change-droits`.

2.1.2 Bref état de l'art dans le domaine de la planification

Un planificateur (ou générateur de plans) est un système capable sans intervention extérieure de définir son propre plan d'action, c'est-à-dire la séquence d'opérations élémentaires permettant d'atteindre un objectif à partir d'une situation d'origine [Haton et Haton, 1989]. La figure 3.3 indique les composantes essentielles d'un planificateur.

Le planificateur peut fonctionner suivant diverses méthodes, faisant appel aux principes de base du développement d'un espace des états ou de la réduction de problèmes en sous-problèmes.

Développement d'un espace des états. Cette méthode consiste à rechercher des opérateurs, les appliquer, et construire de proche en proche l'arbre de résolution. On peut

3. Il ne serait pas forcément nécessaire si l'on pouvait avoir des variables internes

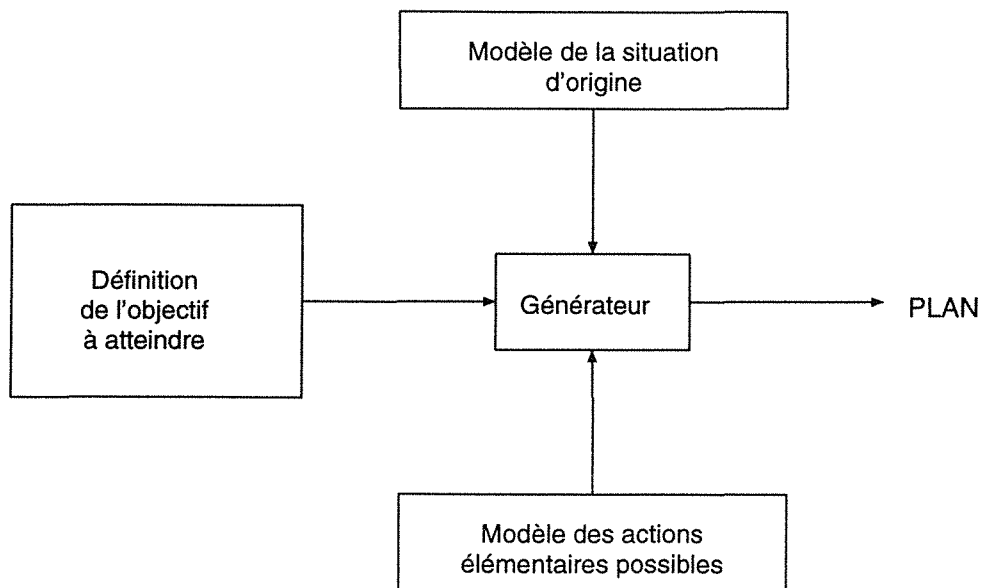


FIG. 3.3 - Structure d'un planificateur

s'y prendre de différentes manières :

- rechercher les opérateurs applicables à partir de la description de la situation initiale, et obtenir ainsi la description d'une nouvelle situation initiale ;
- rechercher les opérateurs conduisant à un état but et obtenir ainsi la description d'un nouvel état but ;
- faire une recherche bidirectionnelle.

Toutefois, une telle méthode pose un certain nombre de problèmes. Mentionnons d'abord le risque d'explosion combinatoire. Pour des problèmes complexes, il s'avère nécessaire de mettre en œuvre une bonne stratégie de recherche de solution guidée par des heuristiques. Enfin, signalons l'importance de la place mémoire que cela nécessite puisqu'il faut conserver toutes les modifications effectuées depuis l'état initial.

Analyse des fins et des moyens. Des exemples connus de tels systèmes sont GPS (General Problem Solver) et STRIPS [Fikes *et al.*, 1974]. La méthode utilisée revient de manière récursive ([Haton et Haton, 1989]) :

1. à calculer les différences entre l'état courant et le but courant (placé au sommet d'une pile des buts) grâce à une fonction d'évaluation spécifique.
2. à sélectionner un opérateur capable de réduire cette différence.
3. à prendre comme nouveaux sous-buts les préconditions de cet opérateur et à les empiler dans la pile des buts.

4. si les sous-buts sont vérifiés et si l'opérateur concerné devient applicable, à l'appliquer à l'état courant et poursuivre le processus.

Un défaut de cet algorithme est qu'il ne permet pas toujours de trouver une solution, même si celle-ci existe. On dit alors que la complétude n'est pas vérifiée. En effet, le planificateur échoue lorsqu'il rencontre des sous-buts interdépendants, c'est-à-dire quand le fait de réaliser un certain sous-but détruit un but précédemment réalisé et quand la reconstruction de ce dernier va détruire l'autre. Signalons aussi le cas où le planificateur se trouve devant un sous-but non réalisé et où aucun opérateur ne permet de le réaliser. Une solution est alors d'effectuer du retour arrière (*back-tracking*). Par rapport au développement d'un espace des états, la méthode a cependant l'avantage, dans le cas où elle fournit une solution, de réduire l'explosion combinatoire. Par contre, le caractère « sur-contraint » subsiste toujours, puisque l'on essaie de résoudre le problème en gardant à l'esprit la chronologie des événements dans le plan final.

Les méthodes hiérarchiques. Dans ces méthodes, on se place dès le début à un niveau d'abstraction élevé où le plan est décrit en termes lâches par des contraintes causales. Puis on essaie de décomposer une action générale en un réseau d'actions plus détaillé et on critique le plan obtenu en recherchant des actions redondantes et en tenant compte de la chronologie possible des sous-actions. Cela est répété jusqu'à obtention d'un plan linéaire. Un exemple d'implémentation est le système NOAH de [Sacerdoti, 1974].

Conclusion. Notre problème étant assez spécifique (il s'agit de trouver les opérateurs expliquant une modification qui interviendrait sur un objet), nous pouvons restreindre le problème de la planification à un cadre raisonnable. Afin de s'épargner une phase de planification, il est en outre tout en fait envisageable d'incorporer dans certains cas une pré-planification. L'intérêt est d'une part d'apporter un gain de temps et d'autre part, de résoudre à l'avance des problèmes trop difficiles. On rencontre notamment cette approche dans certains systèmes à base de connaissances adaptés à la résolution de problèmes. Dans son langage LISA, Delouis [1993] définit des structures *méthodes*, lesquelles s'apparentent à nos opérateurs, et des structures *buts*. Une méthode sera caractérisée par la description de ses paramètres, des résultats qu'elle produit, des *conditions* favorables à son utilisation, et enfin d'un pointeur vers l'entité opératoire (une fonction, une séquence de buts à réaliser, une base de règles, ...). Un but sera caractérisé par un *état-but*, des *critères de satisfaction*, un *contexte de réalisation* et des méthodes associées *a priori* (accompagnées de *préférences* indiquant leur priorité respective). C'est en ce sens que l'on peut parler de préplanification. Mais dans tous les cas, que le plan soit précalculé ou non, il est nécessaire d'aboutir à un état initial et un état final. La question concerne alors la manière d'y parvenir à partir d'un énoncé.

2.1.3 Un exemple de système utilisant une planification intégrée

Un peu en marge des principales classes de systèmes de dialogue de commande que nous avons passées en revue figure le système THJ⁴, l'un des premiers systèmes à notre

4. Il s'agit d'un acronyme chinois.

connaissance permettant de construire une séquence de commandes pour satisfaire la requête de l'utilisateur (requête en chinois par entrée vocale ou reconnaissance de caractères) [Xie *et al.*, 1989; Xie et Du, 1992].

Alors que les interfaces en langage naturel pour des applications de type commande souffrent de leur trop fort couplage avec une application particulière, le système THJ assure une bonne indépendance par rapport au domaine d'application (de commande). Par exemple, un énoncé de commande tel que

(3.8) Imprime le contenu du répertoire.

conduira au calcul de la séquence de commandes :

– **Sous UNIX :**

```
ls >TEMP-FILE
lpr TEMP-FILE
```

– **sous MS-DOS⁵ :**

```
dir >TEMP-FILE
print TEMP-FILE
```

alors qu'aucune ne pourra correspondre sous VMS.

La formation d'une séquence de commandes repose sur un raisonnement déductif orienté par le but. Une telle phase de raisonnement permet de couper court au morphisme de structure que les systèmes classiques présupposent⁴ entre le langage à consonance naturelle et le langage artificiel de commande. Il n'y a donc plus association directe à une commande (ou un script de commandes), mais à un couple de préconditions/postconditions.

Plus précisément, la planification consistant ici à parcourir un espace d'états, il s'agira de trouver un chemin parmi les différents chemins solutions. Ainsi, à l'énoncé de commande :

(3.9) Copie le contenu de ce répertoire < clic > dans celui-ci < clic >.

pourront correspondre (en MSDOS) au moins deux séquences de commandes différentes :

1. `xcopy ?REP1 ?REP2`
2. `mkdir ?REP2; copy ?REP1*. * ?REP2*. *`

Au niveau de l'algorithme de recherche du planificateur, choisir une solution plutôt qu'une autre nécessite d'évaluer une fonction de coût⁶. Les études menées en psycho-ergonomie sont fécondes pour élaborer de telles fonctions de coût (de priorités variables), que l'on

5. La syntaxe de MS-DOS est inspirée de celle d'Unix et de CP/M, ce qui explique l'air de famille...

6. En Intelligence Artificielle, l'adjonction de telles fonctions conduit à un algorithme nommé A*.

pourrait par exemple utiliser dans une interface langagière pour un système d'exploitation [Xie *et al.*, 1989; Xie et Du, 1992] :

Fonction 1 : Une séquence plus courte⁷ de commandes est meilleure qu'une séquence plus longue [Morki, 1984];

Fonction 2 : Une séquence de commandes avec un temps de réponse plus court est meilleure qu'une séquence avec un temps de réponse plus long [Hartson et Hix, 1989].

2.1.4 Conclusion : une mise en parallèle de deux temporalités

Nous avons montré que, dans notre cadre restreint, il était tout à fait envisageable (a) d'incorporer un planificateur, ou sinon (b) d'effectuer une pré-planification. Or, un planificateur nécessitant un état initial et un état final, la question concerne la manière d'y parvenir à partir d'un énoncé. Une solution est de construire linguistiquement l'état final auquel on aboutit ; suivrait alors la phase de planification, comme montré dans la figure 3.4.

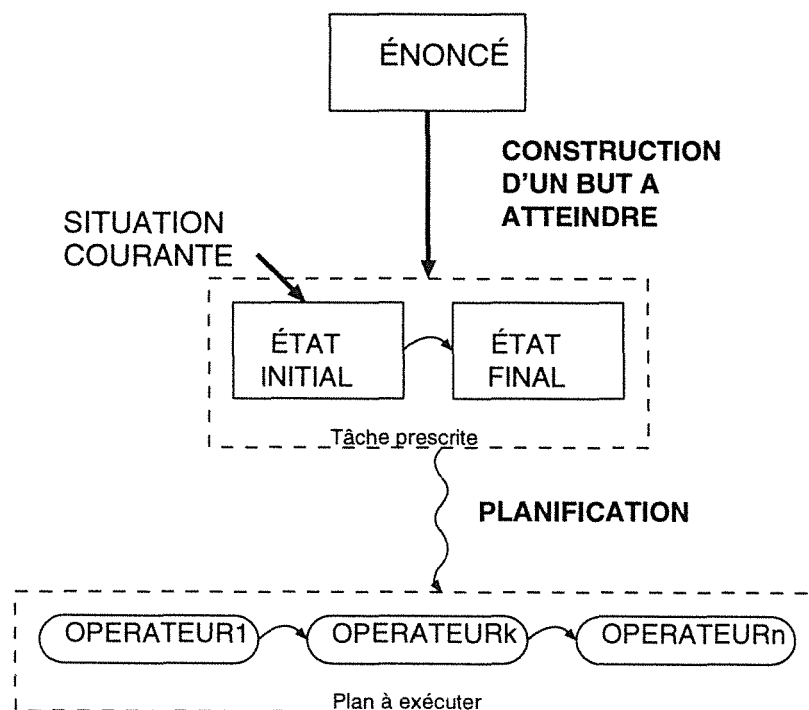


FIG. 3.4 - La mise en parallèle de deux temporalités

Il existe ainsi deux temporalités que l'on peut mettre en parallèle, l'une qui relève de la tâche prescrite, l'autre qui relève du plan à exécuter [Duermael *et al.*, 1994]. La première temporalité est donc relative à la construction d'états initial et final à partir de l'énoncé, alors que la seconde est relative au parcours du chemin pour aller de l'un à l'autre. Dans

7. En termes de nombre de commandes.

la suite de cette thèse, nous laissons de côté ce problème, spécifique au domaine de la planification, pour nous focaliser sur la première temporalité, correspondant à construire linguistiquement un but à atteindre.

2.2 Une analyse linguistique en termes de but à atteindre

2.2.1 Un exemple simple d'analyse

Dans le système THJ que nous avons commencé à décrire, à chaque classe d'énoncés, pour un ensemble donné de préconditions et de postconditions, on énumère la contribution de chaque élément le composant (fonctions objet, élément adverbial, ...).

```
(Word print v
((Property POSITION-CHANGE)
 (Object body)
 (Preaction (<position>(body)=place1))
 (Postaction (<position>(body)=place2))
 (Available (place1 place2 Entity))
 (Place1 (type POSITION)
 (Where (Adverbial))
 (Default ())
 (Scale ())
 (Location preaction)
 (value ((ON pp) (THE pp) (PRINTER n))))
 (Place2 (type POSITION)
 (Where (Adverbial))
 (Default (printer))
 (Schema ())
 (Location postaction)
 (value ((ON pp) (THE pp) (PRINTER n))))
 (Entity (type NAME)
 (Where (Object))
 (Default (file))
 (Schema ())
 (Location postaction)
 (value ((FILENAMES n))))
```

Cette description associe au verbe anglais *print* une rupture temporelle de propriété correspondant à son objet (appelée *position-change*). L'état d'avant est associée à *preaction* alors que celui d'après est associée à *postaction*⁸. La description mentionne ensuite comment les différents constituants de l'énoncé (à des emplacements indiqués par *Where* : Object, Adverbial) se distribuent en ces *preaction* et *postaction*. Notamment, il est spécifié par *type* la classe sémantique à laquelle ils doivent appartenir et par *value* les formes qu'ils doivent posséder.

8. Une composante spécifique raffine ces états pour la planification en utilisant des connaissances spécifiques au domaine d'application.

3. Conclusion

Sans dénigrer cette approche qui présente bien des qualités (cf. section 1 de ce chapitre), on peut toutefois lui adresser certains reproches. D'abord, les pré- et post-actions sont données uniquement par le verbe et ne sont pas construites. On aboutit donc à un lien direct entre le verbe et un couple d'états initial et final. Autrement dit, chaque verbe effectue une répartition entre les fonctions grammaticales qui correspondent à la spécification d'un état initial et celles qui correspondent à la spécification d'un état final. Pourtant, il semble, comme nous allons le voir, que les instructions véhiculées par certains constituants ne sont pas uniquement dictées par le verbe. Il est donc nécessaire de prendre en compte plus finement la dynamique temporelle sous-jacente à l'énoncé.

2.2.2 La dynamique temporelle de l'énoncé

En d'autres termes, nous recherchons à pouvoir déterminer ce qui est nouveau dans un énoncé par rapport à ce qui est déjà connu. Une réponse à cette question a pu être apportée par les psycholinguistes et les pragmaticiens du discours, qui ont beaucoup étudié l'opposition entre les informations données et les informations nouvelles dans le discours. Cela correspond à la distinction entre *thème* et *propos* (ou *rhème*).

Pour saisir la cohérence d'une succession d'énoncés, un auditeur doit généralement reconstituer lui-même un certain nombre d'informations qui ne lui sont pas fournies explicitement. Ainsi, sachant ce que le locuteur attend de lui, il se trouve d'après [Caron, 1993] dans une situation de résolution de problème. Il calcule en fonction des indices qui lui sont transmis l'information pertinente déjà connue, y rattache l'information nouvelle, et ainsi de suite. Selon Caron, l'opposition entre information ancienne et information nouvelle, thème et rhème (ou commentaire) traduit une mise en perspective bipolaire des divers éléments de l'énoncé définissant d'une part un en-deça (discursif ou situationnel, sur lequel l'acte d'énonciation se fonde) et d'autre part un au-delà (les possibilités d'enchaînement que cet acte définit). Ce que met en évidence le thème, qui est donc présupposé, c'est le fait que l'énoncé est orienté. Ainsi, l'acte d'énonciation s'inscrit dans une certaine situation et vise à la transformer. Cela revient à dire que l'opposition posé/présupposé marque la direction de cette transformation. Il est important ici de s'interroger sur le type d'énoncés considérés. Dans le cadre du dialogue de commande, nous nous situons au niveau des *actes directifs*. La spécificité par rapport au cas général est qu'on a une dynamique temporelle sur des objets de l'environnement applicatif plutôt que sur des états de connaissance.

3 Conclusion

À la différence d'autres systèmes qui emploient un planificateur, la planification est ici une partie intégrante de la référence à l'action. En effet, dans ces autres systèmes (par exemple SHRDLU [Winograd, 1972; Winston, 1977]), la planification n'intervient que pour rendre applicable un opérateur supposé dénoté par l'énoncé (ce sont alors les préconditions de la fonction que l'on veut appliquer qui fournissent l'état à atteindre à partir de l'état courant). Or, c'est là restreindre le rôle d'un planificateur, qui consiste précisément à sélectionner des opérateurs.

Nous avons pu donner des arguments empruntés à la psychologie et à la linguistique

pour soutenir l'idée selon laquelle *un énoncé de commande spécifie la construction d'un état final auquel on demande d'aboutir*. Non seulement une telle idée se tient, mais en plus elle présente des avantages par rapport aux démarches classiques, sans rien ôter en contre-partie. Au niveau du déroulement du dialogue, cela implique de pouvoir *vérifier* le résultat de l'application d'une action. C'est dès lors *l'état de l'application* qui s'avère important pour comprendre les enchaînements du dialogue.

Dans la partie qui suit, nous allons nous intéresser à la composante ayant pour finalité de construire à partir de l'énoncé de commande le but visé dans celui-ci. Le traitement des adjectifs et des groupes prépositionnels (nécessaire pour obtenir une régularité langagière) correspond à cette phase, en termes de spécification d'une configuration initiale ou finale. Le reste est relégué à la phase de planification. De plus, ce niveau permet d'effectuer une séparation plus nette entre ce qui relève de la langue et ce qui relève des contraintes d'exécution d'une action.

Deuxième partie

De l'énoncé au but qu'il décrit

Chapitre 4

Représentation des ruptures de propriétés véhiculées par l'énoncé

Sommaire

1	La spécificité de l'acte de langage directif	109
1.1	Les conditions associées aux actes de langage directifs	110
1.2	La forme impérative et ses conditions d'emploi	112
2	Rupture et évolutivité de propriétés d'objets du discours . .	113
2.1	Les différents types de propriétés	113
2.2	La nécessité d'introduire un modèle temporel	115
2.3	Importance de l'autonomie référentielle	121
2.4	Interaction entre référence aux objets et aux actions	122
3	Conclusion	126

Dans ce chapitre, nous nous appuyons sur certains repères bibliographiques pour montrer comment on peut représenter l'énoncé de commande en vue d'exploiter pleinement sa dynamique. Il s'agit en effet d'analyser un énoncé en termes de changements opérant sur des objets (des ruptures de propriétés). Nous montrons ainsi la nécessité de recourir à un modèle temporel fondé sur des objets, sur lequel viendra s'articuler la sémantique du langage employé. C'est donc à ce niveau que nous pourrons distinguer la situation courante de la situation visée.

1 La spécificité de l'acte de langage directif

Comme son nom l'indique, un acte de langage est un acte accompli par le langage, en prononçant un énoncé [Austin, 1962]. Searle représente un acte de langage sous la forme : $FI(cp)$ où FI représente la force illocutoire et cp le contenu propositionnel [Searle, 1972]. Par exemple, les deux énoncés

- (a) Tu quitteras la pièce.

(b) Quitte la pièce.

ont même contenu propositionnel, celui de *quitter la pièce* appliqué à l'auditeur ; mais le premier est une *prédiction* alors que le second est un *ordre*. Compte tenu du cadre dans lequel nous nous plaçons (cf. chapitre 1), nous ne considérerons que ce dernier cas, où un locuteur U (l'utilisateur) demande à un auditeur S (le système informatique) d'effectuer une action A. On parle alors d'*acte de langage directs*. De même que pour la modélisation que nous avons effectuée des actions, il existe des pré- et postconditions rattachées à l'accomplissement des actes de langage, que nous allons détailler.

1.1 Les conditions associées aux actes de langage directs

On peut étudier un acte du point de vue de la réussite pour celui qui le prononce, et du point de vue de son effet par rapport à l'auditeur, ce qui correspond aux conditions de satisfaction.

1.1.1 Conditions de réussite

Un acte de langage directif sera dit *réussi* si les conditions suivantes sont vérifiées :

1. Condition sur le contenu propositionnel

Il s'agit de la condition que doit satisfaire une proposition pour être un contenu propositionnel approprié d'un acte dans un contexte d'énonciation. Pour une requête, la proposition doit prédire un acte futur.

2. Conditions préparatoires

Il s'agit des *états de chose* qui doivent être vrais avant que le locuteur S effectue l'acte de langage. Pour une requête :

- U croit que S est en mesure d'effectuer A
- Il n'est pas certain, ni pour U, ni pour S, que S serait conduit de toute façon à effectuer A de lui-même.

3. Conditions de sincérité

Il s'agit des états psychologiques que l'énonciateur doit posséder à propos du contenu propositionnel s'il effectue un acte en étant sincère (c'est-à-dire si et seulement si il a l'état psychologique¹ qu'il exprime). En particulier, pour une requête, U désire que S effectue A.

4. Condition essentielle

Il s'agit du but de l'acte de langage. Pour une requête, cela revient à essayer d'amener S à effectuer A.

1. Un état psychologique est une notion très problématique. Il n'est pas question que le système s'en fasse une représentation. La condition de sincérité exprime que l'on doit considérer l'énoncé avant tout sans chercher à spéculer sur ce qu'il peut traduire du point de vue du locuteur.

1. La spécificité de l'acte de langage directif

C'est notamment au niveau de l'interprétation qui a pu être effectuée des conditions préparatoires que l'on entr'aperçoit le pas qui a pu être franchi par certains systèmes, dits orientés par la tâche. Les conditions préparatoires ont pu ainsi être interprétées comme la règle normative selon laquelle l'utilisateur veut uniquement ce que le système est en mesure de réaliser.

1.1.2 Conditions de satisfaction

Les conditions de satisfaction concernent l'effet de l'acte de langage. De manière générale, un acte de langage est satisfait si et seulement si :

1. son contenu propositionnel est vrai.
2. il l'est suivant la direction d'ajustement propre à son but illocutoire (tel qu'il apparaît dans la condition essentielle).

Dans le cadre des actes de langage directifs où nous nous plaçons, cela correspond à ajuster le monde aux mots par le biais des actes de l'auditeur, c'est-à-dire rendre *cp* vrai.

L'acte de langage est alors modélisé de la même façon qu'un opérateur en Intelligence Artificielle, avec un ensemble de préconditions, postconditions et un corps. Allen et Perrault [Allen et Perrault, 1980] définissent ainsi un acte de langage, :

« A *speech act* is an intentional action that has as parameters a speaker (i.e., the agent), a hearer, and a propositional content, and whose execution leads to the production of an utterance. Their preconditions and effects are defined in terms of the beliefs and wants of the speaker and hearer. » (p. 153)

« There is a 'precondition' on every intentional action that the agent (speaker) must want to perform the action (the want precondition of the act). » ([Allen et Perrault, 1980] p. 164)

Ainsi, pour un ordre (une requête), on aurait :

REQUEST(speaker, hearer, action)
effect : hearer WANT (hearer DO action)

Les préconditions, correspondant aux conditions préparatoires et de sincérité, seraient alors les suivantes :

speaker WANT *cp*
speaker BELIEVE \neg *cp*

Les postconditions, correspondant aux conditions de satisfaction, seraient les suivantes :

speaker BELIEVE *cp*

Lorsque ces conditions sont réunies, cela correspond, au niveau des croyances du locuteur, d'un passage de $\neg cp$ à *cp*. Dans le cas des actes de langage directifs, le prototype au niveau de la forme est l'utilisation de l'impératif. C'est pourquoi nous allons commencer par étudier la forme impérative et ce que présuppose son emploi.

1.2 La forme impérative et ses conditions d'emploi

Comme nous avons pu le voir dans le chapitre 1, l'acte de langage directif n'est pas spécifique à l'usage de l'impératif. Pour demander à quelqu'un de fermer une porte, on peut aussi bien dire :

(4.1) Ferme la porte !

(4.2) Je voudrais que tu fermes la porte.

(4.3) Pourrais-tu fermer la porte ?

Dans ce dernier énoncé, le locuteur ne s'attend pas à ce que son interlocuteur lui réponde par oui ou par non, comme s'il posait une question ouverte concernant ses capacités (« Oui, je peux le faire . . . »). Il s'agit d'un *acte de langage indirect*. On entend par là le fait qu'une phrase, dont l'indicateur correspond à une force illocutionnaire donnée (ici une question), est en fait utilisée avec une force illocutionnaire tout à fait différente (un ordre dans ce cas) [Moeshler et Reboul, 1994]. Que l'interlocuteur ferme effectivement la porte procède d'une inférence [Moeshler et Reboul, 1994, p.210] faisant appel aux principes de coopération conversationnelle et des conditions de satisfaction des actes de langage. Mentionnons aussi l'existence d'une analyse alternative faisant appel à la notion de *convention d'usage* [Morgan, 1978]. Or, nous nous intéressons à ce qui est spécifié directement par la forme de l'énoncé, et non à ce qui peut être déduit (à partir de ce qui a été spécifié). Pour autant, dans les dialogues que nous considérons, l'impératif est majoritaire. Aussi allons nous plus particulièrement l'étudier.

Guillaume [1970] ne considère pas l'impératif comme un mode de la langue à part entière (tel que le sont le subjonctif ou l'indicatif), mais comme une forme du dialogue :

« L'impératif est une manière de parler qui vise à provoquer, chez le sujet écoutant, l'accomplissement d'un acte que le verbe indique. » (p. 47)

Une première condition d'emploi de l'impératif est que le locuteur assume que l'action soit possible pour l'interlocuteur. Ainsi, toujours selon Guillaume :

« Par suite, l'emploi de cette manière de parler n'a de raison d'être qu'on [le locuteur] sait exister chez le sujet à qui l'on s'adresse les pouvoirs indispensables à l'exécution de l'acte dont on réclame de lui l'exécution. »

Dans les restrictions relatives à l'emploi de l'impératif, on suppose que l'action peut être réalisée. Par exemple, pour « Chante ! » cela signifie d'une part que le locuteur suppose que son interlocuteur ne soit pas muet et qu'il ne soit pas déjà en train de chanter. Une autre conséquence sera qu'à partir d'objets de la situation, le locuteur puisse décrire les (post)conditions pour que le terme de l'action soit atteint.

Nous faisons donc l'hypothèse que l'impératif spécifie l'atteinte d'un état final distinct. Il s'agit en quelque sorte d'une rupture temporelle véhiculée par l'impératif. Dans notre cas, l'interlocuteur ne chante pas au début et c'est seulement ensuite qu'il se met à chanter. Il lui faut atteindre un état final distinct, que l'on pourrait nommer *être chantant*, selon les grammairiens de Port-Royal [Ducrot et Todorov, 1972]. Si l'on considère le cas d'un

verbe admettant un complément d'objet, par exemple *fermer* dans « Ferme la porte! », on aboutit aux mêmes conclusions concernant l'objet. Partant du principe que l'impératif spécifie un état final distinct à atteindre pour "la porte être fermée", on en déduit qu'elle doit être auparavant non fermée, c'est-à-dire ouverte.

L'impératif spécifie alors le passage d'une propriété P à une propriété $\neg P$ ². Nous nous plaçons ici dans le cadre d'une logique du tiers exclus pour exprimer qu'une propriété évolue. On retrouve une approche équivalente dans [von Wright, 1963] (d'après [Dowty, 1979, p. 75-76]) qui propose un formalisme de changement d'état pour représenter un événement. Il s'agit d'une logique propositionnelle classique à laquelle Von Wright rajoute un opérateur dyadique³ T (appelé « Et Ensuite »⁴). Dans le calcul- T , toutes les formules peuvent être réduites à l'un des quatre types élémentaires suivants :

- $\neg pTp$ « l'état p survient »
- $pT\neg p$ « l'état p est détruit, arrive à son terme »
- pTp « l'état p se maintient, continue à avoir lieu »
- $\neg pT\neg p$ « l'état $\neg p$ se maintient » ou « l'état p échoue à survenir »

L'analyse des événements que fait Von Wright peut se rapprocher des verbes abstraits introduits par Lakoff [Lakoff, 1965] :

- $BECOME(p) \equiv \neg pTp$
- $END(p) \equiv pT\neg p$
- $REMAIN(p) \equiv pTp$

Seuls les deux premiers ($BECOME$ et END) traduisent une évolution de propriétés, comme peut le spécifier un énoncé de commande. Selon l'analyse que nous avons faite des conditions d'emploi de l'impératif, la force illocutoire des actes directifs provoquerait le passage d'une propriété p (ou $\neg q$) en une propriété $\neg p$ (resp. q). Le problème se pose alors en termes de capacité d'évolution de propriétés relatives à des objets.

2 Rupture et évolutivité de propriétés d'objets du discours

2.1 Les différents types de propriétés

Il est toujours problématique d'énumérer les caractéristiques ou propriétés relatives à un objet. Un objet possède un nombre indéfini de propriétés, que ce soit à travers toutes

2. En posant $P = \neg Q$, cela revient à spécifier de manière équivalente un passage d'une propriété $\neg Q$ à Q .

3. D'arité égale à 2.

4. « And Next » dans l'original.

les connaissances dont on dispose et que l'on peut lui appliquer, ou bien à travers les observations que l'on peut effectuer dessus. Ainsi, selon [Jespersen, 1971, p. 74] :

« Quand on veut faire naître une idée très précise chez son interlocuteur, on s'aperçoit que cette idée est extrêmement complexe, qu'elle présente un très grand nombre de caractéristiques et qu'il serait impossible de toutes les énumérer, même si l'on avait l'éternité devant soi. »

Cela n'est pas sans poser des problèmes au niveau du langage. Chaque objet du monde présente un ensemble de caractéristiques unique qui le distingue des autres. Or, on ne saurait faire correspondre à chacun de ces objets un mot du langage. C'est à ce titre que Jespersen considère que les mots introduisent des caractéristiques moyennes sous la forme d'une abstraction tendant à l'économie⁵.

Eco [1975] (p. 176) nous fait en outre remarquer qu'il existe des propriétés nécessaires (d'héritage, par exemple qu'une Rolls soit un véhicule, et même une automobile) et des propriétés accidentelles (par exemple qu'elle soit britannique, luxueuse). La différence entre propriétés nécessaires et propriétés accidentelles procède d'un effet d'optique. Si l'on veut donner une idée de ce qu'est une Rolls, on ne mentionnera pas qu'il s'agit d'un contenant mobile, qui a en plus un moteur. Toutes ces propriétés sont implicites par le fait qu'il s'agit d'un véhicule automobile.

Si l'on développe 4.1 (a), on obtient une représentation plus méticuleuse (b). Mais si l'on voulait être encore plus méticuleux, il faudrait expliciter à leur tour *contenant*, *mobile*, *moteur*, . . . , et ce jusqu'à l'infini. C'est pour cela qu'on utilisera un hyperonyme tel qu'*automobile* pour exprimer ce qu'est une Rolls. Ainsi, on ne peut pas expliciter l'ensemble des propriétés ; nombre d'entre elles sont implicites, virtuelles, déductibles. Néanmoins, il intervient des propriétés explicites : ce sont celles qui proviennent de l'énoncé.

Notre thèse est en effet que l'on explicite les propriétés qui vont intervenir pour caractériser l'état final. Par là-même, on spécifie tel ou tel aspect des caractéristiques de l'objet que l'on va actualiser. Prenons un exemple :

(4.4) Dessine un cercle au-dessus du carré.

C'est ici la propriété d'emplacement final qui est mise en évidence : *être au dessus du carré* appliqué au cercle. Notons que l'on peut décrire de plusieurs manières le même état de choses final. Les propriétés mises en jeu ne sont pas intrinsèques mais reflètent un point de vue de l'utilisateur. Ce dernier aurait tout aussi bien pu dire :

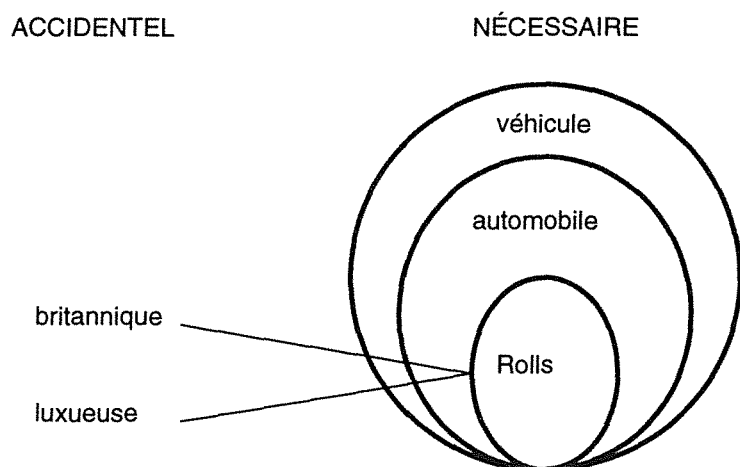
(4.5) Dessine un cercle à droite du rectangle.

Toutefois, c'est toujours par rapport au cercle qu'on exprime la propriété le reliant aux autres objets.

Nous allons montrer que l'on ne représente pas toutes les propriétés mais seulement celles qui changent dans le temps, à travers ce que spécifie le locuteur. Dès lors, le recours à un modèle temporel devient inévitable.

5. Cette conception du lexique a resurgi bien plus tard dans la théorie du stéréotype [Putnam, 1975] et, sous une autre forme, dans la sémantique des prototypes (cf. pour une synthèse [Kleiber, 1990]).

(a) Spécification des propriétés accidentelles



(b) Graphe d'héritage

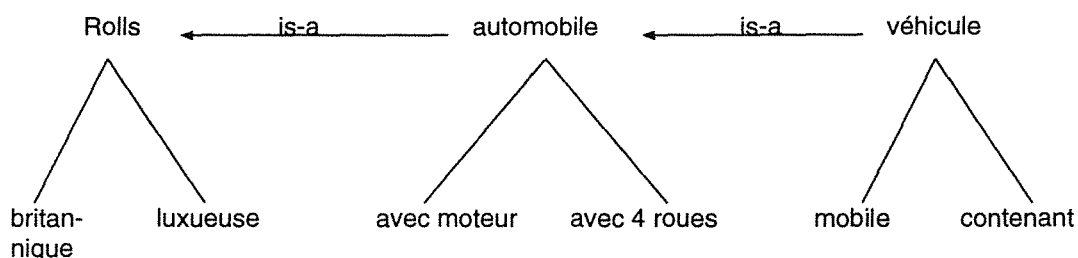


FIG. 4.1 - Propriétés nécessaires et accidentelles

2.2 La nécessité d'introduire un modèle temporel

Il est possible de prendre en compte le temps sans introduire pour autant de primitives temporelles, comme en témoigne le calcul des situations [McCarthy et Hayes, 1969], l'une des premières théories utilisées en Intelligence Artificielle pour modéliser des univers d'objets évoluant dans le temps. Le calcul des situations a été très utilisé pour modéliser les actions séquentielles d'un agent dans un monde statique, l'implantation la plus connue étant le modèle d'espace d'états dans lequel le monde est décrit par une base de données contenant l'ensemble des faits vérifiés dans la situation courante, et les actions sont simulées pour des ajouts et des destructions de faits de la base de données [Fikes *et al.*, 1974]. Cela peut constituer un moyen adapté pour surmonter un inconvénient lié aux modèles qui décrivent le monde en termes de situation globales. Dans ces modèles, le problème est qu'il faut décrire pour chaque nouvelle situation les faits qui ont changé, mais également ceux qui sont restés identiques à eux-mêmes – c'est ce que l'on a coutume d'appeler le *frame problem*.

Parmi les problèmes qui subsistent, citons aussi celui de l'ordre temporel strict et total que cela implique au niveau des actions, l'impossibilité de modéliser le futur et le passé

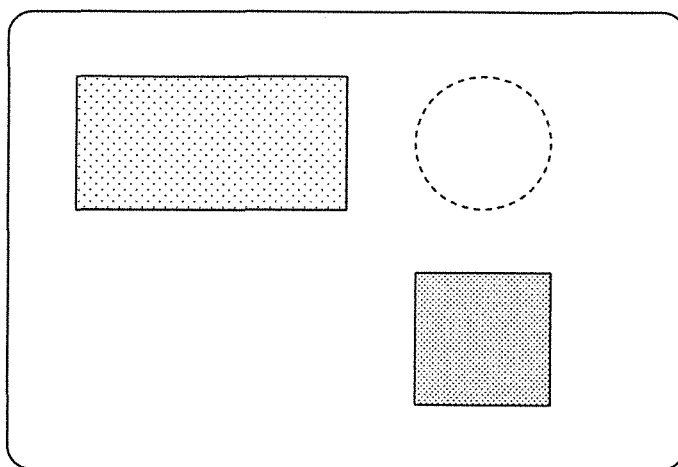


FIG. 4.2 - *Les points de vue de localisation*

(car le monde est tel qu'il est décrit dans la base dans sa globalité et aucune autre information n'est disponible). Enfin, en ce qui nous concerne plus directement, on peut regretter une grave lacune de ce modèle, qui ne se focalise que sur la composante prospective de l'action, le résultat auquel on aboutit, laissant totalement de côté la composante effective relative au déroulement proprement dit. Tous ces problèmes nous incitent donc à chercher d'autres représentation du temps. Une première intuition peut être de le représenter mathématiquement, selon une droite réelle, et considérer des points ou des intervalles.

2.2.1 Le temps sur un ensemble d'instants

L'approche consiste alors à représenter le temps comme une succession d'instants prenant leur valeur dans l'ensemble des nombres réels. Mc Dermott [1982] propose une ontologie de faits et d'événements ainsi qu'une structure temporelle isomorphe à la droite des réels. La structure temporelle, constituée par un ensemble d'instants, est mise en relation avec un ensemble d'états par une fonction d de datation ; chaque état S a donc une date $d(S)$. L'ensemble des états est muni d'une relation binaire de précédence dont les propriétés donnent à l'ensemble une structure ramifiée vers le futur (plusieurs futurs possibles, un seul passé). À partir des états, Mc Dermott définit les éléments de son ontologie, qui sont les faits et les événements.

Un fait est vrai ou faux à chaque état. Formellement, un fait est identifié à un sous-ensemble de l'ensemble des états. La formule $T(S,f)$ indique que le fait f est vrai en l'état S . Un événement peut avoir lieu sur un intervalle d'état $[S1, S2]$. Notons ici que la structure temporelle sous-jacente permet de définir des intervalles d'états comme un ensemble totalement ordonné et convexe d'états. Formellement, un événement est identifié par un ensemble d'intervalles d'états. La formule $OCC(S1, S2, e)$ est vraie si l'intervalle $[S1, S2]$ appartient à l'ensemble dénoté par l'événement e (c'est-à-dire si l'événement e a lieu sur l'intervalle $[S1, S2]$). De façon similaire à $T(S,f)$, on indique par la forme

TT(S1,S2,e) qu'un événement est vrai sur un intervalle. Le système de McDermott apporte

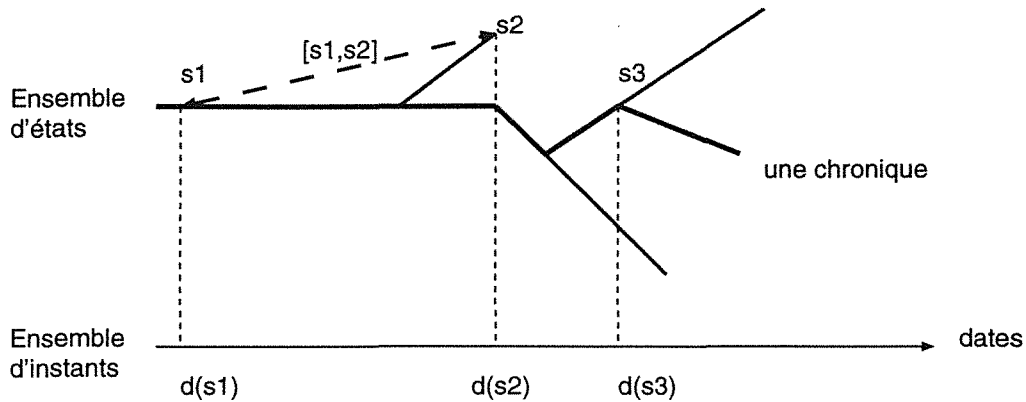


FIG. 4.3 - L'arbre des états de Mc Dermott

certes des solutions aux problèmes posés par le calcul des situations, mais encore faut-il savoir si le raisonnement qu'il met en œuvre peut s'appliquer au langage naturel. Nous allons montrer que nous devons pour cela abandonner la notion primitive d'instant pour laisser place aux intervalles.

2.2.2 Le temps relativement à un intervalle

Nous pouvons reprendre à titre d'illustration l'exemple de [Romary, 1989] (p.110) avec un prédicat $TRUE^6(t1, t2, COLOR(HOUSE17, RED))$ tel qu'on peut en rencontrer dans des logiques temporelles. L'interprétation d'une telle formule devrait fournir soit la réponse *vrai* soit la réponse *faux*. Mais il est facile d'imaginer une situation dans laquelle aucune des deux valeurs ne serait acceptable (cf. figure 4.4).

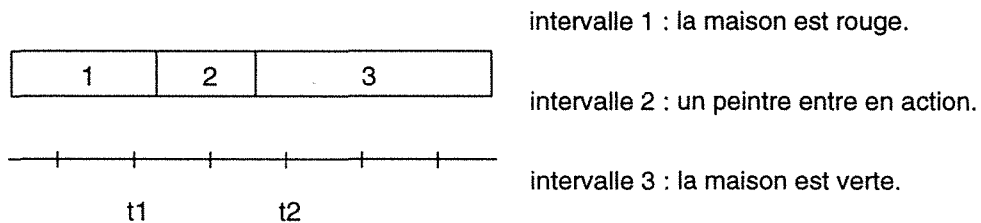


FIG. 4.4 - Un univers indéterminé pour le prédicat : $TRUE(t1, t2, COLOR(HOUSE17, RED))$

Si maintenant on posait à un quidam une question du genre « Quelle était la couleur de la maison entre $t1$ et $t2$? », une réponse naturelle qu'il pourrait donner serait "Ah, mais... la couleur, elle a changé. Un peintre est venu entre temps." En fait, on n'attend pas une valeur de vérité mais une réponse satisfaisante pour l'utilisateur. Dans cet exemple, le temps ne se conçoit pas *ex nihilo* (par exemple $t1$ et $t2$) mais par rapport à un changement relativement à un objet. Plus précisément, la couleur est définie sur $I1$ (avant), $I3$ (après) et non sur $I2$ (pendant). Les bornes des intervalles sont en effet délimitées par les ruptures que

6. noté TT chez McDermott.

subissent les objets. La notion de temps ne saurait se concevoir simplement par rapport à des points quelconques sur une droite, mais par rapport à des intervalles bien déterminés.

On peut rencontrer en IA des systèmes temporels reposant sur une structure primitive d'intervalles, parmi lesquels celui de [Allen, 1984].

Relation	Symbole	Symbole inverse	Exemple
X before Y	<	>	X _____ Y
X equal Y	=	=	X _____ Y
X meets Y	m	mi	X _____ Y
X overlaps Y	o	oi	X _____ Y
X during Y	d	di	_____ X _____ Y
X starts Y	s	si	X _____ Y
X finishes Y	f	fi	_____ X _____ Y

FIG. 4.5 - Les 13 relations possibles d'Allen entre deux intervalles

L'objectif premier de Allen avait été de montrer l'inadéquation des systèmes fondés sur des instants temporels. Mais, ces derniers se sont malgré tout réintroduits à travers les relations *meets*, *starts*, *finishes* (et leurs relations inverses). Par exemple, la relations *meets* signifie que l'intersection entre X et Y est un intervalle réduit à un point. Suivant une démarche analogue, Dowty a rejeté les sémantiques fondées sur des moments au profit d'une sémantique fondée sur les intervalles. Dans une sémantique fondée sur les moments, on pourrait définir [BREAK Φ] comme un changement de $\neg\Phi$ (à un moment t_i) à Φ (à un moment t_{i+1}). Si ces conditions peuvent sembler appropriées pour des verbes impliquant un changement d'état instantané (comme par exemple *reconnaître que P*, *découvrir que P*), cela est problématique pour des changements progressifs (comme *construire une maison*, *traverser un désert*). Dowty propose alors de considérer qu'un énoncé faisant appel à un verbe abstrait BECOME est vrai sur un intervalle, quelle que soit sa taille, si cet intervalle est borné à une extrémité par un état de choses particulier et à l'autre extrémité par un autre état. Plus formellement :

[BECOME Φ] est vrai sur un intervalle I ssi il existe un intervalle J contenant la borne initiale de I tel que $\neg\Phi$ soit vrai sur J et il existe un intervalle K

contenant la borne finale de I tel que Φ soit vrai sur K⁷.

Sur un diagramme linéaire du temps, [BECOME Φ] serait vrai dans la situation suivante :

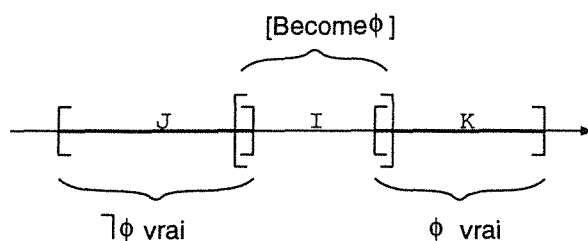


FIG. 4.6 - Conditions de vérité du verbe abstrait BECOME

Par rapport à nos besoins, le système de Allen propose des relations temporelles qui sont trop précises et mal délimitées. Dans les théories de la sorte, un intervalle peut être considéré *ex nihilo*, sans aucune contrainte imposant la délimitation de ses bornes. Or, nous avons vu que ce sont les ruptures que subissent les objets qui délimitent les bornes des intervalles. On se dirige alors vers un modèle à base d'intervalles dont les bornes seraient délimitées par l'existence même des objets et de leurs propriétés.

2.2.3 Le modèle minimal temporel adapté

Le modèle temporel introduit par Romary [1989] pour représenter le temps s'appuie sur deux relations seulement entre des objets unitaires appelés zones temporelles. Les zones temporelles sont des structures contenant des informations, et peuvent être de deux types. Certaines représentent l'existence entière d'objets et d'autres contiennent seulement des informations sur ces objets (incluses temporellement dans l'objet auquel elles se rapportent). Les deux relations possibles sont :

- l'inclusion (notée *in*)

Z1 est incluse dans Z2 si et seulement si les informations contenues dans Z2 ont une durée plus longue que celles contenues dans Z1.

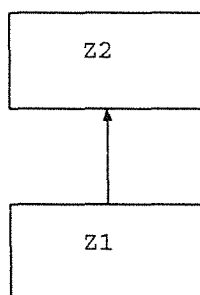


FIG. 4.7 - L'inclusion temporelle

7. Dowty précise qu'une implicite conversationnelle serait qu'il n'existe pas d'intervalle inclus dans I vérifiant ces propriétés.

- la précédence (notée *prec*)

Z1 précède Z2 si et seulement si Z1 est antérieur à Z2 et les informations contenues dans Z1 ne peuvent pas être valides en même temps que celles contenues dans Z2.



FIG. 4.8 - La précédence temporelle

Au delà de ces définitions, il est possible de caractériser formellement ces deux relations⁸. Soit \mathcal{Z} l'ensemble des zones temporelles. Alors *in* et *prec* sont des éléments de $\mathcal{Z} \times \mathcal{Z}$ vérifiant les propriétés suivantes :

$$\forall A, B, C \in \mathcal{Z}$$

- antiréflexivité

$$\frac{A \text{ in } A}{\square} \text{ et } \frac{A \text{ prec } A}{\square}$$

- antisymétrie

$$\frac{A \text{ in } B, B \text{ in } A}{\square} \text{ et } \frac{A \text{ prec } B, B \text{ prec } A}{\square}$$

- transitivité

$$\frac{A \text{ in } B, B \text{ in } C}{A \text{ in } C} \text{ et } \frac{A \text{ prec } B, B \text{ prec } C}{A \text{ prec } C}$$

- exclusivité

$$\frac{A \text{ in } B, B \text{ prec } A}{\square} \text{ et } \frac{A \text{ in } B, A \text{ prec } B}{\square}$$

- règles de priorité

$$\frac{A \text{ in } B, B \text{ prec } C}{A \text{ prec } C} \text{ et } \frac{A \text{ in } B, C \text{ prec } B}{C \text{ prec } A}$$

Comme montré dans [Romary, 1991], ces relations et les contraintes logiques sur les informations contenues dans les zones doivent être imbriquées dans le raisonnement. Cela explique pourquoi l'inclusion temporelle peut représenter les subordinations que nous devons prendre en compte lorsque l'on représente un état final.

Les objets permettent de structurer les propriétés. Ainsi, une rupture de propriété sera représentée par le graphe de zones temporelles suivant (cf. figure 4.9) :

8. La barre de fraction $\frac{A_1, \dots, A_n}{B}$ signifie que de A_1, \dots et A_n , on déduit B . \square signifie « Absurde ».

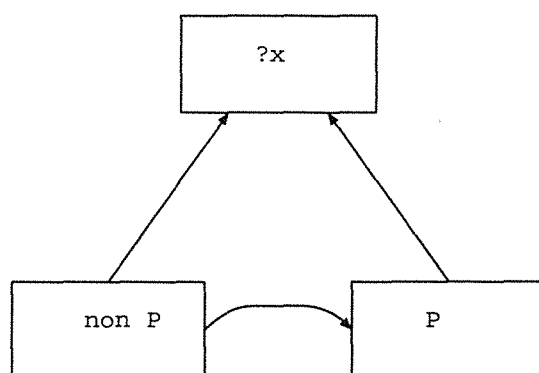


FIG. 4.9 - Représentation d'une rupture temporelle

Un lieu peut lui-même être considéré comme un objet, fournissant une structure. L'avantage est de contourner le *frame problem* en se limitant aux objets structurés. Cela rejoint la conception de philosophes comme Leibniz [Ishiguro, 1990], pour lequel notre conception du temps dépend de notre perception des changements que subissent les objets autour de nous. Nous allons donc examiner en quoi on peut affirmer que le temps n'a d'existence que par rapport aux objets.

2.3 Importance de l'autonomie référentielle

La notion d'autonomie référentielle est directement liée à ce problème de la dépendance du temps par rapport aux objets. Nous employons le terme d'*autonomie référentielle* dans un sens différent de celui de [Milner, 1982], selon lequel seuls les syntagmes nominaux qui ont un sens lexical (les descriptions indéfinies, définies, les syntagmes nominaux démonstratifs pleins, à l'opposé des pronoms) sont dotés d'une autonomie référentielle. Ce sont des critères d'ordre aspecto-temporel qui vont nous permettre de délimiter les formes linguistiques dans un énoncé, selon qu'elles véhiculent ou non une dynamique, une progression. Il est donc important de différencier ce qui est statique (les entités) de ce qui est dynamique (l'évolution due au déroulement du procès).

Le principe de base sur lequel reposent les zones temporelles de Romary est que si A est inclus temporellement dans B, alors A est moins statique que B, et de ce fait est moins autonome référentiellement. Il existe des marques linguistiques qui en témoignent : la classe grammaticale à laquelle appartient un constituant est un indice de son autonomie [Jespersen, 1971; Langacker, 1991]. Verbe et adjectif ne peuvent pas être autonomes, devant s'appliquer sur quelque chose de plus stable. En revanche, le nom peut ne s'appliquer à rien, auquel cas il est autonome⁹.

La subordination des prédicats (issus de verbes, adjectifs, noms prédicatifs) aux objets se justifie très bien si l'on admet que l'existence-même des objets est due à leur extension temporelle. En effet, les prédicats intériorisent du temps mais n'ont pas d'extension temporelle propre : ce sont les objets sur lesquels ils portent qui leur confèrent une telle extension. Alors que depuis Aristote, l'indication temporelle de l'énoncé était classique-

9. Mais il peut aussi s'appliquer à une entité, auquel cas il a un statut assimilable à celui d'un prédicat, de même titre que le verbe ou l'adjectif. Citons comme exemple : *couleur*.

ment rapportée au verbe¹⁰, le logicien Bolzano est le premier à avoir proposé, dans sa *Grundlegung der Logik* (1837), de rapporter au groupe nominal (parfois réduit au nom) l'indication temporelle de l'énoncé :

« Une proposition de la forme

L'objet A – a dans le temps t – la propriété b ,

pour que ses éléments constitutifs apparaissent avec la plus grande distinction possible, doit être exprimée ainsi :

L'objet A dans le temps t – a – la propriété b .

Car ce n'est pas dans le temps t qu'il arrive que la propriété b soit jointe à l'objet A ; mais à l'objet A , dans la mesure où il est pensé comme quelque chose qui se trouve dans le temps t (donc pourvu de cette détermination), nous joignons la propriété b (*op. cit.* ; citation empruntée à [Rousseau, 1993]). »

On aboutit alors à l'espace-temps comme propriété nominale, ce qui constitue la thèse de Quine [Quine, 1960]. Il considère en effet le temps comme totalement inessentiel à la proposition et propose de le remplacer par des quantificateurs renvoyant à des époques¹¹ (« au moment t », « avant t », « après t », ...) qu'il applique aux objets du discours. Ainsi, Quine propose de représenter l'énoncé :

(4.6) Bertrand a épousé Marie-Claire.

par

(4.7) $(\exists t)$ (t est avant l'époque *maintenant* et Bertrand à l'époque t épouse Marie-Claire à l'époque t).

Cette analyse se trouve en outre confirmée par l'existence « d'objets au passé » que l'on rencontre dans certaines langues [Benveniste, 1974b] ou encore, en ce qui concerne le français, par l'existence de marqueurs morphologiques tels que *ex* (par exemple, *ex-femme*) [Reboul, 1993]. Ce point de vue présente le grand intérêt de poser l'existence de catégories spatio-temporelles pour le groupe nominal comme pour l'énoncé verbal. En fin de compte, cela revient à postuler une isomorphie verbo-nominale.

2.4 Interaction entre référence aux objets et aux actions

Nous avons déjà montré dans le chapitre 2 que l'utilisation du verbe seul pour déterminer l'action à effectuer s'avérait insuffisante. Nous avons établi que la détermination de l'action à effectuer dépendait du type de l'objet sur laquelle elle portait, sans pour autant à avoir à en calculer complètement la représentation. Cela montre que la catégorisation de l'objet sur lequel porte l'action a bel et bien un rôle dans la détermination même de celle-ci. Mais le problème est encore plus compliqué qu'il n'y paraît. Comme nous allons le

10. Aristote définissait le verbe comme « ce qui ajoute à sa propre signification celle du temps ».

11. Quine définit une époque comme une portion du monde spatio-temporel à quatre dimensions.

voir, le prédicat verbal joue en plus un rôle dans le calcul de la représentation de l'objet, en concourant à délimiter la zone et les domaines qui lui correspondent.

Dans les deux énoncés :

(4.8) Ecris un A.

(4.9) Ecris un A majuscule gothique.

ce sont les connaissances graphiques qui entrent en jeu pour délimiter la zone relative au A. En revanche, dans les énoncés :

(4.10) Prononce un A.

(4.11) Prononce un A bref.

ce sont les connaissances phonétiques qui entrent en jeu. Toutefois, nous observons un conflit sémantique dans les énoncés :

(4.12) * Prononce un A majuscule gothique.

(4.13) * Ecris un A bref.

Ces exemples nous montrent que les verbes ou les déterminants adjectivaux se comportent de la même façon par rapport au nom « A ». On retrouve sur cet exemple un argument [Rastier, 1987] en faveur d'une sémantique qui ne soit pas exclusivement référentielle :

« Les mots interagissent entre eux, se rajoutent mutuellement des contraintes, avant de dénoter. »

Gaiffe a notamment établi dans sa thèse [Gaiffe, 1992] que le prédicat apporte des restrictions purement sémantiques (et non référentielles) aux groupes nominaux. Nous voyons donc qu'il y a interdépendance entre référence aux objets et référence aux actions.

Intéressons-nous alors plus précisément à la référence aux objets, telle que la conçoit Gaiffe [Gaiffe, 1992; Gaiffe, 1993]. En fait, un nom commun ne suffit pas pour référer ; il ne donne qu'une image prototypique (cf. Guillaume et Kleiber, *op. cit.*). Ce n'est pas le nom tout seul qui réfère, mais accompagné d'un déterminant. Le déterminant lui fournit des instructions de calcul de référence en utilisant des contraintes sémantiques. L'auteur propose un schème de référence pour chaque type de référent, indiquant ainsi le procédé à suivre pour calculer la référence aux objets.

2.4.1 Schèmes de références

L'article défini : le N. Le schème correspondant illustre que dans un contexte discursif Q, on ne considère ce qui est N (i.e. l'objet peut porter le nom N), par opposition à ce qui est non-N. L'axe d'opposition $N \rightarrow \neg N$ constitue par définition une axiologie.

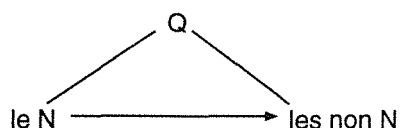


FIG. 4.10 - Schème de référence pour le N

L'article indéfini: un N. Par rapport au schème de *le N*, le schème de *un N* fait intervenir une axiologie supplémentaire. Il y a en effet deux manières possibles de considérer *un N*, comme explicité ci-dessous dans le cas de *un chat*.

1. un chat, par opposition avec un chien ou un canari, dans la classe des animaux domestiques: il s'agit par conséquent de la classe (N) des chats;
2. un chat particulier (s'opposant à d'autres chats), par exemple un chat persan (propriété P1) qui a été égaré il y a une semaine (propriété P2).

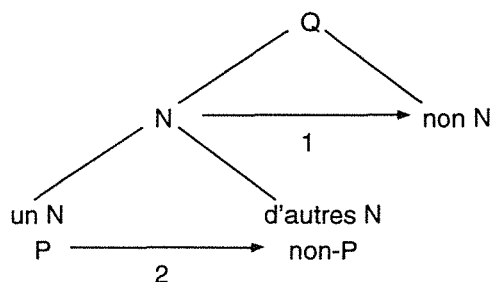


FIG. 4.11 - Schème de référence pour un N

Le démonstratif: ce N. À la différence de l'indéfini, le démonstratif ne peut pas atteindre la classes des N. De plus, alors que les propriétés spécifiques entrant dans le schèmes de *un N* étaient posées, celles de *ce N* sont présupposées.

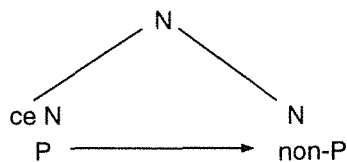


FIG. 4.12 - Schème de référence pour ce N

La question est maintenant de déterminer comment s'appliquent les prédicats (issus d'adjectifs ou d'adverbes) au niveau de ces schèmes de référence.

2.4.2 La participation des prédicats

On peut remarquer que certains verbes opèrent des sélections d'instructions de calcul de référence sur tel ou tel article. Par exemple, on peut dire qu'un verbe inchoatif implique quelque chose à créer et élimine ainsi certaines instructions de calcul référentiel. *Le* contient une instruction d'identification avec quelque chose de préexistant. Il peut y avoir des restrictions avec adjectifs, compléments de nom. Dans ce cas, il s'agit d'un raffinement du nom, une fois ce dernier déterminé. On rencontre là encore un ordre d'application. Par exemple, à l'énoncé :

(4.14) Dessine un cercle rouge.

Gaiffe ferait correspondre le schème de référence :

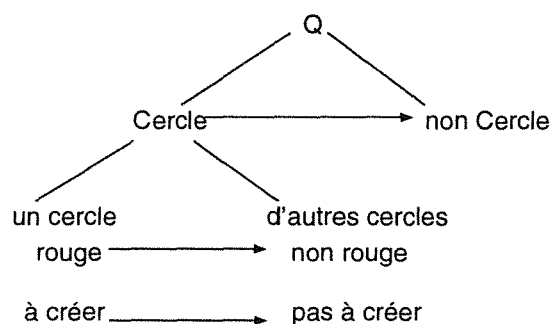


FIG. 4.13 - Schème de référence pour Dessine un cercle rouge

Il se rajoute donc une axiologie supplémentaire, provenant de ce que spécifie le verbe et venant se rajouter aux propriétés particulières du cercle : *à créer* → *pas à créer*. Mais on pourrait concevoir d'autres axiologies, comme par exemple avec *un N* en position sujet (cf. figure 4.14) :

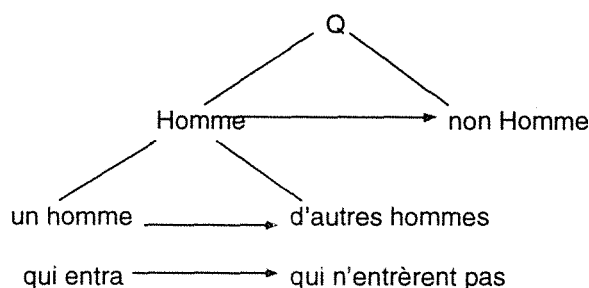


FIG. 4.14 - Schème de référence pour Un homme entra

Il se pose alors le problème de ce qui justifie l'individualisation du référent : *être créé, devoir l'être, ne pas l'être encore, ...*

3 Conclusion

Nous avons défini dans ce chapitre les caractéristiques minimales d'un modèle temporel adapté pour représenter les évolutions de propriétés que véhiculent les actes de langage directifs. Ces propriétés, de même que les objets sur lesquels elles s'appliquent, sont représentées par des zones temporelles d'autonomie relative. Deux zones temporelles en relation de précédence auront même autonomie, alors qu'une zone incluse dans une autre sera moins autonome. Toutefois, l'organisation de ces zones temporelles ne s'obtient pas directement à partir de la structure de l'énoncé. Il y a en effet, par exemple, une certaine différence d'autonomie du complément d'objet entre les expressions *manger une pomme* et *manger des pommes*. Dans le premier cas, il semble que l'on puisse individualiser une pomme particulière, tandis que dans le second, on ne peut même pas individualiser un ensemble. Dans ce dernier cas, tout se passe comme si le complément était intégré au verbe pour former le prédicat *Manger-des-pommes*, ce prédicat ne portant que sur le sujet. Il s'agit là d'une forme extrême de notre remarque concernant la participation du type du complément d'objet direct à la détermination du prédicat. Ainsi, il apparaît nécessaire de recourir à une étape intermédiaire plus linguistique orientant la construction de ces zones temporelles.

Chapitre 5

Analyse linguistique de l'énoncé en termes de ruptures de propriétés

Sommaire

1	Les marques linguistiques de la dynamique temporelle de l'énoncé	127
1.1	Les classes aspectuelles de verbes	128
1.2	Le participe passé comme expression d'un résultat	132
1.3	Conception sémantique de la transitivité	136
2	Les structures sous-jacentes de l'énoncé	138
2.1	La dynamique de l'aspect verbal selon Guillaume	139
2.2	Le verbe comme élément structurant de l'énoncé	140
2.3	Fonctions grammaticales et structures de dépendance	153
3	Conclusion	161

Nous nous sommes donnés dans le chapitre précédent les outils pour représenter des ruptures de propriétés sur des objets du discours. Or, il existe des indices au niveau de l'énoncé qui spécifient comment ces propriétés évoluent dans l'énoncé. C'est ce que nous appelons les marques linguistiques de la dynamique temporelle de l'énoncé et que nous développons dans la première section de ce chapitre. La question que nous traitons ensuite concerne la manière selon laquelle ces marques vont intervenir lors de l'analyse de la structure de l'énoncé.

1 Les marques linguistiques de la dynamique temporelle de l'énoncé

La linguistique générale a beaucoup étudié les caractéristiques que l'on rencontre à travers les diverses langues (dans leur morphologie ou leur sémantique) qui caractérisent

la dynamique de l'énoncé. Il s'agit par exemple de l'aspect ou de la transitivité. Nous allons analyser comment ces diverses propriétés interviennent dans la spécification d'un état à atteindre.

1.1 Les classes aspectuelles de verbes

Il est usuel de considérer qu'un verbe dénote un *procès*. Ce terme a d'ailleurs été introduit par Meillet pour regrouper sous une appellation suffisamment large à la fois les états (statiques) et les actions (dynamiques). Mais comme nous allons le voir, il existe de nombreux tests pour mettre en évidence des classes sémantiques plus fines.

Si l'on reprend le formalisme logique qu'a introduit Von Wright [1963] pour représenter les états et leurs changements, on peut regrouper en deux classes les types élémentaires qu'il a mis en évidence :

1 pTp et $\neg pT\neg p$

2 $\neg pTp$ et $pT\neg p$

Dans la première classe, on a une perpétuation continue dans un état qui confère au déroulement une homogénéité. Par contre, dans la deuxième classe, on a une rupture de propriété qui confère au déroulement une hétérogénéité. On retrouve une classification semblable en linguistique sous l'appellation d'*aspect verbal*.

1.1.1 La catégorie de l'aspect

De manière générale ([Jespersen, 1971] [Ducrot et Todorov, 1972] [Dowty, 1979, p.52] [Lyons, 1977, ch. 15.6]), on utilise le terme d'aspect verbal à diverses fins pour distinguer :

- si l'on réfère au début, au milieu, à la fin d'un procès ;
- le caractère répété ou unique d'un événement ;
- si le procès est achevé ou demeure potentiellement inachevé.

On peut trouver chez [Rousseau, 1993] une définition plus globale. L'aspect y est considéré comme une « prise de vue » sur l'état d'engagement (ou encore le degré d'avancement) d'un procès à un moment donné¹. En ce qui nous concerne, ce sont les aspects

1. *inchoatif* opposé à *non inchoatif* (concernant le début hypothétique du procès).
2. *terminatif* opposé à *non terminatif* (concernant la fin hypothétique du procès).

qui retiendront notre attention. En effet, ces aspects impliquent directement une rupture de propriété (p succède à $\neg p$, i.e. BECOME(p), ou $\neg p$ succède à p , i.e. END(p)) telle que

1. Un danger auquel échappe toutefois cette définition est de faire de l'aspect une notion sémantique trop englobante. Il est en effet important de distinguer la notion même d'aspect des inférences qu'elle permet d'effectuer à partir du contexte. On évite ainsi d'inclure directement des notions telles que la durativité ou l'itérativité.

peut la spécifier un énoncé de commande. Nous cherchons donc à déterminer si un procès est hétérogène ou non, ou (ce qui est équivalent) s'il est délimité ou non, décomposable ou non. Dès lors, il devient intéressant pour nous de classer différents types de prédication selon des critères purement linguistiques.

1.1.2 Les classes sémantiques de verbe

Une partition que l'on peut facilement mettre en évidence est celle qui existe entre les verbes d'état et les verbes d'action, comme l'illustre la classification qu'a effectuée Lakoff [1965] entre prédications statives et prédications non statives. Le critère qu'il utilise est la possibilité d'employer la forme progressive (*to be + -ing* en Anglais ou *être en train de*). Il montre en outre que cette distinction ne s'applique pas seulement aux verbes, mais aussi à certains noms et adjectifs.

(5.1) *John is being careful vs. *John is being tall*

(5.2) *John is being a hero vs. *John is being a grandfather*

On a vu que l'on pouvait distinguer les propriétés (continues) selon leur nature constante ou monotone. Cela permet de faire à l'intérieur des procès homogènes la différence entre les états (ex. *connaître la solution*) et les activités (ex. *chercher la solution*). Concernant les procès hétérogènes, on pourrait citer des exemples tels que *élaborer la solution* ou *trouver la solution*. Vendler [1967] a affiné cette classification, par l'introduction d'*Aktionsarts* (mot allemand signifiant « modes d'action »), en employant des critères linguistiques supplémentaires. Parmi ceux-ci figure le test d'adjonction de compléments de temps comme *en X temps* ou *pendant X temps*.

(5.3) Il a connu la réponse <i>pendant 10 mn</i>	★Il a connu la réponse <i>en 10 mn</i>
Il a cherché la réponse <i>pendant 10 mn</i>	★Il a cherché la réponse <i>en 10 mn</i>
Il a élaboré la réponse <i>pendant 10 mn</i>	Il a élaboré la réponse <i>en 10 mn</i>
★Il a trouvé la réponse <i>pendant 10 mn</i>	Il a trouvé la réponse <i>en 10 mn</i>

Ainsi, Vendler propose la classification suivante :

- les états : procès homogènes duratifs inhérents à leur nature stable et non transitoire (ne présentant pas de terme naturel).

Exemple :

(5.4) Il connaît la réponse.

- les activités : procès homogènes à chaque instant de leur durée, exprimant un déroulement (ne présentant pas de terme naturel non plus).

Exemple :

(5.5) Il cherche la réponse.

- les accomplissements : procès hétérogènes duratifs, n'existant pas indépendamment de leur fin.

Exemple :

(5.6) Il élabore la réponse.

- les achèvements : procès hétérogènes ponctuels, présentant une complétude qui exclut leur déroulement dans le temps.

Exemple :

(5.7) Il trouve la réponse.

À travers les exemples précédents, il apparaît clairement que l'acceptabilité des phrases testées ne dépend pas seulement de leur forme. Nous retiendrons donc que le mode d'action relève aussi de la sémantique lexicale.

Dowty [1979] énumère divers tests, qui permettent de différencier ces classes les unes des autres (et font même apparaître des sous-classes).

<i>Critère</i>	<i>États</i>	<i>Activités</i>	<i>Accomplissements</i>	<i>Achèvements</i>
Vérifie tests non-statiques (<i>être en train de</i> et mise à l'impératif)	non	oui	oui	difficilement
Φ pendant une heure entraîne Φ vrai à chaque instant	oui	oui	non	n.a.p.
Φ en une heure prend une heure à Φ	mal	mal	OK	OK
X en train de Φ entraîne X a Φ -é	n.a.p.	oui	non	n.a.p.
X Φ -é en une heure entraîne X en train de Φ durant cette heure	n.a.p.	n.a.p.	oui	non
Complément de <i>arrêter de</i>	OK	OK	OK	mal
Complément de <i>finir de</i>	mal	mal	OK	mal
Ambiguïté avec <i>presque</i>	non	non	oui	non
Emploi avec <i>attentivement</i>	mal	OK	OK	mal

OK = l'énoncé est grammatical, normal sémantiquement.

mal = l'énoncé est ni grammatical ni normal sémantiquement.

n.a.p. = le test ne s'applique pas à cette classe de verbes.

FIG. 5.1 - Tests d'Aktionsart d'après Dowty

Quelle classification retenir ? Si l'on ne fait pas la distinction entre propriétés statiques et dynamiques, on peut ranger les états et les activités dans la même classe englobante, celle des procès d'aspect homogène et les accomplissements et achèvements dans une seconde classe englobante, celle des procès d'aspect hétérogène. La différence entre ces

1. Les marques linguistiques de la dynamique temporelle de l'énoncé

deux classes est l'aspect terminatif (ou borné), qui est responsable du caractère hétérogène du procès qui se trouve dénoté.

Concernant la première classe, le test de *être en train de* donne un résultat négatif pour les états alors qu'il est positif pour les activités [Lakoff, 1965]. Concernant la deuxième classe, la distinction accomplissement/achèvement peut prêter à caution. L'argument classiquement avancé consiste à dire que le complément *en X temps* constitue une durée d'activité pour un accomplissement alors qu'il constitue un délai pour un achèvement. En fait la différence entre les deux n'a de sens qu'en fonction du niveau d'abstraction auquel on se place. Dowty (d'après [Kamp, 1992] (p. 102)) remarque ainsi que la distinction entre achèvement et accomplissement ne s'exprime pas en termes directement temporels mais relativement à la possibilité de les décomposer en sous-événements.

Romary ([1989] p. 124) donne une assez bonne image de ce que l'on entend par là, en illustrant la double inclusion temporelle.

« Lorsque parlant de son activité de la veille, une personne affirme "hier, j'ai marché toute la journée", elle exprime un état de fait qui, tel qu'il est décrit dans l'énoncé, peut être représenté par une zone temporelle correspondant grossièrement à une journée. Cependant, la personne n'a peut-être pas effectivement *marché* toute la journée au sens propre du terme. Si on lui demande ce qu'elle a fait, elle pourra dire notamment : "je me suis arrêté pour manger". »

Cette situation peut être schématisée par la figure 5.2 où sont rapportées les portions d'énoncés correspondant à chaque zone temporelle. Une des propriétés de la langue, comme de n'importe quel autre système de représentation est de donner une description de la réalité selon des degrés de complexité très variables. On retrouve ce mécanisme dans le raisonnement à profondeur variable de Kayser [1988].

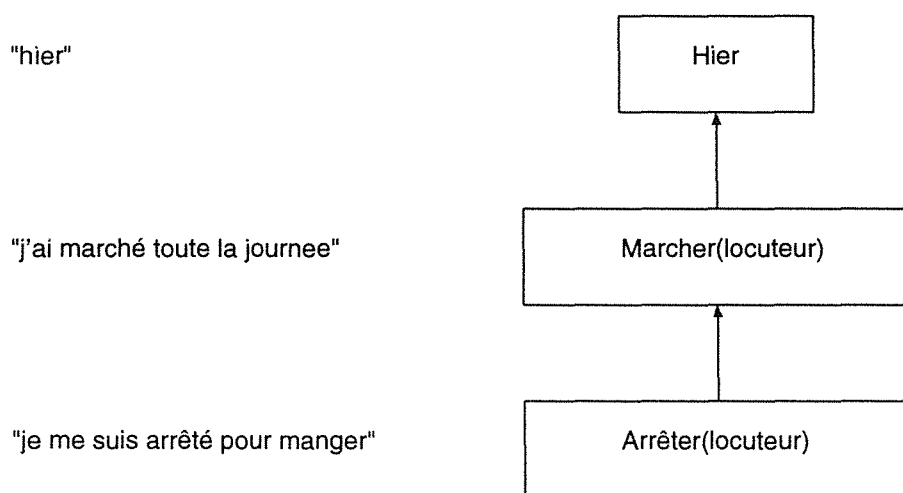


FIG. 5.2 - Représentation d'une promenade

Ainsi, au lieu de parler d'événement ponctuel, on parlera d'événement atomique, dans le sens de non-décomposé. Si activités et accomplissements peuvent se trouver « comprimés » en un point (*blob* en anglais), cela ne signifie pas qu'ils cessent d'avoir une extension temporelle mais que leur structure interne perd son importance.

Dans un dialogue de commande orienté par la tâche, c'est la deuxième classe, ayant un aspect terminatif, qui se voit privilégiée. Terminologiquement, ce qui caractérise d'ailleurs les accomplissements est de se rapporter à la notion de but ou de tâche devant être accomplie. Tout accomplissement en cours appelle une implication résultative. Cependant, puisque nous décomposons le verbe par rapport au déroulement qu'il induit, il nous faut une façon canonique de représenter la propriété résultante.

1.2 Le participe passé comme expression d'un résultat

Étymologiquement, le participe passé tient à la fois (« participe ») du verbe et de l'adjectif [Tesnière, 1969]. Mais à la différence de l'adjectif, le participe passé, comme nous allons le voir, est directement lié à la notion de résultat.

1.2.1 Comparaison entre participe passé et adjectif

À la différence d'un adjectif, un participe passé véhicule la dynamique du verbe dont il est dérivé. Un exemple typique nous aidera à mieux appréhender les différences entre adjectifs et participes passés. À partir de l'adjectif *rouge*, on a formé le verbe *rougir* (intégration de « devenir rouge ») qui a donné le participe passé *rougi*. Ainsi, McCawley propose de décomposer un verbe comme *rougir* de la manière suivante²(figure 5.3) :

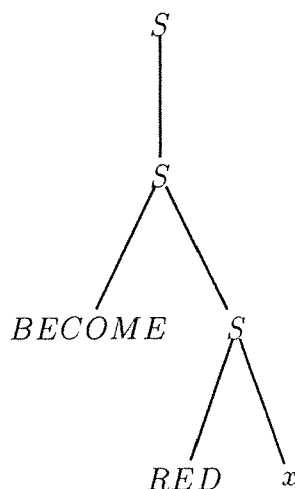


FIG. 5.3 - Structure sémantique de *x reddens (x rougit)* d'après McCawley

Si l'on parle d'un « fil de fer rouge », on exprime pour ce fil de fer une propriété purement statique. On laisse en effet indéfinie l'étendue temporelle durant laquelle on peut dire que le fil de fer est rouge. Par contre, si l'on parle d'un « fil de fer rougi », alors on exprime qu'il est devenu rouge, c'est-à-dire qu'il existe une zone temporelle passée où le fil de fer n'était pas rouge. On pourrait effectuer une analyse tout à fait analogue en comparant par exemple « un chiffon sec » et « une saucisse séchée ». Dans tous ces

2. De la même façon, McCawley analyse le verbe *x kills y* en (CAUSE x (BECOME (NOT (ALIVE y))))).

1. Les marques linguistiques de la dynamique temporelle de l'énoncé

exemples, l'emploi du participe passé implique l'existence d'un événement antérieur³ (un rougissement, un séchage) dont il qualifie le résultat (être rouge, être sec).

Cette approche peut se rencontrer chez certains tenants de la sémantique générative [Lakoff, 1965; McCawley, 1968], pour qui la structure profonde de la figure 5.4 (d'après [Parsons, 1990, p. 107]) est traitée comme une forme à partir de laquelle la syntaxe de l'énoncé *You close the door* doit être dérivée.

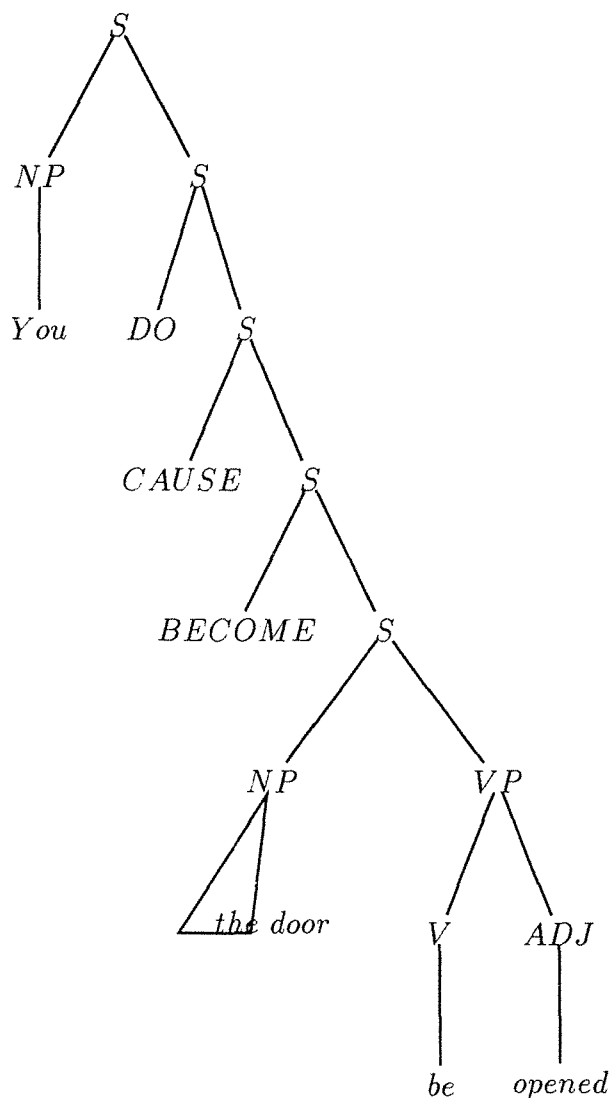


FIG. 5.4 - Structure sémantique de *You close the door* d'après McCawley

Ainsi, plus qu'un état, le participe passé peut exprimer un résultat. L'emploi du participe passé en français apparaissant au passif ou au parfait (ou passé composé), il est intéressant de vérifier notre hypothèse sur le participe passé dans ces deux cas.

3. Il est important de faire la différence entre l'ontologie (ce qui est) et la langue (ce qui est dit) : on ne dénie pas avec un adjectif l'existence d'un état antérieur, mais on n'en parle pas.

1.2.2 Le passif: tournures courtes et longues

On distingue classiquement deux types de tournures passives, les courtes (du type *être + participe passé*) et les longues, auxquelles on ajoute un complément d'agent avec les prépositions *de* ou *par*. Notre étude, visant à décrire un état résultatif, est davantage relative à la tournure passive courte. Par exemple, dans l'énoncé de commande *Affiche une fenêtre*, l'état résultatif qui lui correspond est *la fenêtre est affichée*. C'est donc sur l'objet grammatical de la contrepartie active que s'applique l'implication résultative. Précisons qu'il n'y a pas d'équivalence entre tournures passives et actives, comme le prouve l'existence de phrases actives sans contreparties passives⁴ et, le cas échéant, des considérations vériconditionnelles entre phrases converses⁵. Tout au plus, nous pouvons affirmer que si la situation décrite par une phrase passive est réalisée, alors celle décrite par la phrase correspondant à la contrepartie active l'est également.

Selon les théories classiques, la tournure passive courte est issue de la tournure passive longue avec un effacement de l'agent. On avance alors comme explication le fait que l'on omet l'agent lorsque celui-ci est restituable à partir du contexte. Or, comme le fait remarquer Desclés [1993], cela pose un certain nombre de problèmes. D'abord, l'agent n'est pas toujours restituable. Parfois, on ne veut même pas qu'il soit restitué, comme par exemple dans *le vase a été cassé* (peu importe qui est le coupable). Ensuite, des études menées en linguistique générale nous montrent que (a) il existe un certain nombre de langues (telles que l'arabe ou le turc) [Keenan, 1985] où le passif existe uniquement sous sa tournure courte, et (b) pour les langues admettant les deux tournures du passif, une très grande majorité d'énoncés au passif emploient la tournure courte. D'où l'analyse proposée par Desclés [1993] :

- le passif long est une extension du passif court ;
- le complément d'agent est à mettre au même rang qu'un complément adverbial.

Nous retiendrons en conséquence que la forme verbale passive introduit un changement (a) subi par le référent de son sujet (b) causé par un agent qu'elle implice (même s'il n'est pas reconstituable d'après le contexte).

4. Par exemple,

(5.8)

- (a) Fabrice a pris le TGV à Biarritz.
- (b) ? Le TGV a été pris par Fabrice à Biarritz.

5. Notamment, l'ordre suivant lequel apparaissent des quantificateurs est essentiel, comme illustré par les deux phrases converses ci-dessous :

(5.9)

- (a) Beaucoup de gens lisent peu de livres.
« *Les gens qui lisent peu de livres sont beaucoup.* »
- (b) Peu de livres sont lus par beaucoup de gens.
« *Les livres que beaucoup de gens lisent sont peu.* »

1.2.3 L'acquis du parfait

Le parfait (ou passé composé) est employé pour poser un procès comme accompli mais se rattachant au présent [Benveniste, 1974c, ch. 13]. Sa construction utilise dans la très grande majorité des cas l'auxiliaire *avoir*, et pour les cas restants l'auxiliaire *être* (*sortir, naître, tomber, ...*). Dans le cas du parfait avec *avoir*, Benveniste [1974a, p. 129] parle d'acquis d'opération, en dénotant le sujet comme possesseur d'un résultat qui lui est acquis. Concernant le parfait avec *être*, Benveniste parle d'acquis de situation :

« Par acquis de situation, nous entendons qu'une certaine situation est établie, en conséquence de l'effectuation d'un mouvement qui a nécessairement cette situation pour résultat. » (cf. [Benveniste, 1974c, p.182] cité par [Borillo, 1991, p. 14])

En récapitulatif, que ce soit sous la forme passive ou avec le passé composé, le participe passé reflète que :

- quelque chose a été acquis, mais aussi que
- quelque chose a causé ce changement.

Il devient dès lors concevable d'associer un événement à chacun de ces points.

1.2.4 Caractérisation de l'état résultant en termes événementiels

Dans la lignée de Davidson [1967] (cf. §2.3.2 p. 73) et de Dowty [1979], Parsons [1990] propose un traitement de certains phénomènes liés à l'aspect en termes d'événements. S'inspirant de Dowty, il reprend les relations BECOME et CAUSE, mais avec la particularité de les faire porter sur des événements et des états. Parsons distingue les états des événements⁶ par le fait que seuls les événements culminent (*Culm*), leur point commun avec les états étant d'être vérifiés (*Hold*). L'idée de Parsons est alors de décomposer les procès en faisant apparaître des événements ou des états internes :

État

"the door is closed" = $(\exists s)[\text{BEING-CLOSED}(s) \ \& \ \text{Theme}^7(s, \text{door}) \ \& \ \text{Hold}(s)]$

Évolution

"the door closes" = $(\exists e)[\text{Culm}(e) \ \& \ \text{Theme}(e, \text{door}) \ \& \ (\exists s)[\text{BEING-CLOSED}(s) \ \& \ \text{Theme}(s, \text{door}) \ \& \ \text{Hold}(s) \ \& \ \text{BECOME}(e, s)^8]]$

Causalité

"x closes the door" = $(\exists e)[\text{Culm}(e) \ \& \ \text{Agent}(e, x) \ \& \ (\exists \acute{e})[\text{Culm}(\acute{e}) \ \& \ \text{Theme}(\acute{e}, \text{door}) \ \& \ \text{CAUSE}(e, \acute{e}) \ \& \ (\exists s)[\text{BEING-CLOSED}(s) \ \& \ \text{Theme}(s, \text{door}) \ \& \ \text{Hold}(s) \ \& \ \text{BECOME}(\acute{e}, s)]]]$

6. Parsons regroupe les accomplissements et les accomplissements sous l'appellation commune d'événements.

7. *Theme* (équivalent à notre *Patient*), de même que *Agent*, sont des rôles sémantiques spécifiques à une classe de verbes.

8. Un fait provenant de la relation BECOME est que le thème de l'événement est le même que celui de l'état : $\text{BECOME}(e, s) \rightarrow [\text{Theme}(e, x) \equiv \text{Theme}(s, x)]$.

Dans cette dernière formule, on remarquera que, pour les verbes de causalité, il existe un ordre « causal » selon lequel interviennent les rôles sémantiques (*Agent* précède *Theme*). Pour cette classe de verbes, la présence d'un complément d'objet direct à la suite du verbe va contribuer à caractériser l'état cible et donc l'effet du changement. En tous les cas, le verbe dont est issu le participe passé joue à ce niveau un rôle de premier ordre. À cette fin, nous allons montrer qu'une conception sémantique de la transitivité liée au verbe va directement dans ce sens.

1.3 Conception sémantique de la transitivité

On dit classiquement qu'un verbe est transitif lorsqu'il est possible de lui adjoindre un objet sur lequel il s'applique. De ce point de vue, la transitivité est une notion grammaticale : sont considérés comme transitifs les verbes qui admettent un complément d'objet direct (transitifs directs) ou indirect (transitifs indirects) [Dubois, 1967]. Une telle définition, s'appuyant sur des critères distributionnels, devrait permettre de distinguer, à travers leurs emplois, les verbes qui sont ou non séparables de leurs compléments. Or, même pour des compléments sensés être inséparables, il est presque toujours possible de trouver un contexte où l'objet en question soit suffisamment saillant pour pouvoir s'en passer. Par exemple, le complément d'objet est obligatoire pour le verbe *attraper*. Pourtant, lorsque deux enfants jouent à la balle, l'un pourra dire à l'autre (en même temps qu'il lui envoie la balle) :

(5.10) Attrape !

Les critères distributionnels reposant sur une incomplétude se révélant insuffisants, on en vient à préférer une définition plus sémantique de la transitivité. Partant de l'étymologie même du mot, on peut analyser la transitivité comme un « mouvement porté par le verbe », que le verbe transitif transmet du sujet à l'objet [Tesnière, 1969, chapitre 100, p. 242] [Pottier, 1962]. On dit alors qu'un verbe est intransitif lorsqu'il ne peut exprimer rien d'autre qu'un comportement relatif au sujet. Autrement dit, la transitivité est l'indice grammatical d'une influence extérieure sur un objet qui vient se surajouter à l'image comportementale du sujet [Langacker, 1991]. Elle se manifeste par la présence d'un groupe nominal situé après le verbe ou d'un élément pronominal. Un cas intéressant est lorsqu'on peut faire abstraction de l'influence extérieure pour ne retenir que l'image comportementale liée à la personne. On emploie alors de façon intransitive un verbe transitif. Par exemple, dans la forme *manger quelque chose*, on peut faire abstraction de l'objet pour ne retenir que *manger* dans

(5.11) Il mange bien.

Comme nous allons le voir, c'est au niveau de la présence ou de l'absence d'objet que s'illustrent les notions d'aspect et d'Aktionsart précédemment décrites. Ainsi, dans l'énoncé

(5.12) Il mange son sandwich.

le fait que le sandwich représente une quantité bien délimitée va imposer un terme au niveau du déroulement du procès. On a alors affaire à un *accomplissement* dont le résultat

est *son sandwich mangé*. En revanche, l'objet étant absent dans 5.11, la continuabilité du procès qu'implique le verbe ne se heurte plus à aucune borne, et ce dernier peut se perpétuer indéfiniment dans le temps. On décrit alors une action considérée comme un tout homogène, c'est-à-dire une *activité* au sens des Aktionsarts. Il en est de même lorsque le complément d'objet réfère à une quantité indéfinie :

(5.13) Il mange des sandwiches.

C'est en partant de cette constatation que Verkuyl [1972] a défini sa théorie compositionnelle de l'aspect, reposant sur des traits associés aux syntagmes de l'énoncé, comme l'illustrent (de manière simplifiée) les figures 5.5 et 5.6.

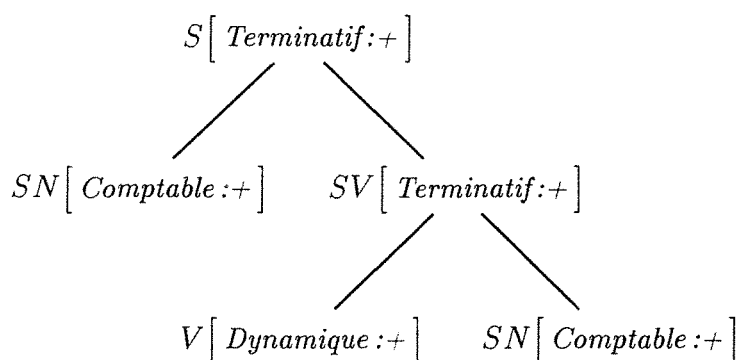


FIG. 5.5 - Compositionnalité de l'aspect chez Verkuyl (1)

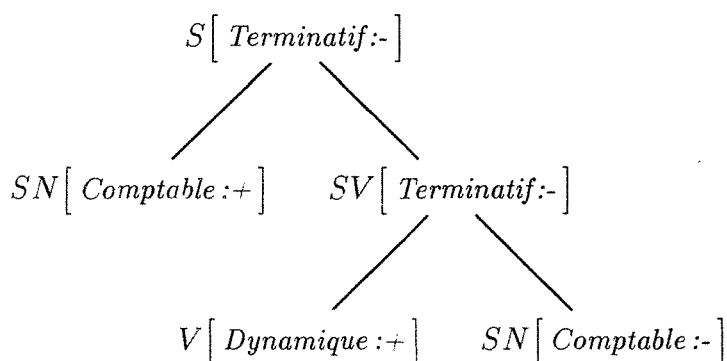


FIG. 5.6 - Compositionnalité de l'aspect chez Verkuyl (2)

Selon Verkuyl, le verbe V et son complément d'objet SN constituent donc une unité aspectuelle séparée (interne) qui doit être distinguée de l'unité aspectuelle formée au-dessus (externe) par le prédicat SV joint à son sujet SN. L'idée principale de Verkuyl est que l'aspect (qu'il considère comme opposition borné/non borné) est obtenu par composition liée à des contraintes syntaxiques sur l'énoncé. Dowty [1979, p. 63-65] a pu reprocher à cette approche d'avoir déplacé vers le plan de la syntaxe ce qui relève de la sémantique :

les propriétés liées aux Aktionsart (et des prépositions permettant de les classer) ainsi que les propriétés spécifiant la référence aux objets.

Un reproche général que l'on peut effectuer à l'approche de Verkuyl est de ne pas s'appuyer sur les référents qui peuvent être impliqués dans le contexte discursif. Pour expliquer alors le lien qu'il peut y avoir entre transitivité et aspect, Hopper et Thompson proposent une théorie de la transitivité s'appuyant sur le discours, les caractéristiques grammaticales et sémantiques en dérivant [Hopper et Thompson, 1980]. Ils ont ainsi remarqué qu'il existait diverses propriétés au niveau de l'énoncé (marquées morphologiquement ou sémantiquement) qui co-variaient très étroitement, et ce de manière spécifique à chaque langue étudiée. Selon eux, c'est la transitivité, transférant une activité, qui serait au cœur de cette co-variance. Parmi les propriétés impliquées, citons le nombre de participants, le caractère dynamique du procès, l'aspect (terminatif/non-terminatif), l'intentionnalité, l'agentivité du sujet, le degré selon lequel est affecté ou individualisé l'objet. En guise d'illustration, supposons une langue pour laquelle il existe une marque morphologique permettant de distinguer les verbes téliques (visant une situation finale) de ceux qui ne le sont pas (dits alors atéliques)⁹. Si alors on constate qu'en présence d'un verbe télique, on a une individualisation de l'objet, on pourra en déduire que nécessairement, en présence d'un verbe atélique, on n'aura pas d'individualisation de l'objet.

Dans cette section, nous avons établi que les notions d'aspect et de transitivité étaient fondamentales pour aboutir à une analyse d'un énoncé en termes de ruptures de propriétés sur des objets. Or, dans les théories linguistiques modernes, le calcul de l'aspect présuppose que l'on connaisse déjà le résultat de l'interprétation (dans l'approche de Verkuyl, reposant sur la grammaire de Montague, il s'agit de la structure de l'énoncé). Les théories de cette famille partent en effet d'une conception purement formelle de la transitivité, simplement fondée sur la nécessité de la présence d'une forme particulière après le verbe. Nous pensons au contraire que des données sémantiques liées aux constituants de l'énoncé agissent dans le processus de construction de la structure de l'énoncé. Notamment, nous allons émettre dans la section suivante l'hypothèse que l'aspect participe à cette construction.

2 Les structures sous-jacentes de l'énoncé

Nous venons de montrer que le contexte avait une grande importance au niveau de la variabilité des propriétés sémantiques relatives au verbe. Mais afin de pouvoir amorcer le processus, il est nécessaire d'isoler préalablement les propriétés qui se trouvent concernées. Autrement dit, il faut mettre au jour ce que le verbe renferme en lui dans sa forme la plus abstraite et donc en dehors de son emploi en discours, ce que s'est efforcé de faire Guillaume [1970] dans les années 1945. Il a notamment montré que l'aspect intervenait avant que le verbe ne soit formé et, de ce fait, était fondamental pour rendre compte de la structure de l'énoncé.

9. C'est typiquement le cas du russe.

2.1 La dynamique de l'aspect verbal selon Guillaume

Le verbe (en français) renferme une potentialité de développement temporel. C'est ce que Guillaume appelle la *tension*. Il lui oppose la *détension*, correspondant à ce qui a déjà été réalisé, à de la tension qui a été dépensée. Un verbe à l'infinitif ne renferme que de la tension, n'offrant qu'une possibilité d'action (pouvant résulter d'une volonté ou d'une intention). Par contre, un verbe au participe présent contient à la fois une tension et une détension, ce qui fait qu'il exprime un déroulement. Quant au participe passé, il ne renferme plus aucune tension, car elle a été entièrement dépensée : le temps relatif au développement verbal s'est entièrement écoulé, a épuisé tout ce qui lui restait de potentiel pour finalement s'achever.

Guillaume définit l'aspect sous la forme d'une part variable de continuable et d'accompli [1970], en l'appliquant avant tout aux formes pré-verbales, tenant aussi du nom (tel que l'infinitif) ou de l'adjectif (participes). L'aspect selon Guillaume est ainsi constitué d'un couple (*tension, détension*) ou, en des termes moins abstraits, un couple (*potentiel, accompli*) comme l'illustre la figure 5.7 ci-dessous. Le déroulement du procès se traduit

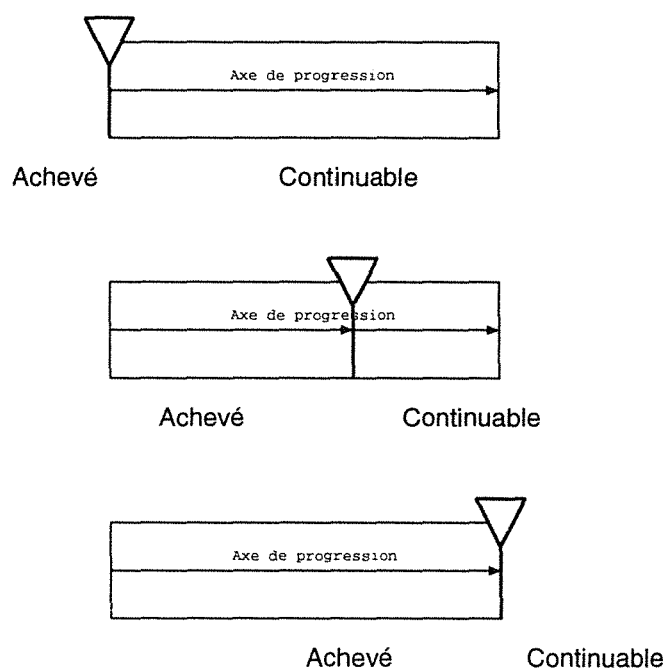


FIG. 5.7 - L'aspect verbal selon Guillaume

donc par une variation de l'aspect, où la part de potentiel s'estompe au profit de la part de l'accompli.

Au delà de la borne correspondant à une tension nulle (i.e. plus de potentiel pour continuer), on ne met plus en évidence le déroulement même de l'image verbale, mais le déroulement d'une séquelle de cette image. Par exemple, la forme infinitive *imprimer un fichier* image le fichier non encore imprimé mais qui se destine à l'être. À l'opposé, la séquelle qui correspond, *avoir imprimé un fichier*, image le fichier déjà imprimé. Plus précisément, le participe passé figurant dans *un fichier imprimé* ne présente plus de tension

et, de ce fait, n'est plus considéré comme un verbe. Pour lui redonner ce statut, il faut lui adjoindre un verbe auxiliaire, ce qui donne une forme composée. Cet auxiliaire a donc pour fonction de renouveler la tension du verbe au moment où elle expire et de la prolonger ainsi au delà d'elle-même. C'est notamment sur ce point là que se distingue la notion d'aspect selon Guillaume par rapport à des conceptions de l'aspect liées à l'opposition *perfectif/imperfectif* induite par les temps grammaticaux composés [Dubois, 1967]. Selon Guillaume, l'aspect se situe sur trois plans :

1. le plan tensif (aspect simple relatif au déroulement même de l'image verbale), par exemple *imprimer, imprimant, imprimé*;
2. le plan extensif (aspect composé relatif au déroulement d'une séquelle de cette image¹⁰), par exemple *avoir imprimé, ayant imprimé, eu imprimé*;
3. et beaucoup plus rare le plan bi-extensif, de la forme surcomposée *avoir eu/été + participe passé* et *ayant eu/été + participe passé*.¹¹

Nous avons à présent tous les éléments pour examiner comment l'aspect intervient au niveau de la structuration de l'énoncé.

2.2 Le verbe comme élément structurant de l'énoncé

La dualité temporel/non-temporel telle que nous l'avons décrite ci-dessus rejoint la complémentarité que mentionne [Vendler, 1967] entre deux conceptions philosophiques duales du monde. La première envisage le monde comme l'idée qui limite la totalité de tous les objets [Vendler, 1967]. Il s'agit d'une vision statique du monde, que l'on pourrait qualifier de *descriptive*. Mais comme le remarque Vendler, il existe un autre côté de la médaille, où le monde est considéré comme un procès en extension selon un axe temporel. Par complémentarité avec le point de vue évoqué ci-dessus, il s'agit là d'une vision dynamique du monde, que l'on pourrait qualifier de *constructive*. Cette vision du monde trouve ses indices dans l'énoncé, notamment dans le verbe.

2.2.1 Le verbe comme structure d'accueil

Le verbe est typiquement considéré comme l'organisateur de la proposition. Mais nous entendons ici marquer une différence majeure avec d'autres théories linguistiques qui placent le verbe au sommet d'une hiérarchie, comme par exemple la conception actancielle de Tesnière. On retrouve un phénomène analogue avec l'approche qu'en donne Mel'cuk [1987]. Dans tous ces cas, le verbe est conçu comme une structure d'accueil et on retrouve dans ces modèles une certaine hiérarchie des constituants. Mais le problème est que cette hiérarchie n'est pas motivée en terme d'autonomie. De plus, le fait que les différents actants sont logés à la même enseigne efface l'aspect évolutif que véhicule le verbe.

10. Parfois réduite à un point d'observation avec des verbes n'impliquant pas d'effet prolongé. Par exemple, pour le verbe *marcher*, il s'agirait de la rupture correspondant à *s'arrêter de marcher*.

11. Cette forme permet d'exprimer le passé sans quitter du regard le présent. Par exemple, dans *Dès qu'il a eu déjeuné, il s'en est allé*, le verbe *a eu déjeuné* exprime le résultat, vu depuis le présent, de l'action passée *déjeuner*.

2. Les structures sous-jacentes de l'énoncé

L'approche que nous prônons est que le verbe est effectivement une structure particulière d'accueil au niveau de la proposition. Mais à la différence des approches précédentes, nous défendons les points suivants :

- Il existe un ordre d'accueil des constituants ;
- Cet accueil s'effectue en imposant des contraintes temporelles entre les différents constituants.

Nous pouvons illustrer cela en prenant l'exemple d'une conception de la phrase japonaise telle que la présente [Tamba, 1991] :

« Manifestement Yamada¹² assimile le mot *principal* à un *hôte*, un *maître de céans* qui habite ce *chez-lui* qu'est l'événement, le procès, où il est amené à recevoir un *mot secondaire*, identifié à un *invité*, un *visiteur*. »

Une telle représentation de type socio-topologique de la structure phrastique japonaise nous éloigne des images qu'évoquent les notions tant grammaticales qu'ontologiques de *sujet* et d'*objet* dans la pensée occidentale.

Notre première hypothèse est que le verbe introduit un schème d'évolution (une demeure) et une structure d'accueil ordonnée. Si cette conception est peu répandue dans la morphogenèse de la phrase française, elle est beaucoup plus manifeste dans la morphogenèse de la phrase japonaise. En effet, le japonais est un exemple de langue où les fonctions sont marquées par des traits morphologiques agglutinants. Le verbe japonais assure ainsi deux fonctions par rapport au procès :

- Fonction référentielle :

Elle désigne un événement en le nommant, présenté comme un tout factuel qui entre dans le champ de connaissance de quelqu'un. À la différence du français, verbe et verbe de propriété (adjectif qualificatif) sont rassemblés.

- Fonction de contrôle :

Elle contrôle l'assemblage des différents éléments constitutifs de l'événement. Ainsi, la particule « ga » postposée à un GN lui donne une valeur de déterminant intrinsèque du procès du début à la fin. La particule « (w)o » ne relie le GN qu'à la phase finale du procès. Elle induit le nouvel état de choses résultant de l'accomplissement factuel.

Les particules en question ont d'emblée un caractère sémantique : elles servent à caractériser une entité.

Nous avons déjà vu qu'un verbe n'était pas autonome, mais qu'il avait besoin de s'appliquer sur quelque chose. Nous allons montrer que dans la structure d'accueil qu'introduit le verbe, il existe une relation particulièrement privilégiée, celle que le verbe entretient avec le sujet.

12. Grammaire japonaise du début du XX^{ème} siècle.

2.2.2 L'élément de fonction sujet

Le verbe est intimement lié à l'expression d'un comportement, en véhiculant un développement dans le temps. Mais, exprimer un seul comportement n'est pas suffisant. Une forme infinitive comme "manger" ne saurait constituer un énoncé à elle seule, en dehors de l'expression d'une personne grammaticale. Or, la fonction du sujet est précisément d'exprimer cette personne grammaticale. Notons qu'il s'agit là d'une fonction purement grammaticale¹³. Cette approche du verbe rejoint celle de Guillaume.

Le passage au verbe Tant que l'on se situe au niveau aspectuel de l'infinitif, le participe présent ou le participe passé, on ne peut pas à proprement parler de la classe grammaticale « verbe ». En premier lieu, le verbe est lié à la personne : la personnalisation du verbe intervient dès que l'on quitte le domaine aspectuel (la forme infinitive soit simple, soit composée) du verbe. Il s'agit là de la principale différence avec le nom. Un nom peut véhiculer un développement (ou déroulement) temporel que l'on considère dans son ensemble (par exemple, *arrivée* ou *séparation*). L'infinitif lui-même a un statut nominal. Par contre, on ne peut pas adjoindre de personne à un nom : il existe des sujets du verbe mais pas des sujets du nom.

Ainsi, la personne est liée au verbe pendant tout son déroulement temporel. De manière générale, elle se manifeste de deux manières (éventuellement en même temps) :

1. sous la forme d'un suffixe lié à la forme verbale ;
2. sous la forme d'un groupe autonome ayant la fonction de sujet.

Dans le cas particulier de l'impératif, la première manière est la seule possible.

Bilan : la fonction de sujet Il apparaît donc que le lien entre le sujet et le verbe est primaire. Cela a été la conclusion à laquelle était parvenue Langacker, repris par [Kleiber, 1993] (p. 119) :

« On s'aperçoit qu'à l'iconicité d'isomorphisme s'ajoute une iconicité de motivation¹⁴. Ces différentes propriétés, qui varient, certes, plus ou moins selon les langues, tendent en effet vers un dénominateur formel commun : la primauté et l'autonomie ou délimitation (relative) du sujet par rapport aux autres constituants. Elles se révèlent iconiques en ce qu'elles traduisent directement la saillance qui constitue le contenu conceptuel du sujet. Elles miment en quelque sorte sa figuralité. Cela apparaît clairement à l'intérieur des langues où le sujet occupe de façon standard la première place : cette propriété « linéaire » qui lui est reconnue marque iconiquement son statut sémantique de figure dans la proposition, c'est-à-dire d'élément qui ressort, qui émerge par rapport au reste de la proposition. »

13. Nous ne considérons pas le rôle que l'on peut prêter au sujet de désigner celui qui fait l'action.

14. Il s'agit d'une distinction introduite en 1932 par Peirce [1978], reprise par Haiman [1980]. L'*iconicité d'isomorphisme* correspond, comme son nom l'indique, à la relation *une forme - un sens*. L'*iconicité de motivation* correspond quant à elle à une iconicité d'image (ressemblance entre la forme du signe et le référent qu'il exprime) transposée sur le plan grammatical.

2.2.3 L'élément de fonction complément d'objet

Le verbe peut véhiculer plus qu'un simple comportement. Il peut exprimer une zone d'influence avec un objet autre que la personne dont on implique le comportement, typiquement introduite par la fonction *complément d'objet*. Comme nous l'avons vu, on rend compte de cette possibilité en utilisant la notion de transitivité.

Une approche guillaumienne de la transitivité. La transitivité peut être généralement conçue comme une propriété de l'énoncé indiquant qu'une activité est transférée d'un participant vers un autre [Hopper et Thompson, 1980]. Pottier précise cette notion d'influence extérieure qui constitue la destination du mouvement porté par le verbe dans sa *Linguistique Générale* [Pottier, 1985] :

« La transitivité est une propriété sémantique qui consiste à orienter une substance (dénotée par un sujet) vers un au-delà (dénoté par un objet), en le suggérant.(cf. §231) »

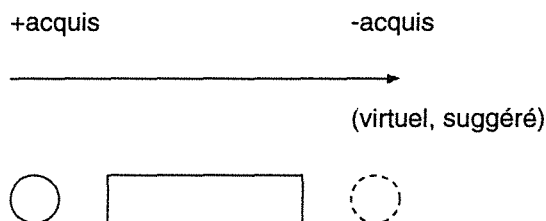


FIG. 5.8 - *La transitivité selon Pottier*

Dans la continuité de Guillaume, on retrouve une approche très similaire chez [Moignet, 1973], qui distingue dans le verbe, d'une part une partie active ou élément causatif, incidente au sujet ; et d'autre part une partie passive ou élément résultatif, qui, dans le cas des verbes transitifs, est incidente à l'objet¹⁵.

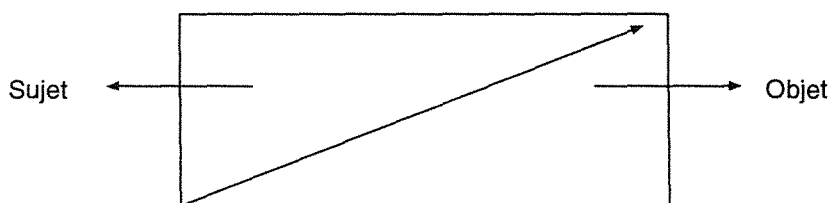


FIG. 5.9 - *La sémantèse du verbe transitif selon Moignet*

La flèche en diagonale montre que l'opération signifiée par le verbe est un passage progressif d'un stade de tension intégrale à un stade de détension pure. L'opération prend son départ à un support nominal externe (le sujet) auquel elle applique son élément causatif et aboutit à un support nominal externe (l'objet) auquel elle applique son élément

15. L'objet est l'au-delà vers lequel s'oriente la sémantèse verbale (i.e. le processus de construction du verbe).

résultatif. La condition pour que cette « application » soit possible est qu'il existe une affinité sémantique entre les éléments mis en relation. Remarquons que cette affinité n'est pas nécessairement donnée mais peut constituer un but pour une chaîne d'inférences.

La relation verbe - complément d'objet. Nous avons vu que le verbe entretenait avec son sujet une relation particulière, que nous avons qualifiée de primaire. Et elle est bien première sur le plan du déroulement temporel. C'est, en effet, seulement ensuite que l'on s'intéresse, le cas échéant, à l'influence potentielle exercée sur d'autres objets que la personne. On retrouve cette dynamique sous forme iconique dans des langues telles que le français [Kleiber, 1993], où l'on place l'objet après avoir formé le verbe. Avec un verbe transitif sans objet direct, le déroulement reste indéfini, aucun terme n'étant marqué. On retrouve l'interprétation de Jakobson sur la relation prédicative entre le sujet comme point de départ et l'objet comme point d'arrivée et « aboutissement naturel du procès » [Pottier, 1962, p. 270]. De ce fait, le lien entre l'objet et le verbe est secondaire. On a donc établi une hiérarchie de liens avec le verbe entre le sujet verbal et l'objet verbal.

Parmi toutes les propriétés partagées par les verbes transitifs, on trouve celle d'employer dans leur forme composée l'auxiliaire *avoir* uniquement. Par contre, pour les verbes intransitifs, il y a deux sous-ensembles (qui ne sont cependant pas disjoints) : les verbes qui peuvent employer l'auxiliaire *avoir* (*marcher, dormir, nager, ...*) d'une part, et ceux qui peuvent employer l'auxiliaire *être* (*sortir, naître, tomber, ...*) d'autre part. La question que nous nous posons donc est d'isoler le facteur responsable de l'emploi de tel ou tel auxiliaire.

Concernant les verbes intransitifs utilisant l'auxiliaire *être*, aussi bien Benveniste que Guillaume ont pu remarquer que leur caractéristique commune se situe dans l'impossibilité d'une reprise immédiate de l'action, une fois qu'elle a atteint son terme. Ainsi, une fois qu'on est sorti, c'est chose faite : on ne peut pas sortir encore. Pour ce faire, il faudrait rentrer auparavant, ce qui signifie que la reprise de l'action de sortir est médiante. Dans la terminologie de Guillaume, on l'exprimerait en considérant qu'un tel verbe enferme intérieurement une limite de tension. L'achèvement du verbe n'ouvre aucune perspective de continuité. Par contre, pour les verbes intransitifs utilisant l'auxiliaire *avoir*, il n'existe pas de pareille limite. Si l'on s'est arrêté de marcher, il doit être possible de se remettre immédiatement à marcher. Le verbe peut continuer, n'enfermant aucune limite de tension.

Nous pouvons donc expliquer une partie de l'opposition entre l'emploi des auxiliaires *être* et *avoir* dans la forme composée. Nous nous plaçons au stade du développement de l'action où l'on considère son terme. On distingue alors deux cas possibles, qui se sont petit à petit inscrits dans la diachronie de la langue française [Benveniste, 1974c] :

- le verbe ne peut plus retrouver le potentiel de développement (= tension) qu'il vient de dépenser entièrement. Il se retrouve bloqué contre une limite de continuité véhiculée par l'évocation-même du verbe. Après que l'on soit né, il n'est pas possible de continuer cette naissance en naissant de nouveau (idem pour *sortir, tomber, ...*). Privé de toute possibilité de continuation de déroulement, la seule possibilité de continuation qui s'offre est celle d'être le résultat du déroulement. De ce fait, le verbe entrera passivement dans sa phase d'extension (emploi auxiliaire + participe passé), où l'on considère ce qui reste du développement temporel de l'image verbale

(=action). C'est alors l'auxiliaire *être* qui sera utilisé.

- après avoir dépensé son potentiel de développement (= tension), le verbe peut le retrouver sans la nécessité de quelque intermédiaire que ce soit. Une fois que l'on a marché, il est possible de continuer cette marche en marchant de nouveau. Comme la perspective de poursuite de la continuation reste ouverte, le verbe entrera activement dans sa phase d'extension. C'est alors l'auxiliaire *avoir* qui sera utilisé.

Bilan : la fonction de complément d'objet direct. Le complément d'objet plonge le verbe dans la réalité en spécifiant le terme du déroulement. Cela s'effectue de façon variable selon le degré de définition de l'objet. L'emploi de l'objet s'inscrit donc dans la nécessité de définir davantage le comportement véhiculé par le verbe une fois qu'il a été actualisé. Selon [Kamp, 1992] (p. 110), l'objet direct n'est pas lui-même le point culminant mais il s'agit plutôt d'une indication qu'il y a un point culminant à associer au procès. L'objet fournit dans ces conditions un contexte supplémentaire pour aider à déterminer ce point. En conséquence, les objets directs *non modifiés* posent le problème de ne pas aider l'auditeur à déterminer le point de culminance (on peut citer *habiter une maison* ou *chercher une solution*). Plus un objet et la modification qu'il subit sont identifiables (la détermination jouant ainsi un rôle majeur), plus il est facile de déterminer un état final. Ainsi l'on peut dire que la prédication est indécomposable quand on ne peut pas isoler (linguistiquement parlant) un élément caractérisant un état final.

2.2.4 Les éléments à caractère adverbial

Les éléments à caractère adverbial se répartissent en deux classes, non disjointes, comme nous l'avons vu : les modificateurs de prédicats et les supports d'actants. Nous allons ici nous intéresser aux théories linguistiques qui permettraient de nous indiquer où l'on peut rattacher ces éléments à caractère adverbial : par rapport à l'état initial, par rapport à l'état final, ou encore par rapport au déroulement sous-jacent.

Le cadre d'actance selon Pottier utilise l'axe d'évolution que nous avons introduit pour illustrer la transitivité (cf. figures 5.8 et 5.10).

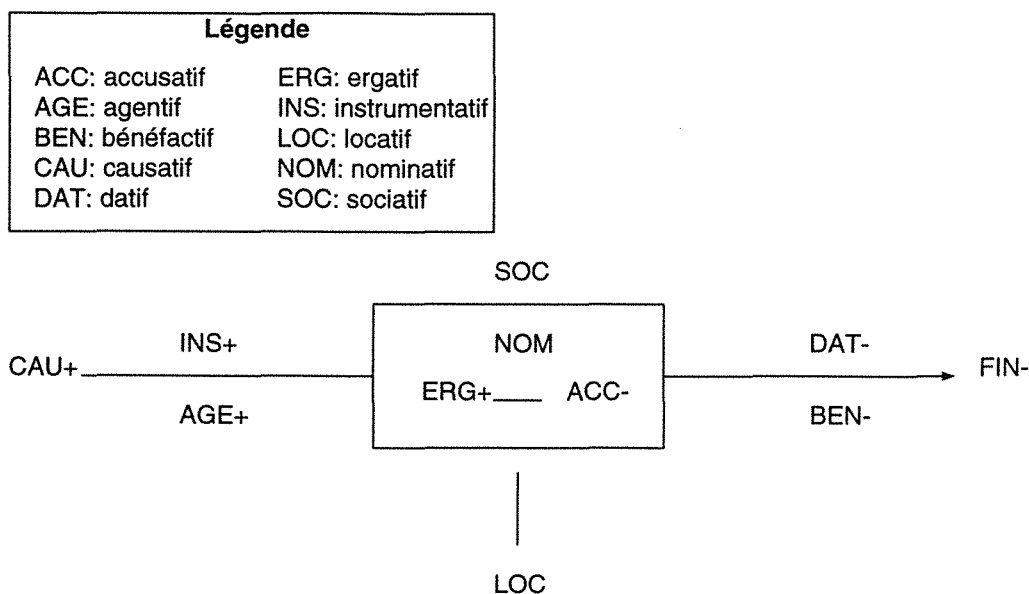


FIG. 5.10 - Le cadre conceptuel selon Pottier

Ces distinctions en zones d'actance se retrouvent au niveau de la classification des prépositions (cf. figure 5.11 d'après [Pottier, 1985] (p. 132)). Dans la figure 5.11, les

		POSITION		
		INITIALE	MEDIANE	FINALE
A C T A N C E		<i>par</i> (grâce à, à cause de, de, avec, sans, si)		<i>à, pour</i> (afin de, dans le but de)
		CAU, AGE, INS.		DAT, BEN, FIN
D E P E N D A N C E	LOC(E)	<i>de</i>	<i>à</i> <i>en, dans</i> <i>sous/sur</i>	<i>à</i>
	LOC(T)	<i>depuis, dès</i>	<i>à, vers</i> <i>pendant</i> <i>avant/après</i>	<i>jusqu'à</i>
	LOC(N)	<i>par, de</i>	<i>de, à</i> <i>-ment</i> <i>avec, sans</i>	<i>à, pour</i>

FIG. 5.11 - Classement aspectuel des prépositions selon Pottier

dépendances se manifestent sous la forme de localisations. Pottier introduit à cet effet

les primitives de domaine : espace (E), temps (T) et notionnel (N). Il peut y avoir des indices linguistiques permettant de privilégier tel ou tel domaine. Par exemple, *pour* et *par* concernent respectivement la causalité et la finalité. Pour autant, nous allons voir qu'il y a des contraintes de liaison indépendantes des domaines (E,T,N).

(5.14)

(a) Il a posé une question sur l'aérodrome de Nancy¹⁶.

(b) L'avion s'est posé sur l'aérodrome de Nancy.

Dans le cas (a), *sur l'aérodrome de Nancy* est un locatif notionnel, alors que c'est un locatif spatial dans le cas (b). Mais il a fallu effectuer un certain nombre d'inférences d'ordre sémantique et pragmatique pour aboutir à cette conclusion. En fait, ce que véhicule strictement *sur* sert à localiser une cible par rapport à un site à l'intérieur d'un domaine sémantique, quel qu'il soit [Vandeloise, 1986]. À un niveau pré-sémantique, ce qui nous intéresse est la caractérisation par rapport à un stade du déroulement.

Des interprétations référentielles avec traits combinatoires. Laur [1993] avait bien remarqué que la nature de la préposition pouvait influencer sur la portée du groupe prépositionnel correspondant, soit au niveau de la situation initiale, soit au niveau de la situation finale. Le problème de départ qu'elle se pose concerne l'interprétation d'énoncés tels que :

(5.15) *Jean est parti de sa chambre* (après le déplacement, Jean n'est pas dans sa chambre).

(5.16) *Jean est parti dans sa chambre* (après le déplacement, Jean est dans sa chambre).

Partant de ce constat, Laur propose une classification des verbes et des prépositions de mouvement à base de traits. Elle introduit à cet effet divers critères pour distinguer différentes classes de verbes ou de prépositions :

- La polarité : *initial*, *médian* ou *final*.
- La localisation : *interne* ou *externe*

La polarité est ici une notion purement aspectuelle. Le lien que l'on peut faire par exemple avec la terminologie de [Pottier, 1987] va de soi :

Initial Inchoatif (apparition d'une propriété) : $\bar{\exists} \rightarrow \exists$

Median Continuatif (perpétuation d'une propriété) : $\exists \rightarrow \exists$

Final Terminatif (disparition d'une propriété) : $\exists \rightarrow \bar{\exists}$

Il serait trop beau que les traits des prépositions et des verbes puissent toujours coïncider, comme le fait remarquer Laur. Lorsqu'il y a coïncidence (dans sa terminologie

16. En tant que thème de la question.

		initial (i)	final (f)	median (m)
changement de lieu de référence (1)	interne (int)	partir se décoller sortir (i,1,int) ○ → AV AP	arriver se coller entrer (f,1,int) → ○ AV AP	passer (par) couper(par) (m,1,int) → ○ → AV PD AP
	interne (int)			courir se déplacer (m,2,int) ⊖ PD
orientation (2)	externe (ext)	s'éloigner reculer (i,2,ext) ○ → AV AP	se rapprocher avancer (f,2,ext) → ○ AV AP	graviter (m,2,ext) ○ ↻ PD

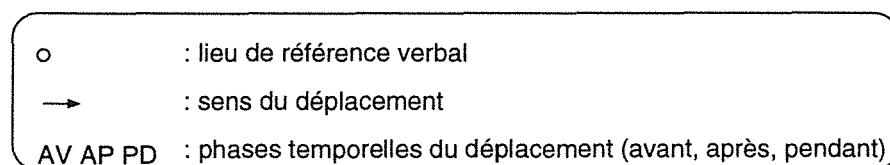


FIG. 5.12 - Classification des verbes de mouvement d'après Laur

congruence), c'est le verbe qui porte l'ensemble des traits relatifs à la description du déplacement. Lorsqu'il n'y a pas coïncidence, les traits véhiculés par la préposition enrichissent la description du déplacement. Ainsi, nombreux sont les verbes qui acceptent des prépositions de type différent des leurs :

Au niveau des polarités, Asher et Sablayrolles [1994] utilisent les règles suivantes :

- 1 Si les polarités des verbes et prépositions sont différentes, le SP est un adjectif.
- 2 Si les polarités des verbes et prépositions sont identiques, le SP est un complément (oblique).

Pourtant, il semble bien que le statut de *de quelque part* soit le même dans les énoncés :

(5.17) On éloigne quelque chose de quelque part.

2. Les structures sous-jacentes de l'énoncé

		POSITIONNEL	DIRECTIONNEL		
			INITIAL	FINAL	MEDIAN
LOCALISATION	INTERNE	à dans sur	de de chez	jusqu'à	par
	EXTERNE	en face de derrière	de derrière	vers jusque sous	autour de par dessus

FIG. 5.13 - Classification des prépositions spatiales d'après Laur

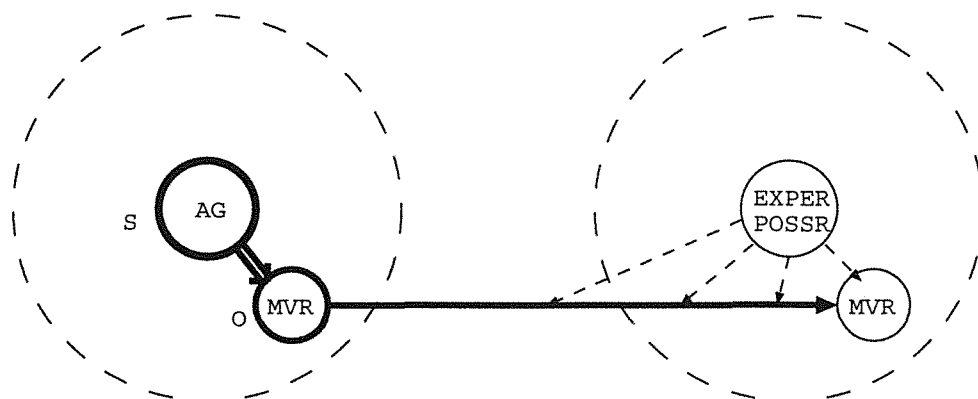
(5.18) On rapproche quelque chose de quelque part.

On peut reprocher à une telle analyse de ne pas tenir compte du fait que le verbe et le complément prépositionnel ne s'appliquent pas forcément en même temps sur l'objet. Certains sémanticiens se sont alors penchés sur la dynamique de l'objet par rapport aux différents actants, notamment Langacker avec sa grammaire cognitive [Langacker, 1991]. Ci-dessous (figure 5.14) est illustrée la dynamique de l'objet dans le cas des verbes *donner* et *recevoir*.

Ag, *mvr* et *experiençer-possessor* sont des rôles thématiques représentant respectivement l'agent, le mobile et le récepteur. L'agent exerce une force qui propulse le mobile (physiquement ou abstraitement), ce dernier passant ainsi de la sphère d'influence de l'agent à celle du récepteur, représentées par les cercles pointillés. La double-flèche représente un transfert d'énergie de l'agent sur le mobile, alors que la simple flèche traduit le mouvement qui en résulte. Les verbes donnent un point de vue différent de la même scène, en ne profilant pas (en gras sur la figure) les mêmes entités et relations. Dans le cas de *donner*, ce sont l'agent (sujet) et le mobile dans sa position initiale (objet direct) qui se trouvent profilés alors que pour *recevoir*, il s'agit du récepteur (sujet) et du mobile dans sa position finale (objet direct). Aussi intéressante que puisse paraître cette analyse, on peut toutefois regretter que, restant à un niveau aussi référentiel, il ne soit toujours pas possible de rendre compte du fait que l'on retrouve la même préposition *de* dans les phrases 5.17 et 5.18. Ainsi, selon nous, le *de quelque part* intervient dans les deux cas dès le début comme point de référence, respectivement *proche* ou *non proche* de l'objet que l'on déplace.

De manière plus générale, il semble que le problème de toutes ces approches provienne du fait que l'on établisse un lien direct entre l'aspect et le mouvement référentiel des objets, et non avec le moment où ces objets devraient être considérés dans le processus inférentiel d'interprétation. On se rapproche ainsi des thèses psychomécanistes de l'école

(a) donner



(b) recevoir

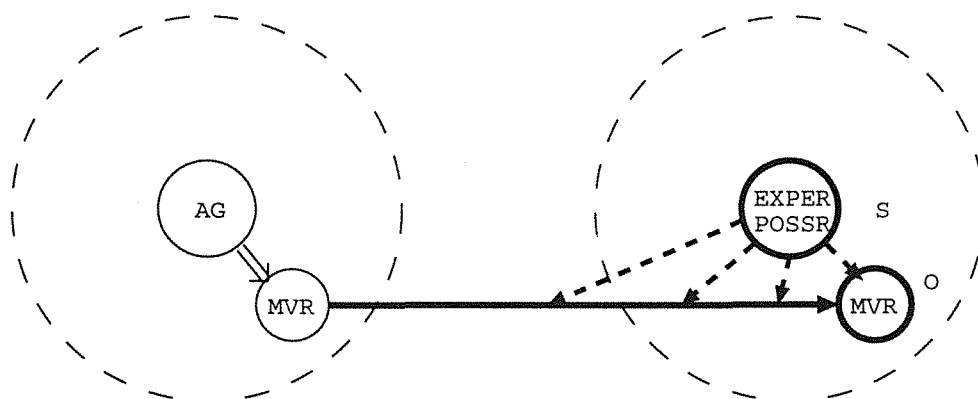


FIG. 5.14 - Dynamique des verbes de transaction selon Langacker

guillaumienne.

Une vision cinétique (plus abstraite) des prépositions L'idée banale que *à* traduit un mouvement d'approche et *de* un mouvement d'éloignement s'élargit à d'autres domaines sémantiques et référentiels que celui du déplacement dans l'espace.

Serres [1974] avait remarqué (avec une certaine verve) qu'il existait bien d'autres domaines auxquels était soumise notre expérience :

« Mon corps, je n'y peux rien, n'est pas plongé dans une variété unique et spécifiée. Il travaille dans l'espace euclidien, mais il y travaille seulement. Il voit dans un espace projectif, il touche, caresse et manie dans une variété topologique, il souffre dans une autre, entend et communique dans une troisième. Et ainsi, tout autant qu'on voudra. » (p. 30)

2. Les structures sous-jacentes de l'énoncé

La différenciation entre domaines est elle-même complexe, tant ceux-ci font intervenir des notions abstraites¹⁷, si bien que l'on ne saurait tracer « une frontière sûre et reconnue entre ce qui serait physique, "réel" et référentiel d'une part, et ce qui serait culturel, imaginaire et non référentiel de l'autre. » [Dervillez-Bastuji, 1982]

On doit alors descendre jusqu'au niveau sémantique si l'on veut espérer appréhender dans un domaine d'application bien particulier l'enchevêtrement entre les différents domaines d'expérience, comme le montre [Langacker, 1991] dans le cadre de la grammaire cognitive. Langacker distingue les domaines cognitifs primitifs (tels que le temps, l'espace, l'ensemble potentiel des sensations de couleur) et les domaines cognitifs non primitifs, qui s'appuient sur des modèles abstraits de hiérarchie inférieure (par exemple le domaine du calendrier, dérivant de celui du temps, par rapport auquel se situent jour, mois, samedi, ...). Il définit alors par rapport à un prédicat la notion de *champ* comme étant *la portion des domaines appropriés que ce prédicat évoque spécifiquement et qui est nécessaire pour sa caractérisation*. Ainsi, pour reprendre notre exemple de départ, le champ des prédicats *se rapprocher* et *s'éloigner* est celui se rapportant à la *distance* domaine qui dérive lui-même du domaine primitif d'espace. À partir de là, il devient aisé d'expliquer que *de* introduit un point-origine par rapport à une distance, comme nous allons le montrer par la suite.

C'est donc seulement par le biais des domaines sémantiques (ou cognitifs) que l'on pourra généraliser des emplois non spatiaux de *à* (ou *de*) en tant qu'extension directe de représentations spatiales. On en vient alors à une vision cinétique dans un espace orienté où, selon le mot de Cadiot [1993], le *à* traduit une « image d'au-delà » (dans un mouvement prospectif) alors que le *de* traduit « une image d'en deçà » (dans un mouvement rétrospectif).

Dans certains cas, ces deux mouvements ne peuvent être déterminés indépendamment d'un point de vue autre que celui du sujet grammatical. On sait par exemple que la position dans l'espace du locuteur est impliquée dans l'interprétation de « venir » (à l'opposé de « aller »):

(5.19) Laurent vient de Paris (vers moi, qui suis à Nancy)

(5.20) Laurent vient à Paris (vers moi, qui suis à Paris)

(5.21) Laurent va à Paris (il s'éloigne de moi)

Dans la série d'exemples précédente, on aurait dans chacun des verbes *venir* et *aller* le schéma suivant pour argument de direction :

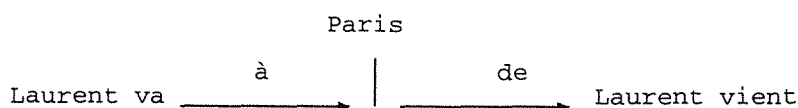


FIG. 5.15 - L'opposition directionnelle entre *à* et *de* selon Guillaume

17. Par exemple, en considérant des notions telles que « champ magnétique » ou « démocratie ».

Cette explication vaut aussi pour notre exemple utilisant *s'approcher de*. *De* indique l'origine, alors que *à* indique la cible. L'énonciation d'un énoncé tel que « Laurent s'approche de Paris » suppose que le locuteur se place mentalement à l'extérieur de Paris, et calcule la situation de Laurent en fonction de ce qui est, de son point de vue à lui locuteur, le point d'origine du mouvement de Laurent. C'est le point d'arrivée (la cible) qui constitue pour le locuteur le point de départ de son calcul, ce par rapport à quoi il veut dire que « Laurent s'approche ». Dans ce cas encore, le nom-régime de *de* est bien la source (dont on s'éloigne), mais à travers l'évaluation et du jugement du locuteur. Cela ne peut en aucun cas être interprété comme une source de l'action de Laurent. On se situe dès lors au niveau du point de vue interprétatif, et non plus du point de vue purement référentiel comme c'était le cas dans les approches précédentes. Pottier identifiant

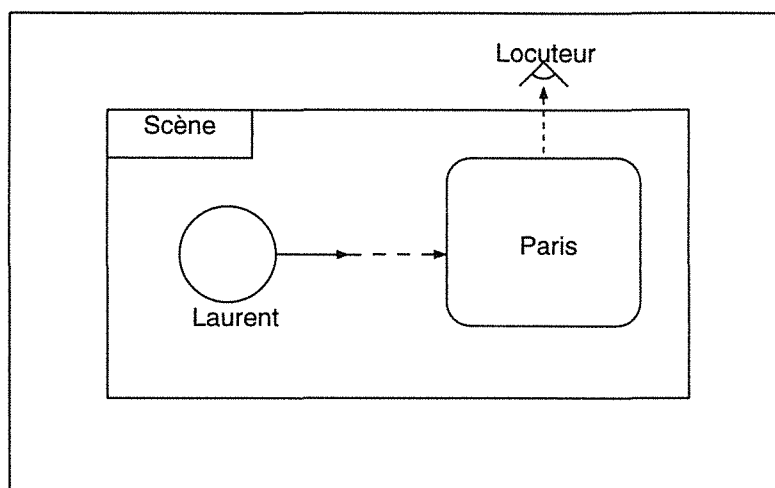


FIG. 5.16 - *Le point de vue du locuteur*

ce phénomène sous le nom de visée, nous représentons le locuteur avec son œil qui vise (cf. figure 5.16) .

Nous retiendrons alors pour *de* et à l'interprétation de Cadiot [1993] :

« *de* correspond à la construction d'une image d'acquis, alors que *à* correspond à celle d'une image à acquérir. Seul l'acquis (l'en-deça) est continu et nous livre une *extension*, c'est-à-dire l'image immédiatement (ou un peu moins ...) fournie avec la prédication dont il procède » (p. 104)

Cervoni [1991], s'inscrivant lui aussi dans une perspective guillaumienne, fournit une explication analogue à propos de l'alternance entre les paires *sur/sous*, *pour/par*. Nous sommes ainsi amenés à nous situer à un niveau d'analyse beaucoup plus fin, où entre en jeu l'ordre d'intervention des éléments constructifs d'images référentielles tout au long du processus d'interprétation.

Mais il convient de délimiter auparavant la portée respective de chacun des constituants, afin de déterminer leur fonction dans l'énoncé. On rentre ainsi dans la grammaire de l'énoncé.

2.3 Fonctions grammaticales et structures de dépendance

À ce niveau, nous pouvons reformuler notre problème en nous interrogeant plus particulièrement sur le rôle de chaque mot de l'énoncé dans la reconstitution d'un passage entre une situation courante et une situation visée. Pour reconstituer des situations, il faut pouvoir exhiber des objets, des propriétés qui leur soient relatives, ainsi que des relations entre eux. Au niveau linguistique, il ne s'agit pas de pouvoir les reconstruire de toutes pièces ; bien des connaissances qui ne sont pas linguistiques entrent aussi en jeu. Il s'agit seulement d'indiquer comment s'y prendre. C'est à ce titre que nous parlerons de la part de spécification propre à chaque mot d'un énoncé dans le processus général de référence. Nous allons établir que cela est directement lié à la notion de fonction syntaxique (ou grammaticale) dans un énoncé.

2.3.1 Rappels sur la notion de fonction syntaxique

De manière générale, on ne saurait s'intéresser à ce que spécifie individuellement un mot donné sans l'avoir préalablement structuré par rapport à d'autres mots. C'est que rares sont les mots qui peuvent fournir une spécification complète à eux seuls sur l'entité (objet, propriété, relation ...) à laquelle ils réfèrent. On ne peut citer comme exception notable que les noms propres (*Toto*, ...) et les indexicaux (*celui-ci*, *tu*, *ici* ...). Il faut alors qu'un mot puisse être en relation avec d'autres éléments pour pouvoir compléter une spécification d'ensemble. C'est précisément à ce niveau que l'on peut délimiter le rôle de la syntaxe.

Relations syntaxiques Si l'on considère par exemple l'énoncé :

(5.22) Liste ces fichiers par pages.

ce que peut spécifier le mot *ces* n'a rien de commun avec ce que peut spécifier le mot *pages*, alors qu'il y a une relation directe avec *fichiers*. Il faut par conséquent qu'il y ait certaines conditions particulières qui soient vérifiées pour que l'on puisse décider si certains mots entrent en relation ou non. Il s'agit de :

- la nature des éléments en question (*par*, *ces*, *pages*, ...) ;
- la position relative qu'ils occupent.

Ainsi, du fait de leur nature et de la position qu'ils occupent, les mots vont mettre en jeu une relation mutuelle où chacun jouera un rôle. Par exemple, dans « ces fichiers », « ces » joue le rôle de déterminant et « fichiers » joue le rôle de déterminé. Dans ce cas particulier, c'est le mot *fichiers* qui se trouve privilégié. D'une manière générale, on cherchera à privilégier un élément de la relation par rapport à l'autre, rendant ainsi la relation univoque. On parlera alors de *fonction grammaticale*. Ainsi, à travers la fonction de *déterminant* sur un nom, le mot *ce* spécifie une désignation (éventuellement ostensive) dans la situation courante d'un objet portant ce nom. La question se pose alors de savoir quand une configuration de mots en relation fournit un ensemble complet (minimal ou non) de spécifications. C'est par ce biais qu'émerge de nouveau la notion de syntagme que nous avons présentée dans le chapitre 2 selon une conception purement distributionnelle.

Une approche fonctionnelle de la notion de syntagme Rappelons la définition du syntagme que nous donne [Grevisse, 1986] :

« Un syntagme est un groupe de mots formant une unité. »

Nous pouvons même aller plus loin dans notre démarche en considérant certains mots comme des syntagmes. Ainsi, si des syntagmes peuvent se combiner pour former des syntagmes d'ordre supérieur, les mots eux-mêmes peuvent être considérés comme des syntagmes combinant des unités linguistiques de hiérarchie inférieure telles que les préfixes, suffixes et radicaux¹⁸. Cette intrusion morpho-syntaxique fait directement écho à la conception du syntagme selon [Benveniste, 1966, p. 126-128] :

« Un syntagme peut donc à la fois être vu comme unité intégrante lorsqu'on cherche à le décomposer en unités plus basses et comme unité constituante lorsqu'on cherche à l'intégrer dans une unité plus haute. »

En termes un peu plus imagés, cela revient à considérer l'énoncé comme un emboîtement de boîtes les unes dans les autres, où une boîte extérieure donnerait une « raison d'être » aux boîtes qu'elle contient. Nous sommes à présent en mesure de relier plus précisément les notions de syntagme et de fonction : *ce qui fait unité dans une unité linguistique peut se définir comme une fonction (ou un rôle) par rapport à l'unité linguistique de niveau supérieur*. On voit ainsi apparaître la dualité entre constituants (mots, syntagmes) et relations entre certains de ces constituants (fonctions).

On peut dès lors envisager l'analyse syntaxique d'un énoncé selon deux approches, suivant que l'on privilégie une composante ou l'autre. Soit on cherche à expliciter un emboîtement de structures, et à partir de là on exhibe des fonctions. Soit on cherche à expliciter directement des fonctions entre mots, et à partir de là on exhibe des emboîtements de structure.

2.3.2 Une première approche : la primauté aux emboîtements de syntagmes

Cette approche a pour conséquence de mettre au premier plan la notion de syntagme, réduisant celle de fonction à une notion dérivée. C'est typiquement ce point qui a préoccupé l'école générativiste chomskienne pour laquelle une fonction grammaticale était déduite de la position absolue de deux syntagmes par rapport à la phrase [Ducrot et Todorov, 1972] (p. 277).

Le point de vue global générativiste. Une question que les générativistes se sont posée est la suivante : étant donné un arbre syntagmatique, comment déduire les fonctions reliant les mots ou les morphèmes de la phrase sans addition d'informations étrangères à celles contenues dans les règles qui ont engendré la phrase ? Considérons à titre illustratif la phrase :

(5.23) Il écrit une lettre en deux exemplaires.

18. Par exemple, le préfixe lié *re-* dans « Réimprime ce fichier. » pourra correspondre à un substitut de « une deuxième fois » dans « Imprime ce fichier une deuxième fois. ». Dans une certaine mesure, le mot « rapidement » pourra correspondre de même à un substitut du syntagme « avec rapidité ».

2. Les structures sous-jacentes de l'énoncé

Il s'agit d'extraire les fonctions de sujet et de complément d'objet direct entre les mots composant cette phrase. Sa structure est typiquement la suivante (5.17) :

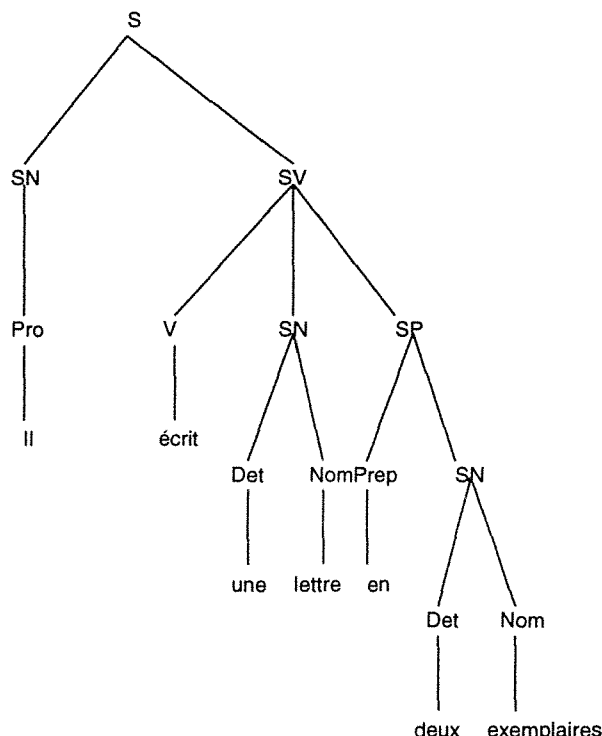


FIG. 5.17 - *Arbre syntagmatique standard de Il écrit une lettre en deux exemplaires*

On pourra alors poser la règle :

Un segment X est sujet d'une phrase s'il est dominé par un noeud SN immédiatement dominé par le noeud S qui domine la phrase.

On définira de manière analogue la relation « Etre verbe principal d'une phrase ». On en déduira alors que *Il* est sujet de *écrit* simplement en posant une règle du genre :

X est sujet de Y si X est sujet d'une phrase et Y est verbe principal de cette phrase.

De la même façon, pour déduire que *une lettre* est complément d'objet direct (C.O.D.) de *écrire*, on pourra passer par une règle :

Un segment X est C.O.D. d'une phrase s'il est dominé par un noeud SV immédiatement dominé par le noeud S qui domine la phrase.

que reprendra une règle du même type que « X sujet de Y » :

X est C.O.D. de Y si X est C.O.D. de cette phrase et Y est verbe principal de cette phrase.

Nous avons pu d'ailleurs constater que certaines méthodes classiques procédaient ainsi. Mais pour ne citer qu'un inconvénient, soulignons celui de devoir remonter jusqu'au niveau de la phrase elle-même alors que les fonctions que l'on cherche à caractériser sont locales. Cela a pour effet que les fonctions locales à l'intérieur d'un syntagme n'occupant pas une position fixe se voient totalement passées sous silence (c'est notamment le cas pour les syntagmes nominaux).

Un progrès sensible a alors été accompli avec les grammaires lexicales fonctionnelles (G.L.F.), où l'on considérait toujours un emboîtement similaire de syntagmes, mais où l'on s'intéressait à la position *relative* de deux syntagmes pour exhiber une fonction.

Le point de vue local lexico-fonctionnel. La théorie lexico-fonctionnelle fait partie, comme nous l'avons vu au chapitre 2, de la catégorie des grammaires d'unification, reposant toutes sur un ensemble de structures attribut-valeur. Partant d'une conception syntaxique de la phrase sujet/prédicat, un mécanisme d'unification réalise des emboîtements de structures fonctionnelles à l'aide de spécifications du type :

- $\uparrow = \downarrow$ (recopie les caractéristiques d'un syntagme de hiérarchie inférieure sur celui de hiérarchie supérieure)
- $(\uparrow \text{ fonction}) = \downarrow$, la fonction en question pouvant être calculée directement ou donnée.

La figure 5.18 donne un exemple de structure fonctionnelle que l'on peut obtenir de cette manière à partir d'un arbre syntagmatique. Toutefois, un tel mécanisme d'association de

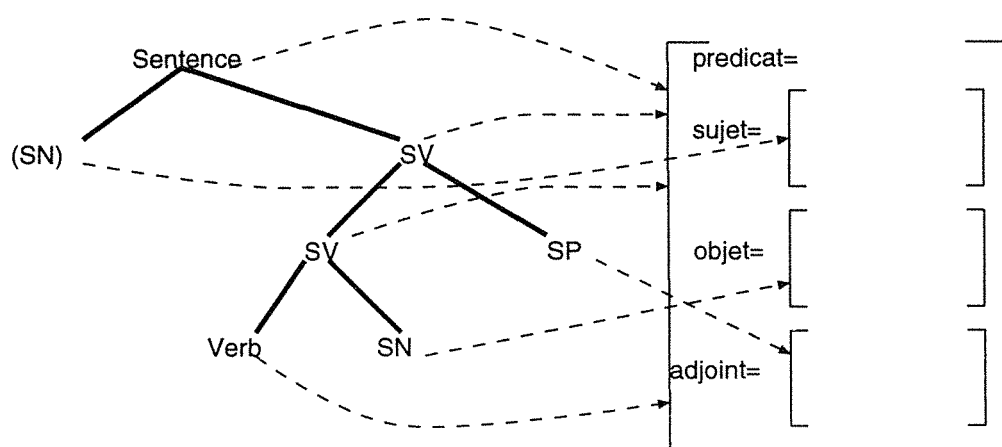


FIG. 5.18 - Structure fonctionnelle

structures fonctionnelles pose divers problèmes [Duermael et Pierrel, 1992] :

- 1° On aboutit de cette manière à une redondance de structures fonctionnelles que l'on retrouve aux différents niveaux successifs (cf. figure 5.18). On remarque en effet que la même structure fonctionnelle est associée au verbe (issue du lexique par insertion lexicale), aux SV et à l'énoncé¹⁹.

19. Nous avons vu combien il était difficile de rattacher tous les compléments sous le SV au prédicat correspondant au verbe. C'est que le nombre de compléments obligatoires dans une proposition dépend

2° Par rapport à la définition de Benveniste de la fonction associée à un syntagme (cf. page 154), on notera une différence majeure de conception : on se place ici du point de vue du syntagme de hiérarchie inférieure (\downarrow) et non de celui de hiérarchie supérieure (\uparrow), ce qui pose des problèmes évidents de visibilité du contexte.

3° Le fait d'introduire un nom de fonction comme attribut introduit une contrainte inutile : qu'il n'y ait pas deux fonctions de même nom à un niveau donné. Or, dans un syntagme nominal comme *un vieux chat tigré avec un collier*, les syntagmes *vieux*, *tigré* et *avec un collier* sont au même niveau et ont la même fonction de *complément de détermination*²⁰.

Face aux problèmes soulevés par la mise en évidence des fonctions grammaticales à partir des imbrications de syntagmes, on peut être tenté d'inverser la perspective en accordant la primauté aux fonctions grammaticales, lors de l'analyse de l'énoncé.

2.3.3 Une seconde approche : la primauté aux fonctions

Benveniste montre la nécessité de raffiner le critère distributionnel de substitution pour dépasser une conception purement formelle de la langue. Cela revient à se poser la question : « Qu'est-ce qui fait unité dans un groupe de mots ? ». L'argument qu'il avance est la possibilité d'isoler une unité linguistique en l'opposant aux autres²¹. On ne peut alors parler d'unité linguistique que si elle peut être identifiée dans une unité plus haute. C'est cette unité intégrante qui donne sens aux unités qui la composent.

Parmi les unités linguistiques composant un syntagme, il en est une qui est principale : la tête (ou *head* dans la littérature anglaise), qui donne la nature du syntagme :

- le syntagme nominal, avec pour tête le nom (auquel participent le groupe déterminant, l'adjectif) ;
- le syntagme prépositionnel, avec pour tête la préposition (auquel participe le syntagme nominal) ;
- le syntagme verbal, avec pour tête le verbe (auquel peuvent participer les syntagmes nominaux et prépositionnels).

Les théories linguistiques qui accordent une primauté aux fonctions vont isoler la tête des syntagmes, et privilégier les relations entre la tête d'un syntagme et les autres éléments qui participent à ce syntagme. Cela revient par conséquent à privilégier en premier lieu les fonctions de dépendance entre mots plutôt que l'emboîtement de ces mots les uns dans les autres. Nous allons étudier les avantages d'une telle approche à travers diverses théories linguistiques faisant appel à des structures de dépendance.

en fait de la valence du verbe, qui dépend elle-même du sens que l'on veut bien retenir pour le verbe. Les compléments obligatoires trouvent vite leur rôle fonctionnel. Mais avec la visibilité dont dispose le module de calcul des structures fonctionnelles, il est difficile de faire le tri entre compléments facultatifs régis par le verbe et circonstants de la proposition. Le défaut est de supposer que les contraintes sont connues une bonne fois pour toutes avant-même l'analyse, sans qu'il n'y ait besoin de prendre en compte les informations contenues dans les énoncés à analyser.

20. Les GLF contournent ce problème en collectant les éléments dans un ensemble d'attribut XADJUNCT.

21. Ce caractère différentiel est typique des systèmes sémiotiques.

Les relations structurelles de Tesnière. La relation de dépendance est issue directement de ce que [Tesnière, 1969] appelle la *connexion*. Tesnière définit en effet la connexion comme une relation établissant entre les mots des rapports de dépendance. Chaque connexion unit en principe un terme supérieur (le « régissant ») à un terme inférieur (le « subordonné »). La connexion à elle seule permet de rendre compte de phrases simples. Mais elle n'explique pas tout. Tesnière introduit deux relations supplémentaires : la *translation* et la *jonction*. La translation consiste à faire changer la catégorie d'un mot, alors que la jonction permet de joindre deux mots de même niveau. La figure 5.19 montre comment peuvent se combiner ces trois types de relation à travers un simple énoncé de commande :

(5.24) Tu crées un répertoire commun à toute l'équipe dans le home « dialogue ».

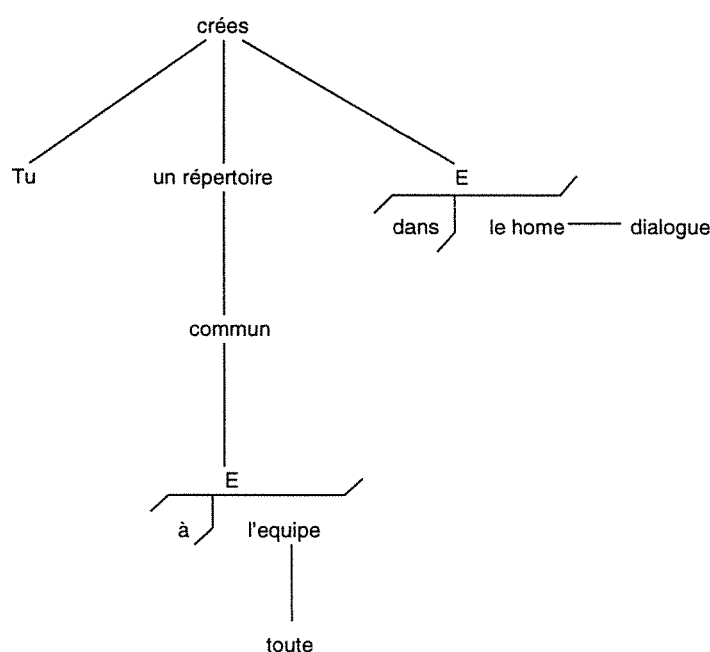


FIG. 5.19 - Un exemple simple d'énoncé illustrant les connexion, translation, jonction

Le translatif « dans » permet de changer la catégorie nominale de *home* en adverbiale (catégorie E).

Tesnière répartit les subordonnés immédiats du verbe en actants et circonstants (de catégorie E). Dans notre cadre restreint, un test que l'on peut effectuer pour déterminer si un syntagme a un rôle d'actant ou n'est qu'un circonstant concerne l'influence de celui-ci par rapport au résultat. Cela est à mettre directement en relation avec la notion de paramètre que nous avons définie dans le cadre (sémantique) des verbes d'action :

« Un élément sera considéré d'un point de vue sémantique comme paramètre s'il affecte le déroulement de cette action. » [Clark, 1986]

Mais cela est indécidable au niveau grammatical ; il y a *a priori* deux solutions envisageables en faisant abstraction de critères sémantiques : il s'agit soit d'un complément verbal, soit d'une fonction (indéterminée) par rapport au discours.

L'approche de Jespersen. Dans son approche de la grammaire, Jespersen [1971] (publié pour la première fois en 1924) ouvre la voie sur une perspective de représentation syntaxique des énoncés qui a le mérite d'être « locale » et adaptable à la symbiose entre référence aux objets et référence aux actions. Une première conclusion à laquelle il aboutit est de ne pas s'appuyer sur une structure de phrase. Cela implique alors une approche locale aux syntagmes.

Des liens de dépendance locaux. Jespersen s'appuie sur des groupes de mots d'autonomie relative. Il attribue un rang aux divers mots de la langue reflétant leur degré d'autonomie. Ce rang est attribué en se fondant sur la classe du mot, mais également sur des bases contextuelles

Si l'on cherche à connecter le mot *très* (pour reprendre la terminologie de Tesnière), on constate qu'il doit s'appliquer à un adjectif (comme *petit* ou *bleu*). Cela implique de considérer différents rangs d'autonomie relative. On est ainsi parvenu à une conception où chaque petit groupe de mots forme une entité. Les liens entre groupes s'effectuent du moins autonome au plus autonome (du rang i au rang $i-1$).

Alors que nous avons analysé la rupture de propriété à un niveau sémantique, la particularité de l'approche de Jespersen est de l'introduire au niveau structural, sous l'appellation de *nexus*, par opposition à la *jonction*²² qui n'introduit aucune rupture.

Une approche structurale de la rupture temporelle. Le *nexus* peut se manifester de différentes sortes. Cela est flagrant dans les exemples suivants, où il y a deux ruptures dans chaque phrase, l'une relative à *voir* et l'autre relative à *arriver*.

(5.25)

- (a) Je vis le docteur arriver.
- (b) Je vis que le docteur arrivait.
- (c) Je vis l'arrivée du docteur.

Jespersen isole dans ces énoncés deux entités formant un *nexus* : (*arriver, arrivait, l'arrivée*) d'une part et (*le docteur, du docteur*) d'autre part. Il montre que ces deux éléments forment dans chaque phrase l'objet de *voir*.

Nous ne suivrons pas totalement l'approche de Jespersen, dans le sens où notre approche aspectuelle en termes de ruptures, utilisera des spécifications préalables liées aux relations de dépendances, sur un plan purement fonctionnel, en faisant abstraction du déroulement.

Le modèle de Mel'Čuk. Mel'Čuk [Mel'čuk et Pertsov, 1987; Sabah, 1988] a adopté un modèle très proche de la syntaxe structurale de Tesnière, dans l'une des couches (syntaxe de surface) qu'il propose dans son modèle Sens↔Texte.

22. Appellation qui n'a rien à voir avec la jonction de Tesnière, la jonction de Jespersen n'étant qu'un cas particulier de connexion.

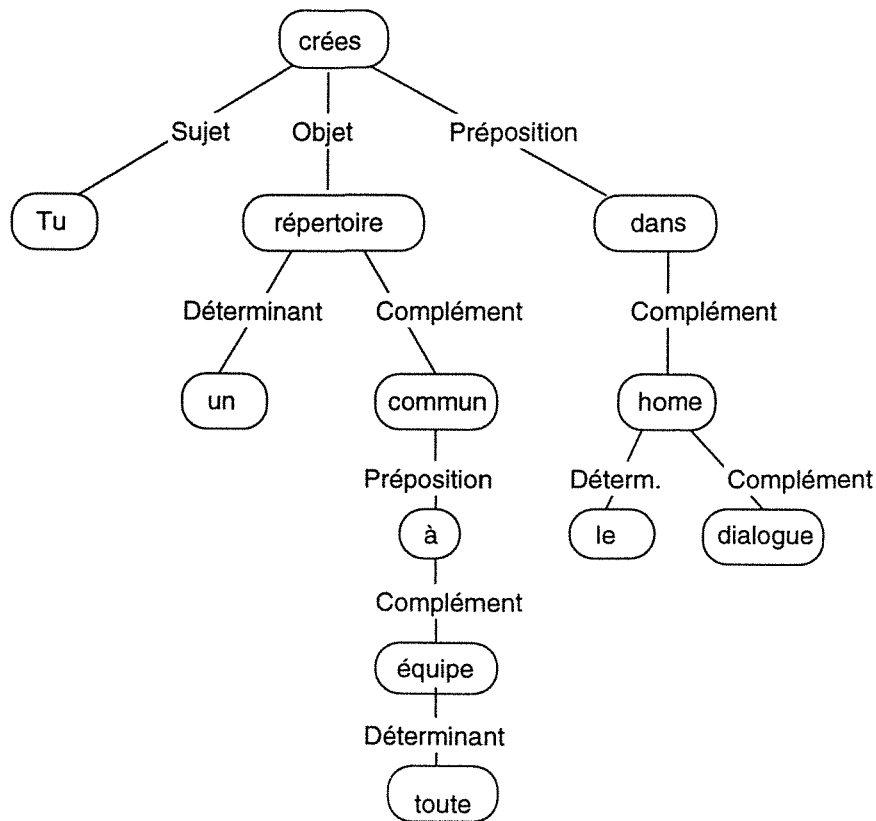


FIG. 5.20 - Le même énoncé avec le modèle de Mel'Čuk

La représentation qu'effectuerait Mel'Čuk de l'énoncé 5.24 ressemble beaucoup à celle de Tesnière, excepté que le phénomène de translation est ici gommé. Mel'Čuk fait intervenir la notion de complément prépositionnel, qui existe aussi chez Jespersen :

« Lorsqu'un adverbe a un objet, il devient ce que l'on a coutume d'appeler une préposition. » ([Jespersen, 1971], p.223)

Là où Tesnière ne fait intervenir que trois relations, Mel'Čuk ne recense pas moins de douze fonctions. Parmi ces fonctions, citons la fonction *préposition* qui n'est qu'un cas particulier de la translation de Tesnière avec un translatif correspondant à une préposition. D'une manière générale, les facteurs qui contribuent ici à spécialiser les fonctions sont :

- la classe de l'élément dominé ;
- la classe de l'élément dominant ;
- leur position relative.

En fin de compte, il apparaît que de faire reposer la structure de l'énoncé sur des fonctions de dépendances entre les mots (ou locutions²³) qui le constituent assainit l'approche que l'on peut avoir de la notion de fonction. En effet, si l'on reprend point par point les objections que nous avons formulées à l'encontre des grammaires lexicales fonctionnelles, il s'avère que :

- 1° Il n'y a plus de redondance des fonctions à travers la structure de l'énoncé ;
- 2° La fonction est ici définie par rapport à l'unité linguistique de niveau supérieur (définie à partir du nœud - dominant - à la racine de la portion d'arbre considérée). On peut en effet reconstituer pour chaque nœud de l'arbre un syntagme ayant pour tête ce nœud²⁴ ;
- 3° Un nœud peut avoir plusieurs fonctions de même nom, correspondant à autant de branches de dépendance.

3 Conclusion

Dans ce chapitre, nous avons caractérisé diverses propriétés sémantiques de l'énoncé de commande (l'aspect, l'Aktionsart, la transitivité, ...) qui permettent de spécifier la construction d'un état résultant. Notre hypothèse est qu'elles participent au processus de construction de la structure de l'énoncé. En partant d'une conception « guillaumienne » de l'aspect sous forme d'un couple [*tension, détension*] (ou [*potentiel, accompli*]), cela implique notamment de prendre en compte l'ordre dans lequel interviennent les participants de l'énoncé. Nous avons enfin établi qu'en mettant directement en évidence les fonctions grammaticales qui sont liées au calcul aspectuel, on était amené à se représenter la structure de l'énoncé en termes de dépendances entre les mots qui composent cet énoncé. Rappelons qu'il s'agit pour l'analyse grammaticale de délimiter les spécifications qui peuvent se suffire à elles-mêmes au niveau du groupe de mots. Dans une grammaire de dépendance, cela revient pour les nœuds à collecter les spécifications des nœuds dont ils dépendent, en plus de la spécification à leur niveau. On reconstitue de cette manière le syntagme correspondant au nœud dominant. Le facteur qui prime est dans ces conditions *la cohésion des mots à travers les relations de dépendance qu'ils entretiennent*. C'est sur cette conception de la grammaire, fondée sur des dépendances fonctionnelles entre des mots, que nous allons bâtir notre modèle de construction d'objets du discours.

23. C'est notamment le cas pour les locutions adverbiales ou prépositionnelles. On peut citer par exemple à *propos de* ou *jusqu'à* comme locution prépositionnelle. De l'autre côté, *dans deux jours* a été intégré sous la forme de *après-demain*. On voit pleinement jouer ici le bras de fer entre la morphologie qui n'a pas intégré certains syntagmes en mots et la syntaxe qui cherche à les relier par des fonctions de dépendance. Une solution est de considérer des nœuds atomiques (des mots) et des nœuds non atomiques (des locutions). Extérieurement, le comportement syntaxique est identique [Mel'cuk et Pertsov, 1987]. Cela correspond au *nucleus* de Tesnière.

24. On peut ainsi retrouver une imbrication de syntagmes les uns dans les autres.

Troisième partie

**Proposition d'un modèle et
validation**

Chapitre 6

Un modèle de construction d'objets du discours pour référer aux actions

Sommaire

1	L'élaboration de schèmes de déroulement	166
1.1	La structure des schèmes de déroulement	166
1.2	Identification des objets et degré d'autonomie référentielle	167
1.3	Les phases du déroulement	169
1.4	La modélisation de l'interaction contextuelle : l'intervention des promoteurs	181
2	Les promoteurs comme pivots de la construction de zones temporelles	187
2.1	De la dépendance structurelle à la dépendance sémantique	188
2.2	L'influence des spécifications sémantiques	191
2.3	Formalisation des règles de construction de zones temporelles	195
3	Conclusion	198

Nous développons dans ce chapitre les bases théoriques constituant notre apport propre dans le domaine de la référence aux actions. Le chapitre suivant aura pour but de justifier nos différents choix de mise en œuvre, aussi ne rentrerons nous pas à ce niveau dans des considérations d'implémentation informatique. Ne sont concernées ici que les bases d'une spécification.

Puisqu'il s'agit de proposer un modèle de construction d'objets du discours pour référer aux actions, la première question que nous nous poserons est de savoir sur quoi l'on construit. Cela revient à s'interroger sur la structure de l'énoncé. Dans le chapitre précédent, nous avons été amenés à effectuer des choix en nous appuyant sur quelques variétés de grammaires de dépendance. Pourtant, un problème bien connu est le nombre d'arbres proposés par un analyseur pour un même énoncé [Wehrli, 1987]. Cela correspond au fait que, partant d'une forme linéaire, on peut parvenir à plusieurs structures de dépendance. Ensuite, choisir la bonne est une affaire de stratégie d'interprétation (par échecs

successifs, en parallèle, par factorisation, au fur et à mesure de l'analyse grammaticale, ...) [Ait-Kaci et Lincoln, 1989]. Nous considérons pour cette raison que l'interprétation commence à partir du moment où l'on construit ce qui a été spécifié par les mots en relation les uns avec les autres. De ce fait, vouloir privilégier une structure possible parmi d'autres est au-delà des compétences de la grammaire car nous prétendons qu'au niveau de la grammaire on ne reste qu'au stade de la spécification. Le rôle de la grammaire est de délimiter les spécifications qui peuvent se suffire à elles-mêmes au niveau du groupe de mots, les mots en dépendance venant rajouter leurs spécifications respectives à celles du mot qui les intègre. Pour reprendre une formule de Marandin¹, il ne faut pas faire dire à la syntaxe (ou la grammaire) plus qu'elle veut bien nous en dire.

A partir de là, on peut envisager une analyse complémentaire de l'énoncé en termes de ruptures de propriétés. Nous allons modéliser cette analyse par la construction de schèmes dits de déroulement, qui serviront de tremplin à la construction de zones temporelles. Nous pourrions alors proposer un formalisme qui rende compte de ces diverses interactions.

1 L'élaboration de schèmes de déroulement

La référence aux objets a pour fonction d'isoler des entités, alors que la référence aux actions a pour fonction de montrer comment elles évoluent, quels changements elles subissent. Plus précisément, nous avons vu que le point important était de montrer comment sont provoquées des ruptures de situation. Cela sous-entend de préciser les lignes directrices que nous avons déjà évoquées pour dresser une sémantique des ruptures. La catégorie de l'aspect que nous avons discutée dans le chapitre précédent peut s'interpréter comme la mise en correspondance de la vision d'une programmation spatialisée du procès avec son déroulement temporel effectif – ce qui se marque par des bornes, des limites, des termes. Ces points sont à la base de la structure des schèmes de déroulement, comme nous allons le préciser.

1.1 La structure des schèmes de déroulement

Un schème de déroulement a la forme d'un graphe, pour lequel on distingue deux types de nœuds :

- les nœuds pour identifier des entités, faisant entrer en jeu des propriétés et relations statiques ;
- les nœuds indiquant comment ces entités évoluent.

Une dynamique (acquis, \neg acquis) permet d'orienter les arcs du graphe et d'introduire ainsi une relation d'ordre partiel. Le graphe étant orienté, nous pouvons munir l'ensemble des nœuds d'une relation d'ordre partiel. Le but est d'imposer des *contraintes générales au niveau du processus d'interprétation*. Cette dynamique (acquis, \neg acquis) n'est pas à confondre avec les ruptures de propriétés au niveau des objets ($P(O_i)$, $\neg P(O_i)$) que nous avons développée au début du chapitre 4. La dynamique dont il est question ici concerne

1. À un séminaire du pôle Cogniscience Est de l'Association pour la Recherche Cognitive.

des contraintes au niveau de l'ordonnancement du calcul de référence, des spécifications devant être actualisées avant d'autres, afin de guider l'actualisation des suivantes.

Soient A et D deux nœuds d'un graphe \mathcal{G} . S'il existe un arc l tel que $l(D, A)$, alors on dira que D est le *donneur* et A l'*accepteur*. D est alors plus grand que A , en ce sens que l'interprétation de A passe par l'interprétation de D . Une condition pour que D puisse exercer son rôle de donneur sera alors qu'il ait un acquis actualisé. Au niveau de la négociation du sens, cela se traduit par le fait que c'est le donneur qui propose le premier et l'accepteur qui négocie (ou dispose).

Nous allons détailler à présent le premier type de nœud rentrant dans la composition du graphe : ceux utilisés pour identifier des objets.

1.2 Identification des objets et degré d'autonomie référentielle

Les *objets* dont il est question dans ce paragraphe correspondent à toute entité à laquelle on peut faire référence à l'aide d'un nom. De manière générale, on identifie un objet par rapport à autre chose (un repère), de manière plus ou moins définie. Il existera de ce fait toute une échelle de degrés d'autonomie (§1.2.1) et de définition (§1.2.2).

1.2.1 La hiérarchie des degrés d'autonomie

Nous concevons ici la notion d'objet au sens large. Nous appelons en effet *objet* tout ce à quoi peut référer un nom [Dervillez-Bastuji, 1982].

Il faut pouvoir repérer un objet par rapport à un lieu (qui constitue de fait le repère). Ce lieu peut lui-même être considéré comme un objet par rapport à un autre repère. Dans la situation que l'on décrit, le repéré sera toujours moins autonome que le repère. De la même façon, ce qui rentre dans la composition d'un objet est moins autonome que l'objet lui-même.

Notons que, même au niveau des propriétés, il y a différents degrés d'autonomie, variant selon la position des mots dans la phrase qui les spécifient. Par exemple, dans :

(6.1) Le ballon est rouge.

la propriété *être rouge* est subordonnée à *être un ballon* et de ce fait, moins stable que la propriété dont elle dépend. Cela signifie que l'on peut changer de couleur tout en restant un ballon. Et que si l'on prédique sur le ballon qu'il est rouge, c'est par opposition avec toutes les autres couleurs qu'il aurait pu avoir. L'autonomie se situe dans cet énoncé au niveau de *le ballon*. En revanche, dans :

(6.2) Le ballon rouge est sorti du terrain.

elle se situe au niveau de *le ballon rouge*.

1.2.2 La définition de l'objet

Pour qu'un objet puisse servir de borne, il est important de pouvoir délimiter son étendue spatio-temporelle. Par exemple, dans la phrase

(6.3) Le petit chat est aveugle à la naissance.

il ne s'agit pas d'un petit chat particulier, mais de tous les petits chats déjà nés ou à naître. L'étendue spatio-temporelle où les petits chats peuvent naître n'est donc pas délimitée dans la phrase 6.3.

Quantité de facteurs entrent en jeu pour l'attribution de cette généralité [Gaiffe, 1992]. Dans la phrase 6.4 (a), le simple fait d'employer un passé composé à la place d'un présent impose une limite à l'étendue spatio-temporelle où les petits chats peuvent boire du lait, ce qui conduit à une interprétation spécifique du même syntagme *le petit chat*. On peut vérifier ceci dans la phrase 6.4 (b) où le seul changement de temps grammatical autorise de nouveau une interprétation générique.

(6.4)

(a) Le petit chat a bu tout le lait qu'on lui avait donné.

(b) Le petit chat boit tout le lait qu'on lui donne.

Lorsque rien n'imposera de borne à l'étendue d'une classe, nous la représenterons avec un cercle en pointillé. On se situe lors de l'interprétation au niveau du type de l'objet (*intensionnel*) et non de l'individu (*extensionnel*) (distinction introduite en 1882 par [Frege, 1971], [Berrendonner *et al.*, 1992]). On représentera par un entourage² continu les entités correspondant à un ensemble borné (cf. figure 6.1) et par un entourage en pointillé celles correspondant à un ensemble non-borné (cf. figure 6.2).

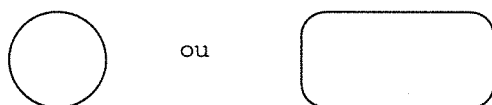


FIG. 6.1 - Ensemble d'objets borné

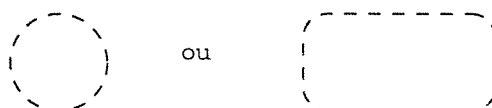


FIG. 6.2 - Ensemble d'objet non borné

L'identification de l'objet joue par conséquent un grand rôle. Dans l'énoncé 6.5, il n'est pas question d'identifier *les carrés à dessiner* : la seule connaissance de leur type est en effet suffisante.

(6.5) Dessine des carrés.

Mais on rencontrerait une difficulté avec un énoncé du genre de 6.6, présentant pourtant la même forme de surface que l'énoncé 6.5 :

(6.6) Efface des fichiers.

2. Circulaire ou arrondi aux angles, pour des raisons de présentation.

Le système sera contraint de demander à l'utilisateur : « *Lesquels?* », puisque l'effacement est à l'initiative de l'utilisateur (qui commande et non à celle du système, qui lui obéit). Il s'agit, on le voit, d'un problème d'existence préalable. Les marqueurs syntaxiques sont ainsi loin d'être suffisants pour la définition de l'objet : de nombreux critères sémantiques et référentiels agissent en parallèle.

Dans les énoncés de commande que nous rencontrerons, le sujet sera toujours défini, bien que non spécifié à l'impératif (nous l'appelons *TU*). Le problème se posera davantage au niveau du complément d'objet verbal.

1.3 Les phases du déroulement

Il s'agit d'explicitier ici les éléments du graphe qui indiquent comment les entités évoluent par rapport au degré d'avancement du procès (ou encore son état d'engagement). Il faut alors débiter par un processus d'amorçage qui partira de ce qu'il y a de plus autonome.

1.3.1 Le point de départ

Le *TU* est autonome et représente ainsi une entité à part entière. Dans le cas de l'impératif (et en général, pour les actes directifs), nous le considérerons comme point de départ pour la construction de nos schèmes de déroulement.

MAINTENANT

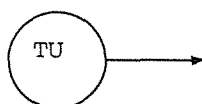


FIG. 6.3 - *Processus d'amorçage*

1.3.2 La contribution du verbe seul

Ce cas recouvre les verbes intransitifs ou les verbes transitifs employés de manière intransitive, comme dans :

(6.7) Imprime.

Avec un verbe seul, nous nous limitons à la phase comportementale du verbe par rapport à son sujet. Nous avons déjà vu que le propre du verbe était d'introduire une tension – ou potentiel de continuabilité – ne demandant qu'à s'écouler. Nous allons montrer que cela peut se modéliser en introduisant un *élément émetteur*³ au niveau du verbe.

3. Dans une conception analogue, [Moignet, 1973] employait le terme d'*élément causatif*, terminologie qui peut entraîner des confusions pour des verbes tels que *subir, tomber, trébucher, ...*

L'élément émetteur : On peut considérer l'élément émetteur du verbe comme un module qui produit (actualise) de la tension, lorsqu'il y est autorisé par le sujet. L'élément émetteur cherche à s'appliquer au sujet. Cela veut aussi dire que le sujet doit avoir une propriété susceptible de fournir une tension de départ (une impulsion) compte tenu du domaine où l'on se place. On retrouve ici la condition que nous avons explicitée au chapitre 4 selon laquelle le sujet doit être capable d'effectuer ce qu'on lui demande.

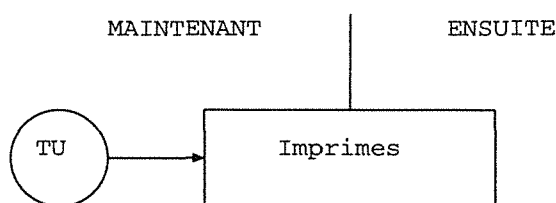


FIG. 6.4 - La rupture temporelle du verbe

En plus d'une prédication sur le sujet, il se produit à ce niveau une rupture de situation par rapport à la situation *maintenant*, introduisant ainsi une nouvelle situation *ensuite*. On peut interpréter le déroulement relatif à cette dernière situation comme une reproduction infinie d'un état récurrent alimentée par le sujet, comme nous l'expliquons dans le paragraphe suivant.

Uniformité de déroulement et succession d'états : Un tel phénomène se retrouve notamment au niveau discret événementiel. Dans la phrase :

(6.8) Il est entré puis est ressorti.

le « re- » porte sur la reproduction d'un état précédent⁴. C'est la propriété *être hors de* qui se trouve reproduite. De la même façon, dans la phrase :

(6.9) Il lui a donné et lui a repris.

c'est la propriété *être en sa possession* qui se trouve reproduite. D'une manière un peu plus générale, c'est une succession d'états similaires par rapport au sujet qui fait que l'on a uniformité. Il faut en effet un potentiel pour continuer à partir de ce qui a déjà été accompli, accompli nécessitant lui-même un potentiel préalable.

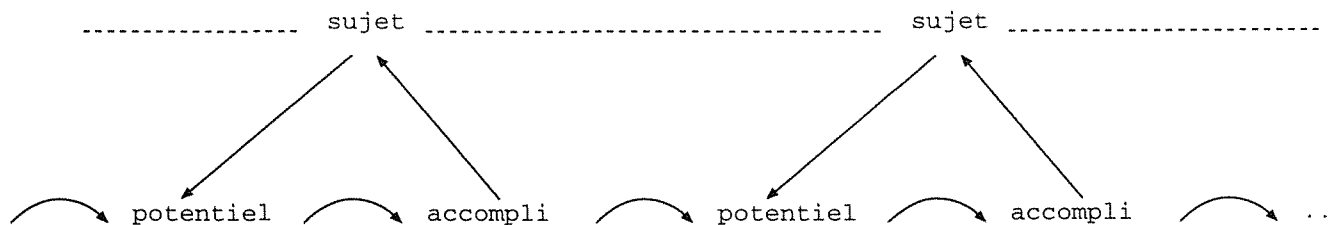


FIG. 6.5 - La dualité entre potentiel d'accomplissement et déjà accompli

4. Cette interprétation n'est toutefois pas générale, ne valant pas pour *rentrer*.

1. L'élaboration de schèmes de déroulement

Il s'agit là d'une explicitation en extension d'un mécanisme global que synthétisent les schèmes que nous avons introduits, qui se centrent uniquement sur les ruptures, initialisant ou mettant fin à la chaîne de déroulement ci-dessus (figure 6.5). L'intérêt des schèmes de déroulement est en effet essentiellement visuel, mettant en relief les ruptures entre situations.

Une représentation plus détaillée sous forme de graphe : Les graphes de déroulement que nous allons développer sont équivalents aux schèmes de même nom sur le plan conceptuel, mais se rapprochent davantage de la structure informatique qui les représente. Dans le cas présent, le graphe complet correspondant serait (cf. figure 6.6) :

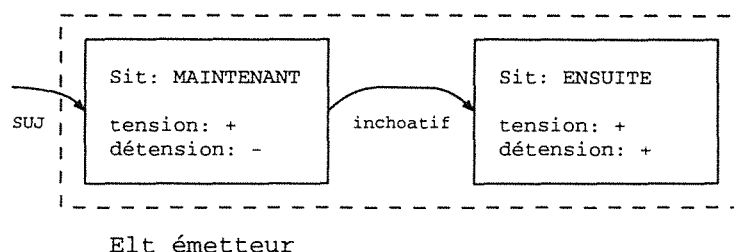


FIG. 6.6 - Graphe d'un verbe intransitif

Nous avons traité ici d'un verbe transitif employé sans objet. Or, pour reprendre la conception de Pottier d'un verbe transitif, ce qui fait qu'un verbe est transitif est qu'il suggère un au-delà. En l'occurrence, pour le verbe *imprimer*, il s'agit d'un fichier. Or, nous avons vu qu'un objet pouvait être plus ou moins déterminé. Lorsqu'il n'est pas exprimé, on peut voir là un cas extrême d'indétermination :

(6.10) Imprime ...

Dans ce cas, le schème de déroulement ne comporte pas non plus de rupture de situation au-delà en *ensuite*, comme illustré dans la figure 6.7 :

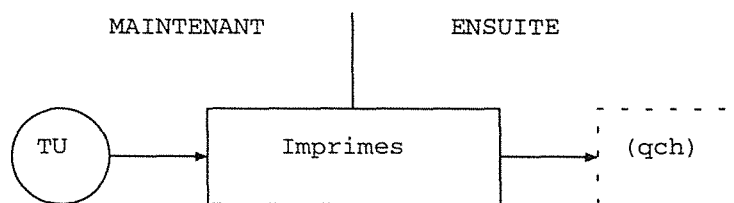


FIG. 6.7 - La rupture temporelle du verbe transitif sans objet

Nous allons à présent nous intéresser au cas où l'objet figure plus explicitement.

1.3.3 La contribution de l'objet

L'objet dont il est question dans ce paragraphe est un constituant linguistique (ayant fonction de complément du verbe) qui réfère à un ensemble d'objets, possédant *a priori* un degré d'autonomie quelconque. Deux cas sont alors à distinguer, selon que cet ensemble est borné ou non.

Ensemble d'objets borné et élément récepteur: Un ensemble d'objets parfaitement délimité peut s'obtenir en employant un groupe nominal démonstratif⁵ :

(6.12) Imprime ce fichier.

Nous avons ici affaire à une borne pouvant provenir d'une instruction référentielle de l'article démonstratif, compte tenu de la situation courante. Le schème de déroulement auquel on aboutit est alors le suivant (cf. 6.8) :

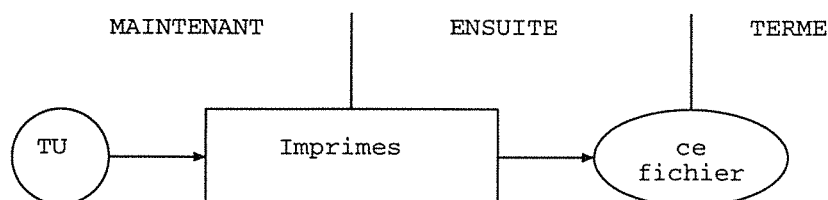


FIG. 6.8 - La rupture temporelle avec objet direct

Parallèlement à l'*élément émetteur* que nous venons de décrire, on peut considérer pour les verbes transitifs un *élément récepteur*, sous la forme d'un module qui consomme de la tension pour la dégrader en détension. Cette dégradation ne pourra s'effectuer que sous réserve de trouver un objet ayant une propriété susceptible de consommer toute la détension d'arrivée compte tenu du domaine dans lequel on se place. On retrouve à ce niveau le même procédé de négociation sémantique que celui appliqué au sujet.

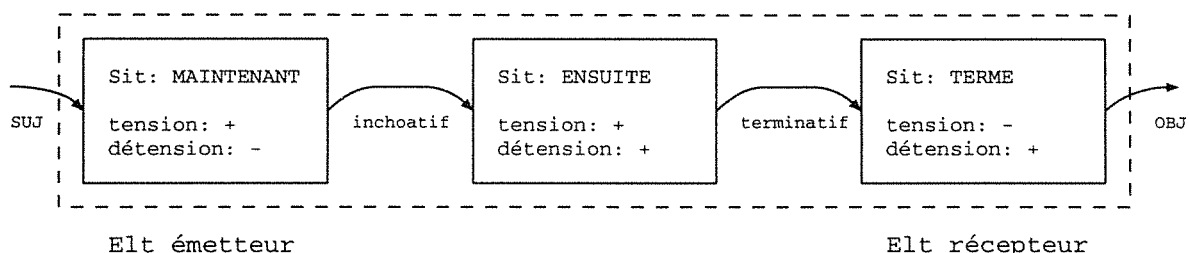


FIG. 6.9 - Graphe d'un déroulement avec verbe transitif

Dans ce cas, la borne fixe une limite à la détension, qui doit de ce fait se trouver actualisée. On obtiendrait une interprétation identique en employant un pronom, un groupe nominal comportant un article défini, anaphorique ou déictique, tel que dans :

(6.13) Imprime le.

(6.14) Imprime le fichier texte⁶.

(6.15) Imprime le fichier en cours d'édition.

5. Lorsqu'il y a ancrage avec la situation. On peut citer des exemples où l'on est amené à faire une interprétation générique avec un démonstratif ([Kleiber, 1992] d'après [Gaiffe, 1992] (p. 99)) :

(6.11) Les italiens exportent cette voiture depuis deux mois.

exemple dans lequel les propriétés spécifiques de l'objet (sa plaque d'immatriculation, sa couleur, ...) ne sont pas pertinentes par rapport au procès indiqué par le verbe (l'exportation).

6. avec un fichier texte précédemment mentionné.

1. L'élaboration de schèmes de déroulement

Ensemble d'objets non borné : Si, en revanche, l'ensemble d'objets est dépourvu de bornes, par exemple, s'il est déterminé par un article indéfini pluriel, il n'y aura aucune limite imposée à la détension. De ce fait, il se produit un effet de réitération, la détension ne pouvant s'actualiser sur un ensemble non borné d'objets. Considérons en guise d'illustration l'exemple :

(6.16) Mange des sandwiches.

Cette activité peut se dérouler à n'importe quel endroit où se trouvent des sandwiches, tant que ce mets continuera à se rencontrer : il n'y a aucune borne imposée tant au niveau spatial que temporel, ce qui fait que les sandwiches mangés peuvent se dupliquer à l'infini.

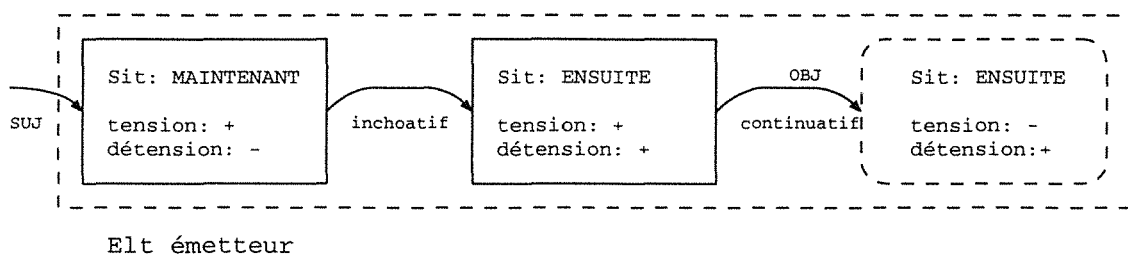


FIG. 6.10 - Graphe réduit d'un déroulement avec verbe transitif et objet non borné

On obtient alors un effet de réitération sur une succession de sandwiches mangés, contribuant à donner l'impression d'uniformité (homogénéité) relative à une activité, comme illustré par le graphe suivant (figure 6.11) :

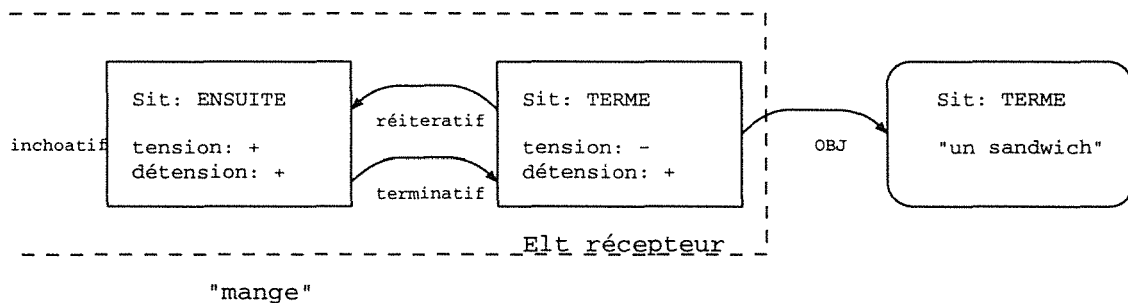


FIG. 6.11 - Graphe d'un déroulement avec verbe transitif et objet non borné

On peut alors compléter la chaîne causative entre potentiel et accompli qui figurait en 6.5 (figure 6.12) :

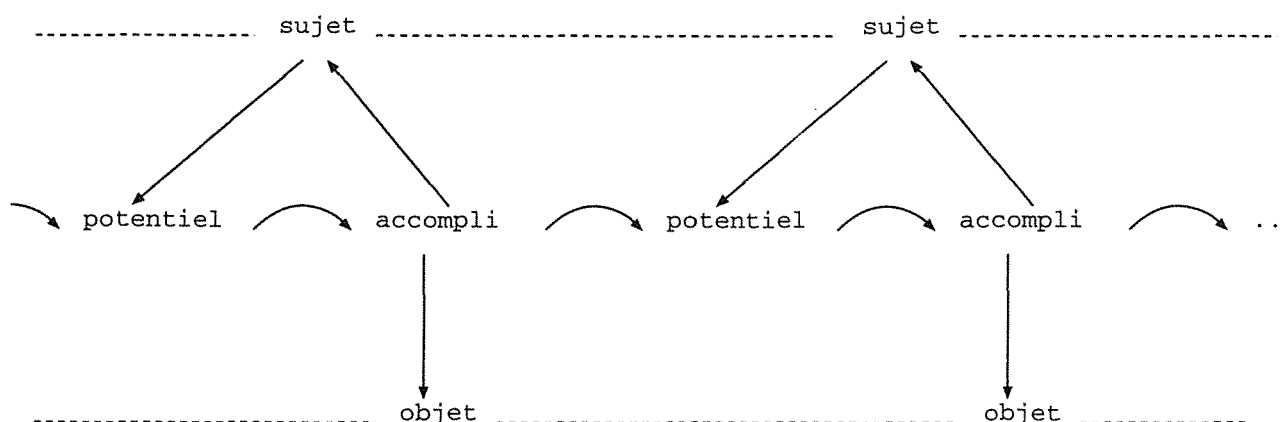


FIG. 6.12 - La dualité entre potentiel d'accomplissement et déjà accompli

Le complément d'objet direct n'est cependant pas le seul élément à influencer sur l'aspect. Les prépositions introduisant les éléments adverbiaux sont également susceptibles de modifier l'aspect.

1.3.4 La contribution des prépositions

Il ne saurait être question ici de détailler le fonctionnement de toutes les prépositions ou locutions prépositionnelles, mais de recouvrir la diversité de leur fonctionnement en détaillant un représentant typique de chaque classe. Nous allons ainsi commencer par les prépositions dont le caractère dynamique fait qu'elles sont utilisées pour classer les verbes, en mettant en évidence leurs propriétés aspectuelles. Nous examinerons ensuite un autre rôle actif des prépositions, consistant à poser des conditions d'activation d'interprétations.

Les prépositions d'Aktionsart

La locution prépositionnelle « jusqu'à ». Nous allons montrer que *jusqu'à* exprime une fonction de rupture de propriété⁷. En ce sens, *jusqu'à* se situe en dépendance par rapport à une propriété, elle-même en dépendance par rapport à une entité plus ou moins autonome. Afin de mieux isoler la fonction de *jusqu'à*, nous n'employons pas ici d'impératif mais le présent, temps grammatical neutre, pour ne prendre en compte que la partie instructionnelle véhiculée par la préposition. Prenons pour commencer l'exemple de l'énoncé :

(6.17) Le train va jusqu'à Nancy.

Le fait d'*aller* n'est valide, en ce qui concerne le train, que tant que l'on n'a pas atteint Nancy (i.e. on se situe *avant*). Il y a donc une rupture de propriété qui se situe au niveau de Nancy. Nous pouvons effectuer la même analyse avec un verbe transitif :

(6.18) Mange le sandwich jusqu'au tiers.

7. Cette caractéristique est fort probablement inhérente au seul « jusque » ; toutefois nous nous situons au niveau de la locution, ne désirant pas rentrer à ce niveau dans des considérations d'ordre compositionnel.

1. L'élaboration de schèmes de déroulement

On a le même effet lorsque le prédicat est exprimé sous la forme d'un attributif :

(6.19) Le train est sans supplément jusqu'à Nancy⁸

Nous pouvons avancer l'idée que *jusqu'à* impose une rupture dans le contexte d'un prédicat. Ce prédicat n'est pas uniquement verbal, mais peut être adjectival, comme dans :

(6.21) Le train, rapide jusqu'à Nancy, est ensuite omnibus.

En ce qui concerne son intégration dans les schèmes de déroulement, nous sommes amenés à introduire un nouveau symbole (un triangle dont la pointe est dirigée vers le lieu sur lequel se traduit la rupture de propriété). Illustrons ceci avec un énoncé simple de commande à un robot :

(6.22) Marche jusqu'à cette balise.

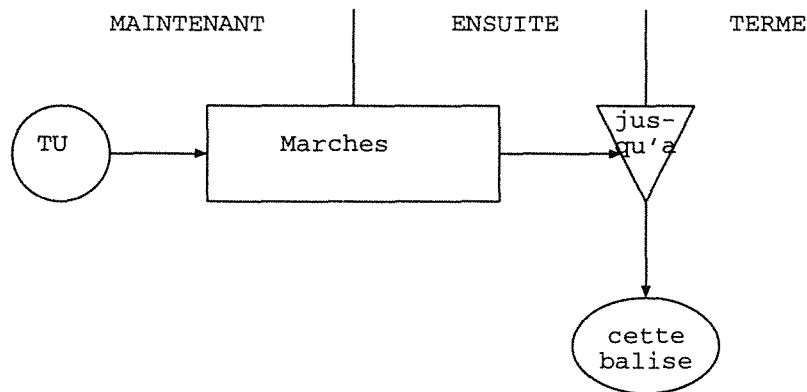


FIG. 6.13 - La rupture induite par « jusqu'à »

La préposition « en ». Nous avons déjà vu que la préposition « en » était utilisée dans les tests d'Aktionsart pour forcer une hétérogénéité dans le déroulement du procès, et éventuellement provoquer un conflit.

Si l'on reprend la caractérisation que fait Pottier de cette préposition [Pottier, 1985] (§150) :

« *En* évoque une intériorité recouverte, coïncidente⁹. »

on est amené à penser que *en* s'étend le plus possible dans le contexte prédicatif. D'où la spécification suivante :

En impose au prédicat une extension aspectuelle maximale (compte tenu de ses caractéristiques propres).

8. Nous retrouverions la même chose avec :

(6.20) Le train est à supplément à partir de Nancy.

9. On peut citer comme exemple : *une maison en feu, un canard en sucre.*

En guise d'illustration, considérons l'énoncé :

(6.23) Imprime ce fichier en recto-verso.

pour lequel la préposition *en* s'intégrerait comme illustré sur la figure 6.14.

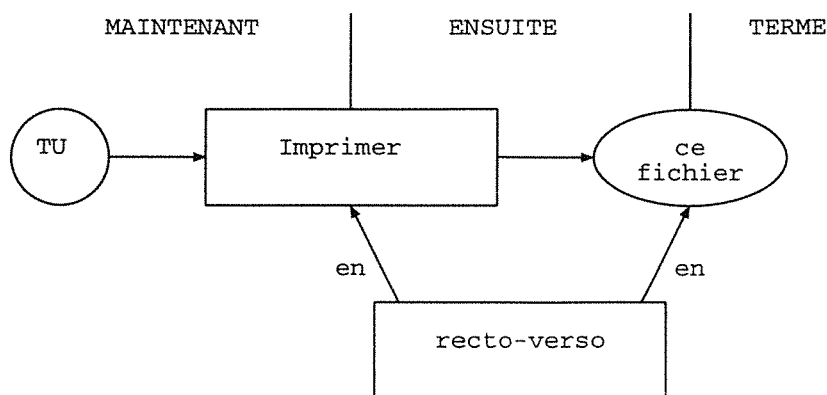


FIG. 6.14 - *L'intégration de en (a)*

Par extension maximale, nous entendons aussi loin aspectuellement que le permet le contexte prédicatif. Remarquons que l'*intérieurité recouverte*, *coïncidente* concerne également l'objet; aussi peut-on considérer *le document en recto-verso*. De même, dans des énoncés avec ellipse du complément d'objet tels que :

(6.24) Imprime en recto-verso.

la préposition *en* s'intégrerait comme illustré sur la figure 6.15 :

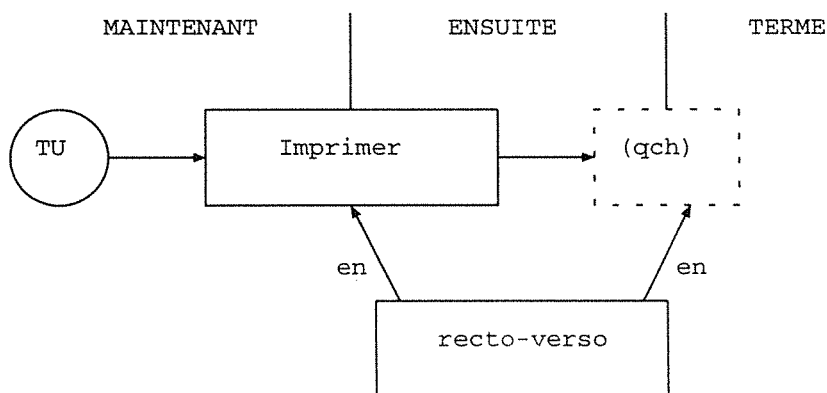


FIG. 6.15 - *L'intégration de en (b)*

Remarquons que le complément de *en* décrit toujours une propriété (en intension). Un cas intéressant est celui où il y a antagonie entre le complément de *en* et le contexte prédicatif, comme dans cet exemple emprunté à G. Sabah :

(6.25) Peins ce mur en salopette.

1. L'élaboration de schèmes de déroulement

Dans ce cas, le syntagme *en salopette* spécifie un attributif qui s'applique sur un élément situé en dehors de la phase de déroulement : le sujet autonome. La liaison avec l'objet (comme celle de la figure 6.14) échouerait dans ce cas, imposant ainsi un autre rattachement.

La préposition « pendant ». *Grosso modo*, la préposition « pendant » suppose que l'on puisse mettre en évidence une durée dans un procès. Un peu moins prosaïquement, elle isole une homogénéité lors du déroulement d'un procès. Comme nous l'avons vu, le fait que certains verbes puissent être rétifs à cet isolement fait de cette préposition un critère pour distinguer les achevements et les accomplissements. En effet, les achevements (ex : *gagner une course, atteindre le sommet, ...*) sont considérés comme atomiques, et donc non sécables. Par contre, la préposition peut s'appliquer sur un *accomplissement*, puisque ceux-ci englobent par nature le terme vers lequel culmine le procès. Nous retiendrons ainsi que l'intégration de la préposition *pendant* nécessite qu'une précondition soit vérifiée au niveau du schème de déroulement, à savoir qu'il doit déjà exister une situation *ensuite*. Par exemple, dans l'énoncé :

(6.26) Imprime ce fichier pendant la pause de midi.

on aboutirait au schème de déroulement suivant (figure 6.16) :

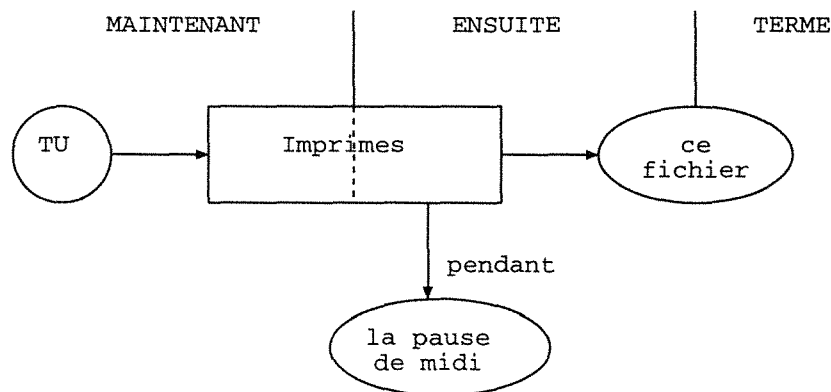


FIG. 6.16 - L'intégration de pendant

Mais en retour, la préposition *pendant* impose qu'il y ait préservation de la situation *ensuite*. Notamment, dans le cas des achevements, elle impose de casser l'atome relatif au terme. En cas de conflit avec la phase finale du verbe, on aura une interprétation itérative. Si enfin, cette interprétation itérative est elle-même impossible, l'énoncé sera ininterprétable. Là intervient de toute évidence un phénomène de négociation sémantique sur lequel nous aurons l'occasion de revenir. Prenons finalement le cas d'un verbe d'achèvement tel qu'*éteindre*.

(6.27) Tu éteins la lumière ?

Le fait que l'on puisse employer le présent témoigne d'une coexistence entre tension et détension, et donc d'une phase de déroulement. Mais on ne peut pas dire que la lumière

est à moitié éteinte ni à moitié allumée. Une analyse possible est alors qu'*éteins* englobe le mouvement du sujet qui se dirige vers l'interrupteur, phase à laquelle l'objet n'est pas impliqué¹⁰. Or, cette phase ne peut être isolée en faisant abstraction de son terme. C'est la raison pour laquelle on peut employer *en* et non *pendant* dans :

(6.28) Il a éteint la lumière en cinq secondes.

Prépositions généralistes

La préposition « à ». La préposition *à* introduit classiquement une destination, situé sur le plan du non achevé, du prospectif, comme c'est le cas pour :

(6.29) Envoie ce message à Laurent.

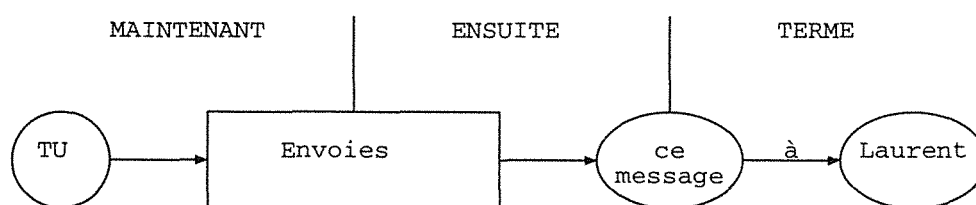
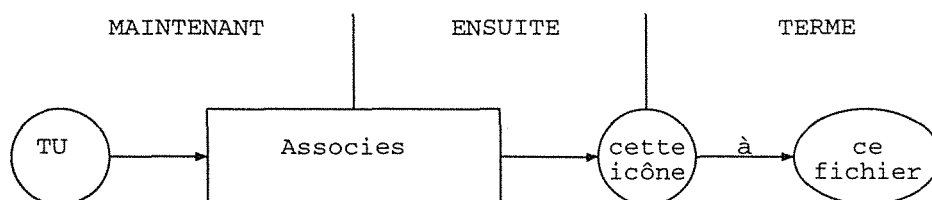


FIG. 6.17 - L'intégration de à

Ainsi, pour être interprété comme une destination, *à* n'exige rien de la part du donneur car il a tout ce qui lui faut en lui (la direction).

Nous avons vu que les critères de congruence entre verbes et prépositions posaient problème. Se plaçant à un niveau trop référentiel, ils occultent certains phénomènes plus fins. Le fonctionnement de ces prépositions est en fait tellement général, remontant jusqu'au plan structural, qu'il peut inclure des interprétations sémantiques radicalement opposées. Cela permet en outre d'expliquer des variations diachroniques de formes transitives à intransitives (ou vice versa) ou encore avec sa concurrente *de*. C'est pourquoi nous allons tâcher de chercher des critères davantage abstraits, en nous appuyant sur des considérations moins référentielles.

(6.30) Associe cette icône à ce fichier.



10. Cette souplesse de la langue montre à quel point est pertinente la réflexion dont fait part Cadiot :

« La langue n'est pas un calque de la réalité, mais une façon de la mettre en forme » [Cadiot, 1993] (p. 76).

1. L'élaboration de schèmes de déroulement

FIG. 6.18 - L'intégration de à

Pour justifier le fait que l'on rattache *ce fichier* à *cette icône* et non directement à *associes*, on peut avancer que la liaison entre l'icône et le fichier n'a de sens que pour l'icône « associée ». Même si la structure linéaire (et aspectuelle) de cet énoncé semble identique à celle de 6.29, il apparaît une grande différence au niveau de la symétrie du verbe par rapport à ses objets (directs et indirects).

(6.31) Associe cette icône et ce fichier.

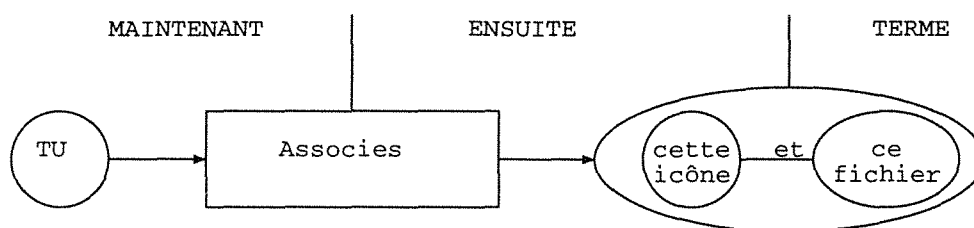


FIG. 6.19 - L'intégration de et

En effet, nous constatons qu'il est possible avec le verbe *associer* de regrouper les deux objets par une conjonction, pour former un seul objet complexe (alors que cela ne serait pas possible avec *envoyer* par exemple). Mais dans ce cas, nous ne rendons que l'état final en gommant toute la dynamique inhérente à la préposition *à*. Ainsi, en utilisant *et* dans « *associer A et B* », on spécifie que l'on peut « déplacer » tant A que B. Par contre, avec « *associer A à B* », on s'attend à ce que seul A soit modifié.

La préposition « de ». On retrouve un phénomène semblable avec la préposition *de*. La figure 6.20 donne une illustration de l'énoncé :

(6.32) Dissocie cette icône de ce fichier.

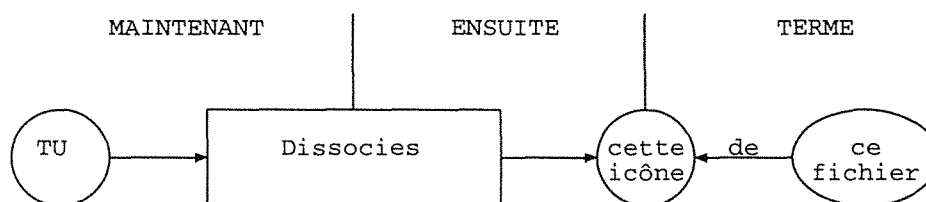


FIG. 6.20 - L'intégration de de

Ainsi, dans *dissocie A de B*, c'est A qui est le mobile, pour reprendre la terminologie de Langacker.

La préposition « avec ». La préposition *avec* comporte principalement une instruction de mise en parallèle. Le groupe nominal qui suit *avec* doit désigner une réalité appréhendable indépendamment du prédicat introducteur ; il doit être doué d'un certain coefficient d'existence dans un domaine propre (ou dans un « monde » autre que celui du prédicat, par exemple la réalité matérielle). Notamment, la préposition *avec* n'est pas appropriée à l'expression d'une relation intrinsèquement programmée entre le prédicat et le complément. Enfin, par comparaison avec la préposition *de* que nous venons d'étudier, la préposition *avec*, employée en complément adverbial d'un verbe possédant un complément d'objet, affecte les propriétés suivantes :

- la transitivité : le complément introduit par *de* est affecté par l'action directe du verbe, pas le complément introduit par *avec* ;
- l'aspect verbal : derrière *de*, on met l'accent sur le résultat, derrière *avec*, sur la phase processuelle ;
- la structuration thématique : avec *de*, il n'y a pas de mise en relief alors qu'il y en a une avec *avec*.

D'où l'hypothèse (dans le domaine restreint qui nous concerne) que *avec* introduit un processus secondaire, qui implique une certaine autonomie de l'objet de ce processus (ou détachement). En revanche, le fait pour un objet d'être lié à un processus primaire implique une certaine dépendance. Par conséquent, *avec* oblige à distribuer le prédicat. En guise d'illustration, considérons l'exemple :

(6.33) Imprime ce fichier avec un entête.

L'impression de l'entête est une sous-tâche de l'impression plus générale du fichier complet¹¹.

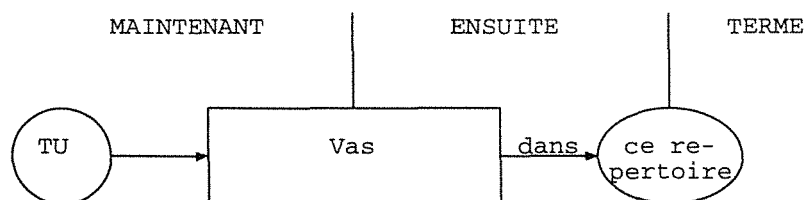
Prépositions spatiales statiques

La préposition « dans ». Pottier [1985] compare le fonctionnement de la préposition *dans* avec celui que nous avons déjà mentionné pour la préposition *en*¹² :

« *Dans* évoque une situation à l'intérieur d'une enveloppe distincte. » (§150)

(6.34) Va dans ce répertoire.

pour lequel la préposition *dans* s'intégrerait comme illustré par la figure 6.21.



11. À comparer avec l'impression d'une lettre commerciale avec un entête représentant un logo.

12. Rappelons que Pottier définissait *en* comme une intériorité recouverte coïncidente.

FIG. 6.21 - L'intégration de dans

Il se crée ici une rupture du déplacement dont *le répertoire* constitue une borne. En effet, à l'état initial, on a $\neg in(\text{Toi}, \text{Répertoire})$ alors qu'à l'état final, on a $in(\text{Toi}, \text{Répertoire})$. Cette rupture peut s'expliquer par une induction de dynamisme du verbe « aller » sur la préposition « dans ». C'est le même phénomène que l'on retrouve en anglais, par la préposition *in* qui se transforme en *into* dans le contexte des verbes de mouvement (tels que *to go*).

Remarquons que certains accepteurs sont beaucoup plus difficiles que d'autres à contenir. Nous avons vu que la préposition *à* renferme une instruction d'interprétation en tant que destination. Mais pour que la préposition *dans* puisse être interprétée comme une destination, elle doit avoir un donneur qui lui propose un contexte prédicatif impliquant un changement de lieu de référence (tel que dans l'exemple 6.35), et non un changement de position à l'intérieur du référentiel (tel que dans l'exemple 6.36). Dans ce dernier cas, la préposition porte sur la phrase entière.

(6.35) Laurent va dans le parc de la Pépinière. *vs.*

(6.36) Laurent court dans le parc de la Pépinière.

1.4 La modélisation de l'interaction contextuelle : l'intervention des promoteurs

Arrivés à ce point, nous avons établi que l'interprétation de l'énoncé était un processus dynamique et contextuel. La dynamique de l'interprétation est rendue par une partie instructionnelle caractérisant chaque nœud du graphe. Cette partie instructionnelle associée à un nœud, nous l'appellerons le *promoteur* du nœud.

Or, toutes ces instructions ne sauraient entrer en jeu simultanément ; certains promoteurs ne peuvent se mettre en veille que si la portion de graphe à laquelle ils sont associés présente un ensemble de caractéristiques données. Il existe ainsi un ordonnancement partiel dans l'activation des promoteurs. Les arcs entre les nœuds ont justement pour fonction d'exprimer une relation partielle de précédence qui en rend compte (permettant ainsi d'effectuer un chaînage avant ou arrière).

1.4.1 Les différentes classes de promoteurs

Nous avons déjà mentionné quelques promoteurs – les éléments *émetteur*, *transmetteur* et *récepteur* – respectivement associés à chaque nœud caractérisant une phase du déroulement du procès – *initiale*, *médiane* et *finale*. Ces promoteurs constituent la classe des *promoteurs de phase*. Nous les appelons ainsi car ils imposent une nouvelle étape (une phase) dans le déroulement du procès, et donc une rupture de situation. Les autres promoteurs, qui n'introduisent pas de rupture de situation, nous les nommerons des *promoteurs de configuration*. La spécificité des promoteurs de phase va être de restructurer progressivement les nœuds du graphe de déroulement d'après les propriétés sémantiques qu'ils renferment, au fur et à mesure de leur connexion avec les autres nœuds du graphe.

1.4.2 L'interaction contextuelle entre les nœuds du graphe de déroulement

Nous avons montré que le verbe changeait de visage tout au long de la construction du graphe de déroulement. Durant toute cette transformation, la part de détension tend à augmenter alors que celle de tension tend à diminuer. Les exigences par rapport aux nœuds avoisinant vont ainsi varier au cours du temps, de même que la configuration du graphe, qui, elle aussi, va évoluer. Les figures 6.22, 6.23, et 6.24 en détaillent les principales étapes.

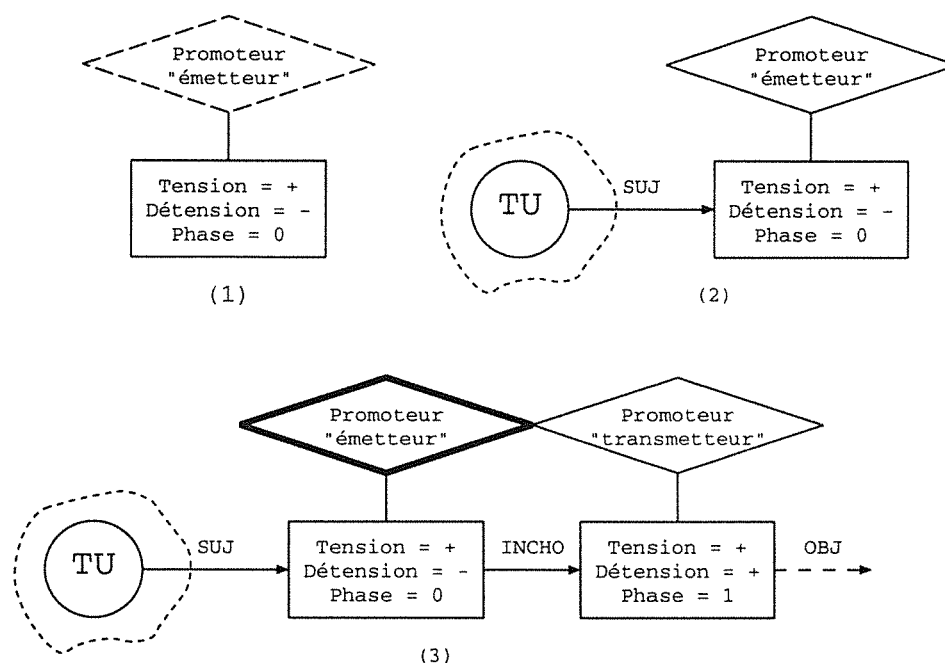


FIG. 6.22 - La partie commune initiale du fonctionnement des promoteurs

Au commencement de cette métamorphose, on trouve le verbe à l'infinitif, qui ne renferme aucun accompli mais uniquement un potentiel de déroulement. Il s'agit du verbe dans sa phase initiale. Dans le but d'activer le promoteur qui lui est associé, le verbe va alors chercher au niveau de son promoteur la liste des prérequis concernant les connexions des nœuds du graphe avec le nœud de phase (1). Au stade de la phase initiale, on aura pour les différents verbes la même liste de prérequis concernant le recrutement des nœuds candidats à connecter : il s'agit de trouver un nœud ayant la fonction sujet (correspondant à une position dans l'énoncé, ou déjà intégré dans le graphe).

À partir de là (2), il va se produire une négociation sémantique entre les promoteurs de chaque nœud ainsi connectés, afin de trouver un domaine d'entente, ce que nous entreverrons en 2.2.1. S'il y a terrain d'entente, le promoteur du nœud accepteur va pouvoir rentrer en activité (3). Dans le cas présent, il s'agit du nœud correspondant à la forme infinitive du verbe. Une fois activé, il va libérer la tension renfermée par le nœud qu'il contrôle. Cette libération de tension va se manifester par une application de détension, provoquant ainsi un passage *→accompli* \rightarrow *accompli* et donc une rupture de situation. Au niveau du graphe de déroulement, cela va se traduire par la création d'un nouveau

1. L'élaboration de schèmes de déroulement

nœud, correspondant à une phase médiane, avec son promoteur associé – le promoteur « transmetteur ». Pour poursuivre la construction du graphe, on a alors accès à la liste des prérequis sur les candidats à connecter, en vue de mettre en activité le promoteur. Le promoteur de verbe transitif (selon la définition d'inspiration « psychomécanique » que nous en avons donnée au chapitre précédent), réagira différemment, suivant que le nœud objet qui se connectera à lui sera borné ou non.

Dans le premier cas (4a), il n'y aura aucune rupture.

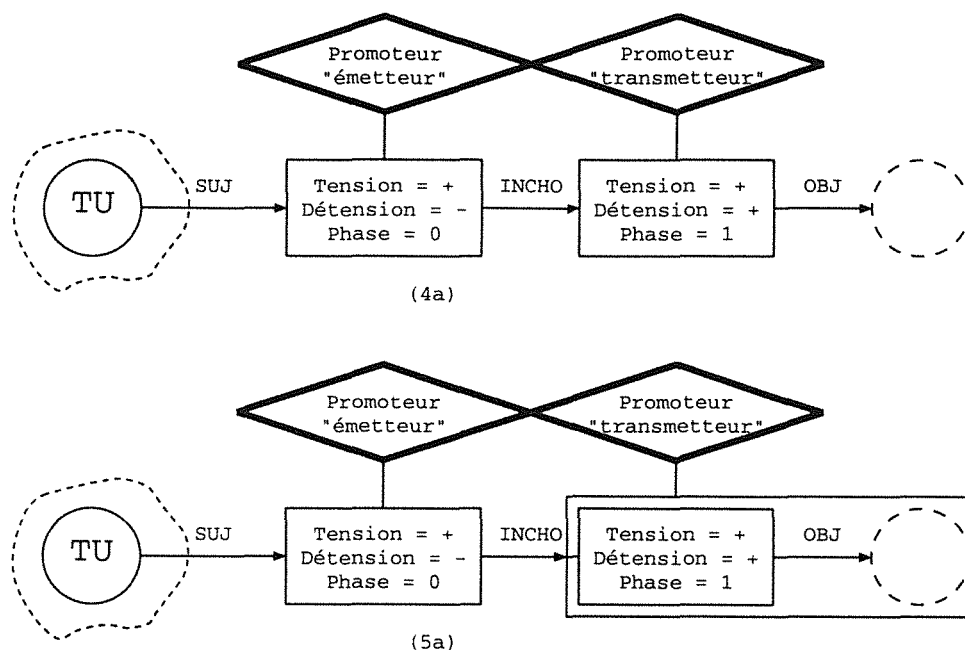


FIG. 6.23 - Réaction d'un promoteur face à un objet non borné

Le promoteur intégrera donc le nœud « objet » au nœud de phase médiane, pour en faire un nœud complexe (un sous-graphe) auquel il s'associera (5a). Dans le second cas (4b), lorsque l'objet peut borner la production de détente, cela provoque une rupture de situation.

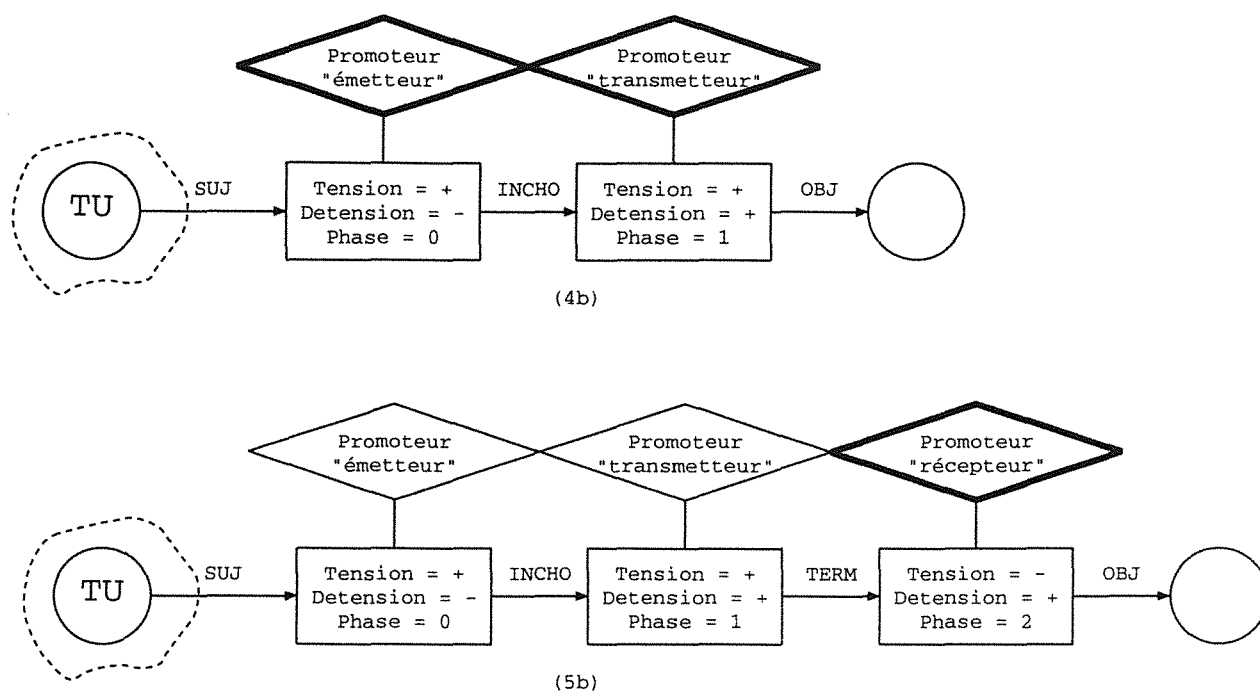


FIG. 6.24 - Réaction d'un promoteur face à un objet borné

Une particularité du fonctionnement des promoteurs est que l'activation de certains peut conduire à la désactivation d'autres, ce qui correspond en (5b) à la transformation de toute la tension en détension actualisée sur l'objet.

Nous rentrons ici dans une catégorie de profil aspectuel ne concernant qu'une partie des verbes transitifs pour une classe de contextes donnée. On pourrait aussi citer le cas de verbes intransitifs tels que *sortir* ou *arriver*, pour lequel la désactivation du promoteur *émetteur* après l'activation du promoteur *transmetteur* dépend du caractère borné du sujet. Par exemple, il y a un effet itératif créé dans la phrase (empruntée à [Borillo, 1991] p. 10) :

(6.37) Ils partent depuis un mois.

alors que l'on peut difficilement dire :

(6.38) ? Il part depuis un mois.

Un autre cas d'interaction est celui qu'il peut y avoir avec un promoteur rattaché à un nœud correspondant à un syntagme prépositionnel¹³.

Nous développerons plus en détail ces jeux d'interactions dans le chapitre suivant. Nous allons passer maintenant à une vision plus formelle du processus de contrôle des graphes.

13.

« Dans le cas où un verbe est suivi d'un adverbe (ou d'une préposition) introduisant un objet, il arrive souvent que l'on perçoive cet objet comme objet de la combinaison verbe plus adverbe ; c'est pour cela qu'il y a certaines hésitations dans leur emploi. » ([Jespersen, 1971], p.223)

1.4.3 Eléments de formalisation

Un profil aspectuel \mathcal{A} est une famille de couples (noeud de phase, promoteur de phase) à support I non nul.

$$A = ((Np_i, Pp_i))_{i \in I}$$

Pp_i désigne le promoteur de phase i (à 0 correspond la phase initiale).

Les noeuds : Soit \mathcal{S} l'ensemble des structures de dépendance qui peuvent être dérivées à partir de la grammaire de dépendance. Il existe une bijection entre une partie de \mathcal{S} et l'ensemble des noeuds \mathcal{N} , à partir duquel on définit $\mathcal{G} = \mathcal{P}(\mathcal{N} \times \mathcal{N})$, l'ensemble des graphes de déroulement.

Un noeud peut être atomique ou complexe, auquel cas il correspond à un sous-graphe. Il renferme les informations en provenance de l'énoncé qui vont servir de données de base aux promoteurs placés sous son contrôle.

On définira alors un attribut comme une fonction de \mathcal{N} vers un ensemble de valeurs. Soit Att_N l'ensemble des attributs sur \mathcal{N} :

$Att_N \supset \{\text{blocage-entree, blocage-sortie, situation, specification, forme, domaine} \circ \text{specification, quantite} \circ \text{specification, neg} \circ \text{specification, implicite} \circ \text{specification, fonction} \circ \text{forme, nom} \circ \text{forme, borne, tension, detension}\}$

Précisons la sémantique rattachée à chacun de ces attributs :

- *blocage-entree* et *blocage-sortie* indiquent qu'un noeud est saturé respectivement comme accepteur ou comme donneur ;
- *situation* est un entier incrémenté par un promoteur, correspondant à ce que nous avons ci-dessus défini par *maintenant, ensuite, terme* ;
- *specification* renferme les données sémantiques utilisables pour tester l'affinité entre des spécifications de noeuds connectés structurellement :
 - *domaine* indique le domaine sémantique concerné par le noeud ;
 - *neg* précise si une spécification est effectuée négativement ;
 - *implicite* précise si le noeud provient ou non d'une structure de l'énoncé ;
 - ...
- *forme* contient des données issues de l'analyse structurelle :
 - *nom* indique le nom du mot associé ;
 - *fonction* indique la fonction grammaticale occupée par la structure que représente le noeud ;
 - ...
- *borne* est spécifique aux noeuds de configuration, indiquant si un objet est borné ou non ;

- *tension* et *détension* sont spécifiques aux nœuds de phase, indiquant le caractère *pouvant être accompli* ou *étant déjà accompli* (trait binaire).

Comme nous allons le voir, l'utilisation de ces données par les promoteurs de phase va permettre de réorganiser l'agencement des nœuds qu'ils contrôlent les uns par rapport aux autres.

Les promoteurs de phase : On associe au promoteur un état prenant sa valeur dans l'ensemble :

$$\text{Stades} = \{\text{repos, veille, actif, épuisé}\}$$

L'évolution du promoteur sera alors modélisée par un automate à état fini

$$\Pi_e = (\text{Stades}, \text{repos}, \cdot_e)$$

sur un vocabulaire que nous préciserons ultérieurement, mais dont l'intersection avec

$$\{\text{att} \in \text{Att}_N \mid \text{att}(\mathcal{N})\}^*$$

est non vide.

Tant qu'un promoteur est actif, les actions qu'il est susceptible d'effectuer sont les suivantes :

$$\text{Actions} = \{\text{ajoute-noeud, connecte-noeud, retire-noeud, groupe-noeuds, degroupe-noeuds, associe-promoteur, invoque-promoteur, succes, echec}\}$$

consistant respectivement à :

- **ajoute-noeud** : ajouter un nœud dans la portion du graphe de déroulement que le promoteur contrôle ;
- **connecte-noeud** : connecter deux nœuds dans la portion du graphe de déroulement que le promoteur contrôle ;
- **retire-noeud** : supprimer un nœud dans la portion du graphe de déroulement que le promoteur contrôle ;
- **groupe-noeuds** : regrouper un ensemble de nœuds dans la portion du graphe de déroulement que le promoteur contrôle, et ainsi créer un sous-graphe ;
- **degroupe-noeuds** : dégroupier un ensemble de nœuds dans la portion du graphe de déroulement que le promoteur contrôle, et ainsi rétablir les connexions précédentes ;
- **associe-promoteur** : associer un promoteur à un nœud (atomique ou complexe) dans la portion du graphe de déroulement contrôlée ;
- **invoque-promoteur** : faire évoluer l'état d'un promoteur en destinant à son automate d'évolution un mot qu'il reconnaît.

– **succes et echec** : retournant le constat de l'exécution.

Dès qu'il est activé, le promoteur va donc exécuter un programme (qui est propre à la sémantique du mot qui correspond) sensible au contexte. Ce programme peut être modélisé par *un automate à états finis* $\Pi_a = (\Sigma, s_0, \cdot_a)$ auquel on associe *une table de décision* τ indiquant l'action à effectuer en fonction de l'état de l'automate Π_a et de la donnée lue¹⁴. Cette table est ainsi une application :

$$\tau_D : S \times D \rightarrow \text{Actions}$$

où $D = \{att \in Att_N | att(\mathcal{N})\}$ est l'ensemble des valeurs d'attributs, correspondant à l'ensemble des données acceptables par un programme d'instructions. L'exécution du programme correspondra à la donnée d'un couple (Π_a, τ_D) et d'une suite de D à support non nul. Remarquons que D^* constitue aussi le vocabulaire de l'automate Π_a .

En récapitulatif, un promoteur de phase est un 6-uplet $Pp_i = (\Pi_e, V, D, \Pi_a, A, \tau_D)$ où :

$\Pi_e = (S, repos, \cdot_e)$ est l'automate d'évolution du promoteur, de vocabulaire spécifique V , $S = \{repos, veille, actif, epuise\}$;

$D = \{att \in Att_N | att(\mathcal{N})\}$ est l'ensemble des valeurs correspondant aux attributs de N ;

$\Pi_a = (\Sigma, \sigma_0, \cdot_a)$ est l'automate des instructions du promoteur, de vocabulaire D^* , $\sigma_0 \in \Sigma$;

A est un ensemble de fonctions, correspondant aux actions que peut effectuer le promoteur dans l'état *actif* de S ;

$\tau_D : \Sigma \times D \rightarrow A$ est la table de décision associée.

Enfin, les promoteurs permettent de faire la liaison entre l'interprétation aspectuelle de l'énoncé et la construction de zones temporelles, opérant à un niveau référentiel. C'est seulement lorsque les données contextuelles lues par l'automate d'instructions sont conformes aux exigences du promoteur – ce qui revient à dire que la dernière action du programme est « succes » – que l'on peut envisager une construction locale de zones temporelles, par rapport à la portion du graphe que le promoteur contrôle.

2 Les promoteurs comme pivots de la construction de zones temporelles

Si l'on peut défendre l'idée d'une autonomie de la syntaxe par rapport à la sémantique, il est clair qu'il n'y a pas indépendance de l'une par rapport à l'autre. De manière un peu

14. Dans le cas que nous avons présenté ci-dessus, il pourrait s'agir pour le promoteur transmetteur du caractère borné ou non de l'objet connecté en sortie de son nœud de phase.

plus figurée, on peut considérer qu'il existe des passerelles entre des mondes différents mais pas pour autant isolés. Les schèmes de déroulement que nous avons introduits à la section précédente entendent précisément constituer l'une de ces passerelles (cf. figure 6.25) avec les schèmes de référence que nous avons introduits au chapitre 4.

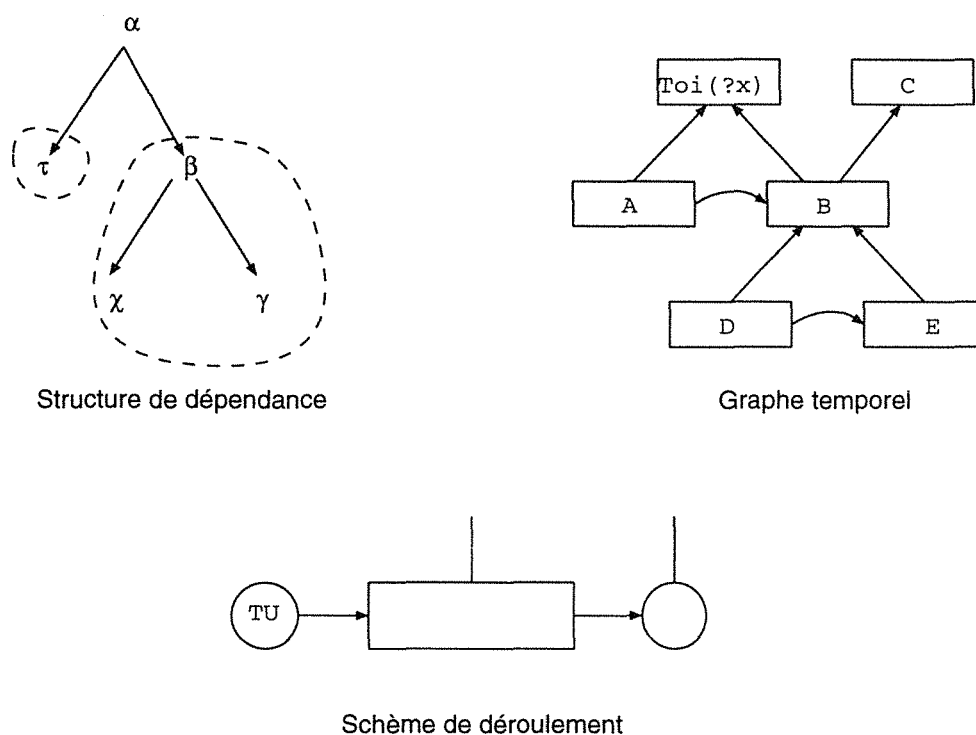


FIG. 6.25 - Les schèmes de déroulement comme représentation intermédiaire

Dans notre processus de référence aux actions, le niveau de représentation auquel nous souhaitons aboutir doit refléter les changements à faire intervenir sur des objets, comme spécifié dans les énoncés de commande. Nous avons défini (cf. chapitre 4 §2.2.3) les caractéristiques minimales d'un modèle temporel qui représente, sous forme de zones temporelles, ces objets et les changements qu'ils subissent. Il nous reste donc à montrer comment on peut construire ces zones temporelles, en utilisant la structure de l'énoncé et le schème de déroulement associé. Dans cette section, nous décrivons les mécanismes relatifs à la construction de ces zones temporelles.

2.1 De la dépendance structurelle à la dépendance sémantique

Un des points forts du modèle temporel adopté est de ne reposer que sur deux types de relations entre zones temporelles : l'inclusion et la précédence. La précédence se rencontrera dans les descriptions de passages entre situations (typiquement au niveau des liens *inchoatif* et *terminatif* des schèmes de déroulement). En revanche, les descriptions de situations vont être propices à la seule inclusion de zones temporelles les unes dans les autres, traduisant ainsi des propriétés à des degrés différents d'autonomie. Nous allons détailler ici les contraintes relatives à leur construction.

2.1.1 Critère d'inclusion-dépendance

On a pu parfois critiquer les grammaires de dépendance, plus ou moins issues des travaux de [Tesnière, 1969], selon lesquelles on voyait mal quels étaient les critères qui présidaient au choix du gouverneur et des gouvernés [Sabah, 1988, p. 150]. Pour répondre, du moins partiellement, à une telle question, il est nécessaire de s'interroger sur la finalité (au niveau du processus d'interprétation) des structures de dépendance produites. Si l'on considère par exemple l'état final EF que spécifie l'énoncé *Crée un répertoire public ...*, on doit notamment y trouver une zone temporelle ZT_2 , contenant la propriété $\text{public}(\text{?x})$, incluse dans une zone temporelle ZT_1 , contenant la propriété $\text{repertoire}(\text{?x})$. Or, dans la structure de dépendance (cf. figures 5.19 et 5.20) correspondante, on retrouve que l'adjectif *public* dépend du nom *répertoire*. On peut généraliser cette correspondance par la proposition suivante (cf. figure 2.1.1) :

Soit T un arbre de racine α spécifiant un objet E . Alors pour tout sous-arbre U_i de T tel que :

- la racine β_i de U_i est dominée structurellement par α ,
- U_i spécifie une propriété (simple ou complexe) P_i ,

on a P_i inclus temporellement dans E .

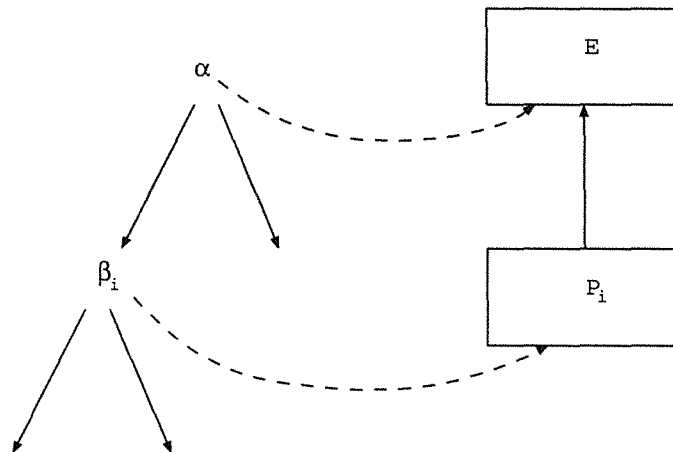


FIG. 6.26 - Critère d'inclusion-dépendance

L'idée à l'origine de ce critère est que :

Si le mot β est en dépendance par rapport au mot α , alors la spécification sémantique au niveau mineur de β contribue à la spécification sémantique au niveau majeur de α .

Il émerge alors le problème de la compositionnalité entre les propriétés ainsi spécifiées.

2.1.2 Le problème de la compositionnalité

Dans l'énoncé de commande, il peut apparaître des éléments de structure qui spécifient des paramètres intervenant au niveau de l'état résultant. Le problème que nous nous posons ici concerne la combinaison des propriétés ainsi spécifiées. Dans les problèmes de planification, les prédicats décrivant l'état final auquel on doit parvenir ne sont pas structurés les uns par rapport aux autres. Or, qu'en est-il au niveau des états finaux spécifiés par des énoncés de commande? Ce sont des considérations d'ordre sémantique qui vont permettre de répondre à cette question. Considérons par exemple simple de l'impression d'un fichier avec un en-tête; *avoir un en-tête* n'a alors de sens que par rapport à une impression. Dans la logique que nous avons introduite, cela se traduit par une inclusion de zones temporelles : ?x ne peut posséder un en-tête que s'il est sous forme imprimée. Si maintenant, on veut représenter les ruptures de propriété qui interviennent, la figure 6.27 pourra constituer une bonne représentation, mais non la figure 6.28.

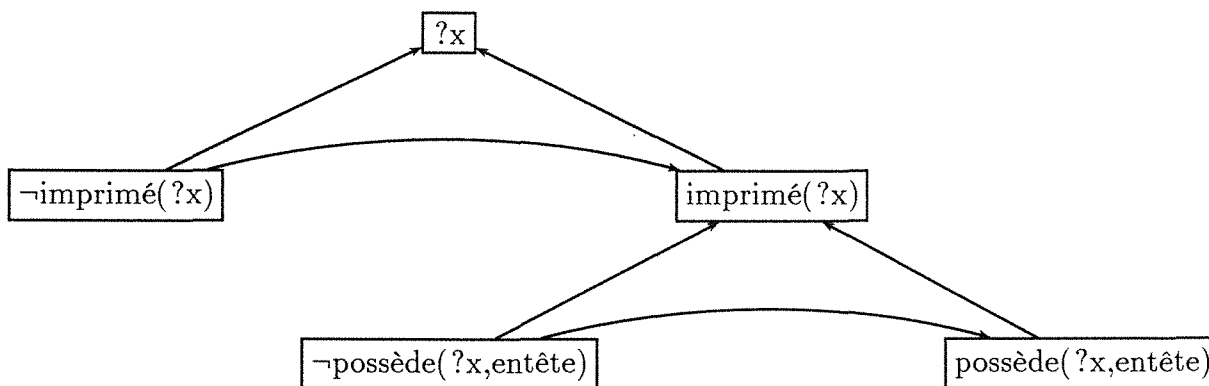


FIG. 6.27 - Une bonne représentation temporelle d'imprimer avec entête

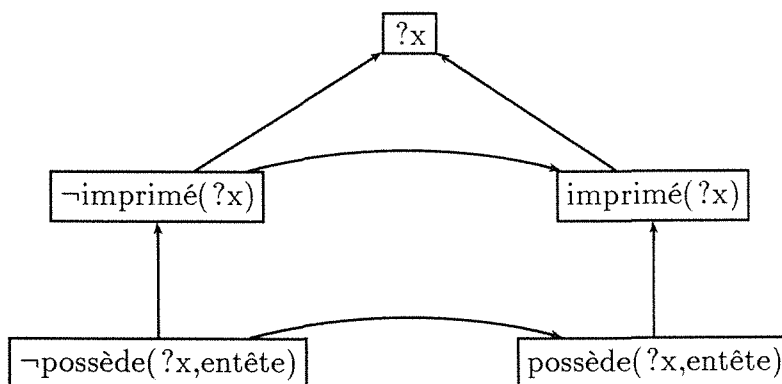


FIG. 6.28 - Une mauvaise représentation temporelle d'imprimer avec entête

On en arrive donc à la conclusion qu'il est *nécessaire* de structurer l'état résultant spécifié par l'énoncé. Cette nécessité s'avère même être un avantage au niveau de la

planification, diminuant le nombre de chemins en parallèle à envisager. Mais il reste à considérer l'état final non plus comme une collection de faits indépendants mais structurés par des propriétés d'autonomie différentes les unes des autres. La structuration d'un état en propriétés nous conduit naturellement à soulever la question de la dépendance sémantique.

2.1.3 La question de la dépendance sémantique

La question de la dépendance sémantique concerne les propriétés sémantiques partagées par deux éléments susceptibles d'être reliés entre eux. Il s'agit par exemple de déterminer s'il y a affinité entre *peint* et *en rouge*). La figure 6.29 illustre différents types de combinaisons que l'on est susceptible d'obtenir, suivant qu'il y a dépendance sémantique ou non entre des anciennes propriétés et de nouvelles propriétés à ajouter. Nous partons de l'hypothèse que :

- A est une zone temporelle représentant un objet ;
- B est une zone temporelle incluse dans A, représentant une propriété de A ;
- C est une zone temporelle à insérer représentant une propriété de A.

Dans le cas (a), il n'y a aucune propriété partagée entre B et C que l'on puisse inférer. Dans le cas (b), les propriétés impliquées dans C le sont également dans B. C est par conséquent inclus temporellement dans B. À la différence des cas (a) et (b), il se produit dans les cas (c) et (d) une rupture temporelle, manifestée par une relation de précédence, car il existe des propriétés impliquées dans C qui ne le sont pas dans B. C'est par exemple le cas d'un objet dont on savait qu'il était vert, et dont l'on apprend qu'il vient d'être peint¹⁵. La couleur a alors nécessairement évolué ; c'est ce qu'illustre le cas (c). Plus général est le cas (d) où il y a seulement partage des propriétés inférables.

Il devient alors possible d'expliquer pourquoi l'on peut dire 6.39 et non 6.40.

(6.39) Imprime ce texte en deux colonnes.

(6.40) * Imprime cette image en deux colonnes.

L'explication est que *colonnes* fait partie des attributs des documents de type *texte*, et non des documents de type *image*. Entrer dans des considérations de cet ordre, qui font par ailleurs l'objet actuel d'une thèse en préparation dans notre équipe, nous éloignerait de notre problème de départ. Nous retiendrons, en ce qui nous concerne plus directement, que l'ordre dans lequel on applique les propriétés sur un objet est d'une importance majeure. C'est sans perdre de vue cette conclusion que nous allons nous intéresser aux spécifications sémantiques, déterminantes à ce sujet.

2.2 L'influence des spécifications sémantiques

Les connaissances rattachées à un promoteur ne se suffisent généralement pas à elles-mêmes : il faut recourir à un contexte, et donc interagir avec les promoteurs des noeuds de configuration associés. Cela suppose donc un domaine sémantique commun.

15. On a même envie de dire *repeint*, reflétant la rupture temporelle.

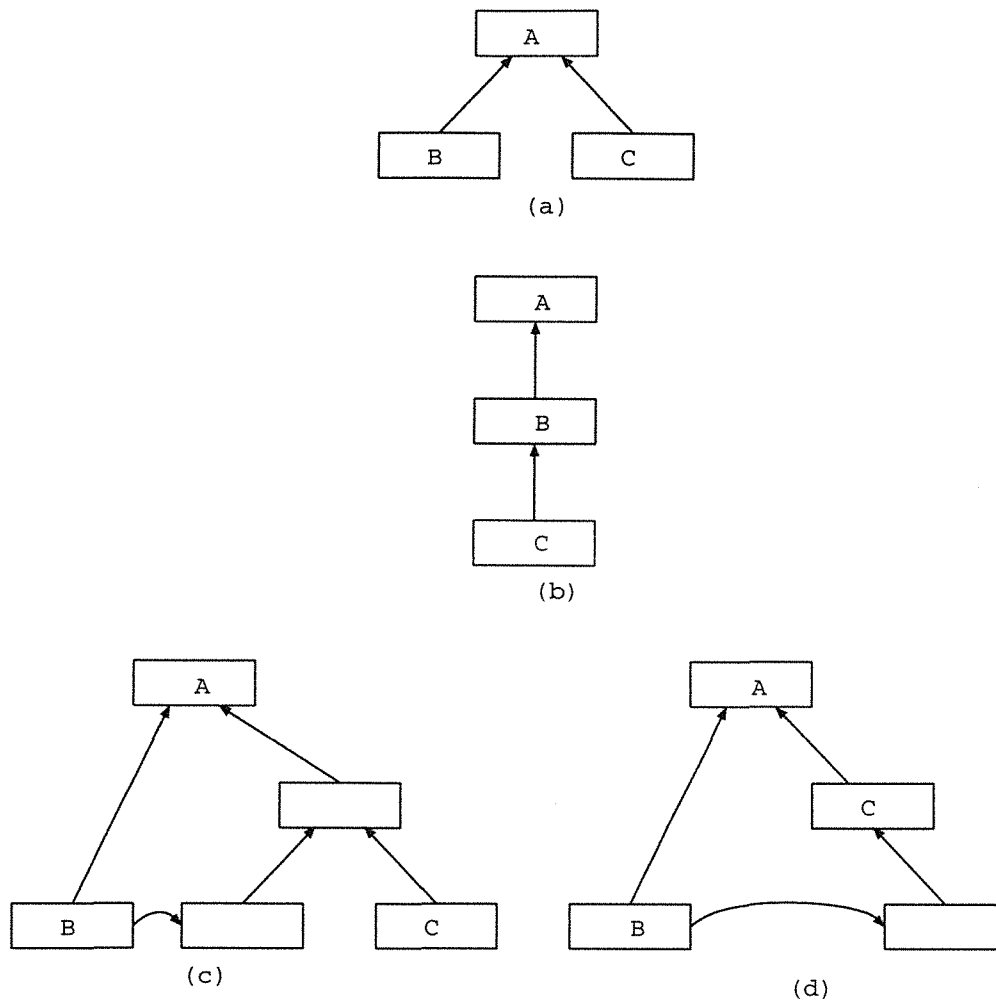


FIG. 6.29 - Les quatre types de combinaisons de propriétés de C par rapport à B

2.2.1 L'influence des domaines sémantiques

La bipartition (*acquis*, \neg *acquis*) évoluant le long de la chaîne interprétative, cela suppose que l'on oriente les domaines sémantiques qui sont successivement traversés. Comme ils ne sont pas orientés *a priori*, la question qui se pose alors est de savoir comment les divers espaces-domaines acceptent d'être orientés. *Orienter un domaine* revient à définir dans quel sens il évolue et selon quelle composante. L'évolution se traduira par une bipartition (*acquis*, \neg *acquis*) suivant une composante *C* caractéristique du domaine (à valeur considérée comme positive¹⁶). Une orientation (\neg *C*, *C*) sera alors dite *positive* et, à l'opposé, une orientation (*C*, \neg *C*) sera dite *négative*.

16. La sémantique différentielle [Rastier, 1987; Rastier, 1991] définit des oppositions binaires signées. Ainsi seront considérés comme positifs par rapport à leur opposé négatif (selon une dimension culturelle plus ou moins marquée) : haut, tête, debout, devant, externe, visible, lumière, jour, chaud, sec, actif, grand, futur, vie, oui, ...

Prenons le cas d'un espace topologique : l'orientation suit la dipartition (lieu d'origine, lieu de destination). Les prépositions peuvent alors directement opérer sur le domaine ainsi orienté. La préposition *de* interprète son nom-régime comme un lieu d'origine considéré comme acquis¹⁷. Si l'on prend en revanche un autre domaine, cela peut modifier l'interprétation que l'on peut avoir. Considérons en guise d'illustration l'exemple des verbes converses *éloigner* et *rapprocher*. Le domaine auquel se rattachent ces verbes est celui de la distance. Si l'on choisit comme caractéristique : $Pr(X,Y)$ où Pr indique une proximité entre X et Y (prédicat de relation symétrique de même que la distance), le domaine sera orienté positivement pour *rapprocher* et négativement pour *éloigner*. On peut le vérifier dans les formules suivantes, où S représente l'entité désignée par le sujet du verbe et $[O1,O2]$ représentent deux entités désignées par un ou deux objets du verbe (objet primaire et secondaire, ou objet complexe composé de deux objets)¹⁸ :

"S éloigner $[O1,O2]$ " : $CAUSE(S, BECOME(\neg Pr(O1,O2)))$

"S rapprocher $[O1,O2]$ " : $CAUSE(S, BECOME(Pr(O1,O2)))$

La préposition *de* désigne l'objet d'origine qui constitue la visée du sujet. Par rapport à un domaine relatif à la distance, cela revient à se fixer un lieu ponctuel considéré comme origine dans le repère où il se place. La détermination de cette origine devra de ce fait constituer un préalable lors du processus interprétatif. On aboutit alors à un objet complexe orienté intérieurement. C'est la sémantique d'*éloigner* qui fait que l'on est amené à considérer les deux compléments d'objet par rapport à une dérivation d'un objet complexe. D'une manière plus générale, les domaines peuvent être liés au locuteur, qui impose alors sa visée.

Lorsque nous avons comparé sur le plan aspectuel *associer* et *envoyer* nous avons obtenu des schèmes de déroulement similaires. Or, à la différence du verbe *envoyer*, le verbe *associer* peut être symétrique. Cela va se traduire par une organisation différente des objets rentrant dans la construction des zones temporelles correspondantes. C'est ici qu'intervient le « cadre » sémantique du verbe.

2.2.2 Le cadre sémantique du verbe

Dans le chapitre 2, nous avons montré les problèmes qui résultaient d'une conception syntaxique de la valence du verbe (rappelons qu'il s'agit du nombre de compléments qu'il accepte). En introduisant des domaines sémantiques reliés les uns aux autres, caractérisés par des propriétés :

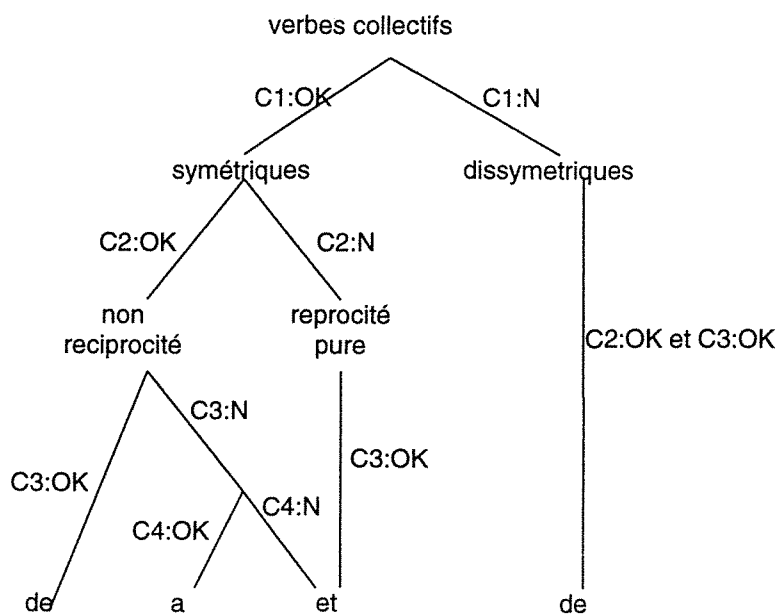
- simples, relatives à un seul objet ;
- complexes, partagées entre deux objets.

on s'ouvre la voie à une conception purement sémantique de la valence.

Un exemple intéressant est celui des verbes collectifs employés dans les formes *verbe-collectif A [de/à/et] B* (par exemple *rapprocher*, *dissocier*, ...). Le tableau suivant présente un arbre permettant d'en effectuer une typologie fondée sur leurs propriétés sémantiques.

17. Et en continuité avec le prédicat.

18. Je reprends ici (pour simplifier) le formalisme de Dowty [1979].



C1: objet simple au pluriel	C3: possibilité de donner le tout au départ
C2: possibilité d'individualiser un objet particulier	C4: présence d'une direction

FIG. 6.30 - Critères sémantiques des verbes collectifs

Comme l'illustre la figure 6.31, on a besoin dans tous les cas de spécifier un tout. La préposition *de* spécifie l'extraction d'un tout, qui constitue son point de départ. Avec la préposition *à*, ce tout constitue le point d'arrivée.

Forme	Avant	Après
Verbe A de B	(A) Tout (B)	(A) (B) Tout-A
Verbe A à B	(A) (B) Tout-A	(B) Tout

FIG. 6.31 - Ruptures temporelles avec verbes collectifs

C'est par exemple le fichier qui joue le rôle de tout dans l'exemple suivant :

(6.41) Dissocie cette icône de ce fichier.

On a donc affaire au départ à un fichier comportant un lien avec une icône, qu'il perd ensuite. Dans le cas du jonctif *et*, il y a un nouvel objet crée. Par exemple, *Additionne A à B* revient à l'affectation $B := B + A$, alors que dans *Additionne A et B*, on suppose la création d'un nouvel objet *résultat* tel que $résultat := A+B$.

2.3 Formalisation des règles de construction de zones temporelles

Le théorème d'inclusion-dépendance que nous avons exposé en 2.1.1 suppose que l'on puisse déterminer une zone temporelle correspondant à une propriété se rattachant par inclusion temporelle à une entité X . Les propriétés en question seront dites *simples* si elles se rattachent à cette seule entité X . Cela renferme typiquement le cas des adjectifs renvoyant à un attribut d'objet dans un domaine sémantique donné. Nous nous intéressons ici à des constructions plus complexes de zones temporelles : ce sont celles qui sont activées par les promoteurs de phase associés aux nœuds des schèmes de déroulement. Cela concerne essentiellement les relations induites par des prépositions ou les liens marquant une rupture de situation. Illustrons ci-dessous quelques exemples représentatifs de ces relations, permettant d'aboutir à des gabarits de zones temporelles qui leur soient propres.

2.3.1 Gabarit associé au lien *inchoatif*

Soit une zone temporelle $?p$ incluse dans une zone temporelle $?x$ vérifiant les faits suivants :

$$\left[\begin{array}{l} PRED(?x) \\ tension(?p)=+ \\ détension(?p)=- \end{array} \right]$$

L'*inchoatif* va spécifier le passage d'une détension nulle ($détension(?p)=-$) à une détension non nulle ($détension(?p)=+$). Cela correspondra dans le graphe temporel à une relation de précédence (faisant écho au *nexus* de Jespersen).

Dans le cas d'un verbe intransitif, cette détension ne va concerner que la zone temporelle correspondant au sujet. En revanche, pour les verbes transitifs, la détension va aussi s'appliquer sur une entité extérieure $?y$ qu'il faut créer dans le graphe temporel. On peut voir là une entité abstraite reflétant le *qch* d'une forme au participe présent *PRED-ant qch*. Nous n'introduisons cependant pas un identificateur *PRED-ant*, puisque cela est impliqué par $détension(?p)=+$. Le gabarit va donc provoquer le partage de la zone temporelle :

$$\left[\begin{array}{l} PRED(?x)(?y) \\ tension(?p)=+ \\ détension(?p)=+ \end{array} \right]$$

entre $?x$ et $?y$. Nous pouvons remarquer que cela introduit une arité augmentée de 1 au niveau de *PRED*. Il est intéressant de constater que l'on amorce de cette manière le procédé du calcul d'un éventuel référent en position objet, comme nous l'avons entrevu au chapitre 4.

En bref, le gabarit associé à un lien *inchoatif* permettra d'aboutir au graphe suivant, dans le cas d'un verbe transitif (cf. figure 6.32) :

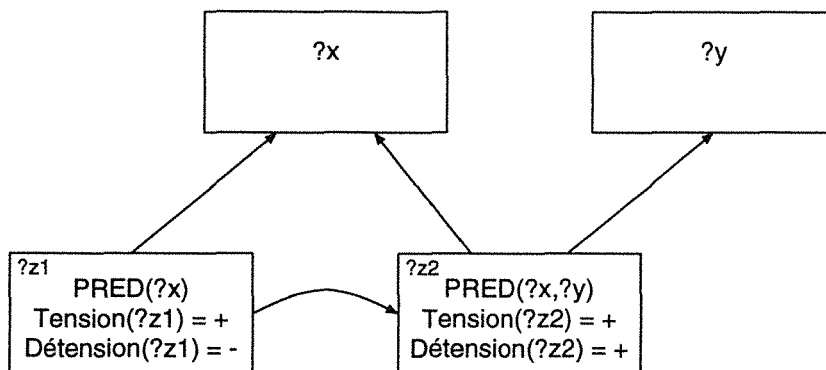


FIG. 6.32 - Gabarit de zones temporelles rattaché à inchoatif

2.3.2 Gabarit associé au lien *terminatif*

Les gabarits correspondant au lien *terminatif* se rencontreront avec les verbes transitifs (dans la conception sémantique que nous en avons donnée au chapitre précédent) ou avec les verbes demandant une direction cible tels que *aller*. Le *terminatif* va spécifier un passage d'une tension non nulle à une tension nulle. Ainsi, cela conduira à un stade de détension pure (ce qu'exprime une forme au participe passé *PRED-é* :

$$\left[\begin{array}{l} PRED(?x)(?y) \\ tension(?p)=+ \\ détension(?p)=+ \end{array} \right]$$

Le gabarit associé à un lien *terminatif* permettra d'aboutir au graphe suivant, dans le cas d'un verbe transitif (cf. figure 6.33) :

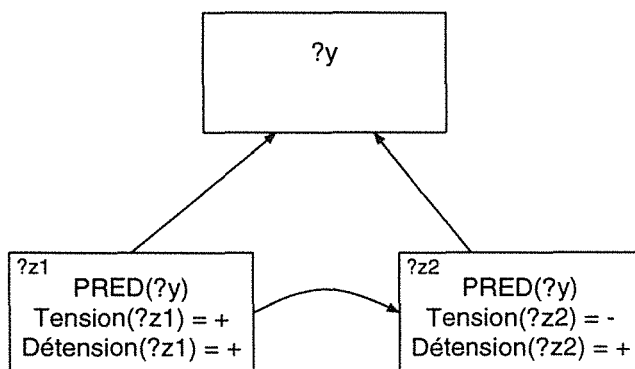


FIG. 6.33 - Gabarit de zones temporelles rattaché à terminatif

S'il s'agit d'un verbe collectif, l'objet ?y devra avoir la propriété d'être complexe (c'est-à-dire d'être inclus temporellement dans plusieurs objets simples).

2.3.3 Gabarits associés aux liens de localisation

Pour construire les zones temporelles spécifiées par des prépositions de localisation *dans*, *sur*, *sous*, *à côté de*, ... à partir d'une zone temporelle *X*, il est nécessaire d'introduire un objet supplémentaire correspondant au lieu *L* introduit par la préposition. Cela

correspond respectivement aux notions de cibles et de sites introduites par [Vandeloise, 1986]. [Kaufmann, 1991] suppose de même qu'il existe une composante partagée entre toutes les propositions spatiales, portant l'information qu'un objet X est localisé dans une région devant être définie par rapport à un objet Y . A cet effet, elle introduit la relation LOC entre une entité et une région, et la métavariable $REG(y)$ représentant un voisinage de l'objet Y :

$$\lambda x \lambda y (LOC(x, REG(y)))$$

De façon similaire, nous employons la relation de repérage \mathcal{E} introduite par Culioli [Dervillez-Bastuji, 1982; Caron, 1993]. $X \mathcal{E} L$ signifie que L est un élément localisateur de X . L a la forme $rel(Y)$ d'un objet Y (un lieu) sur lequel s'applique un opérateur sémantique rel (*dans, sur, sous, à côté de, ...*). On introduira en conséquence un prédicat d'arité 1, dont l'argument fait référence à une zone temporelle représentant un lieu.

On aboutit donc au gabarit général suivant, correspondant à une préposition de localisation spécifiant un opérateur rel (cf. figure 6.34) :

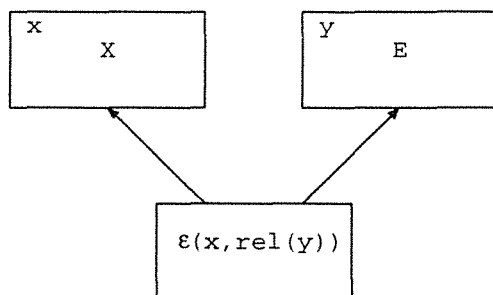


FIG. 6.34 - Gabarit de zones temporelles rattaché aux prépositions de localisation

2.3.4 Gabarit associé au lien « jusqu'à »

Pour construire des zones temporelles telles qu'elles sont spécifiées par la préposition « jusqu'à », il est nécessaire d'introduire deux arguments : un contexte prédicatif et un lieu cognitif. Précisons que ce contexte prédicatif peut aussi constituer en une propriété nominale, comme dans *le président jusqu'au mois prochain*.

Jusqua (contexte-predicatif : P) (lieu-cognitif : E)

Soit X une entité parmi celles sur lesquelles s'applique le contexte-predicatif P vérifiant $X \mathcal{E} E$.

La rupture de propriété que *jusqu'à* impose à P permet de déduire les deux implications suivantes :

1° (E n'est pas (encore) un élément localisateur de X) $\Rightarrow P(X)$ est vrai.

2° (E est devenu un élément localisateur de X) $\Rightarrow P(X)$ n'est plus vrai.

Or, on retrouve l'implication logique sous forme d'inclusion temporelle :

$$in(A, B) \Rightarrow A \subset B$$

si bien que l'on aboutit au gabarit général :

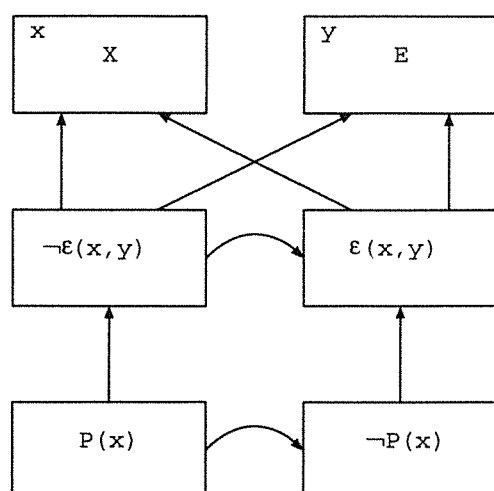


FIG. 6.35 - Gabarit de zones temporelles rattaché à jusqu'à

Si nous reprenons l'exemple 6.22 :

(6.42) Marche jusqu'à cette balise.

le gabarit précédent s'instancie pour donner le schéma de zones temporelles suivant :

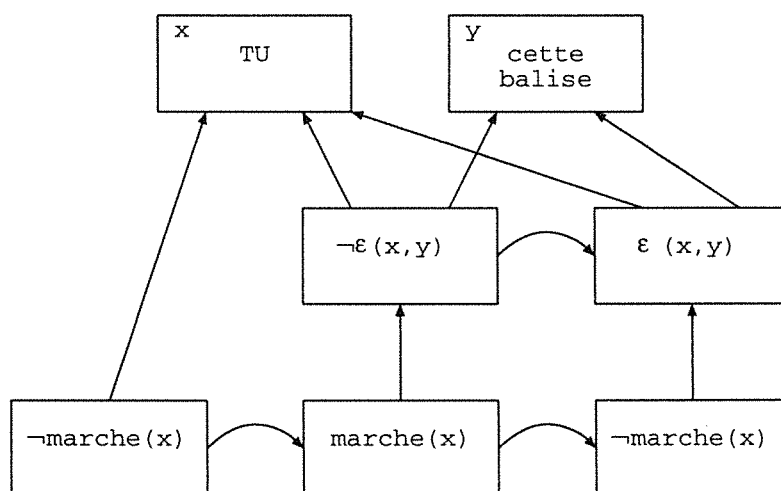


FIG. 6.36 - Zones temporelles associées à Marche jusqu'à cette balise

3 Conclusion

Voilà refermée la boucle de la référence aux actions. Nous avons montré la nécessité d'introduire une représentation intermédiaire - les schèmes de déroulement - entre une structure de dépendances de l'énoncé et le graphe temporel auquel nous souhaitons aboutir. Nous avons formalisé ces schèmes de déroulement sous forme d'un graphe et des éléments de contrôle (des promoteurs) associés aux noeuds représentant les phases du déroulement. L'utilité de ces promoteurs est de prendre en compte (à l'aide d'automates)

3. Conclusion

le contexte dynamique des portions de graphe qu'ils contrôlent, de restructurer ce graphe en fonction des propriétés aspectuelles mises en présence et d'alimenter le processus de référence. À des arcs du graphe de déroulement sont associés des gabarits rentrant dans la construction de zones temporelles. Cette construction est activée par un promoteur dont l'automate renvoie pour dernière instruction *succes*. Le chapitre suivant explicitera la mise en œuvre informatique de ce fonctionnement.

Chapitre 7

Choix de mise en œuvre

Sommaire

1	Implantation des principales composantes	202
1.1	Les principes architecturaux relatifs à notre démarche	202
1.2	Analyse structurale de l'énoncé	203
1.3	Schémas de déroulement	205
1.4	Processus inférentiels	210
1.5	Résolution de problèmes	213
2	Tests et résultats	214
2.1	Choix du cadre applicatif	214
2.2	Un exemple détaillé complet	216
3	Conclusion	221

Ce chapitre montre comment s'articulent les différentes composantes intervenant dans notre conception de la référence à l'action. Nous exposons à cet effet le principal algorithme sur lequel repose notre modèle. S'agissant de réaliser un système avec pour fin essentielle la validation de notre approche, notre choix au niveau de l'implémentation s'est porté sur le langage LISP (Common Lisp en l'occurrence) et sa couche objet standard (CLOS: Common Lisp Object System). En effet, la rapidité de prototypage que procure un tel langage, alliée à son aisance de manipulation des symboles, en font un outil de premier choix dans la validation de systèmes d'Intelligence Artificielle. Après avoir présenté les principales raisons qui nous ont poussés au choix d'un type particulier d'applications de commande, en l'occurrence la classe des systèmes d'exploitation, nous détaillerons le fonctionnement et l'interaction des différentes composantes de notre système à travers le développement d'un exemple d'énoncé de commande.

1 Implantation des principales composantes

1.1 Les principes architecturaux relatifs à notre démarche

Notre démarche repose sur une satisfaction de contraintes à différents niveaux, chaque module participant à une synergie d'ensemble. Les spécifications que chaque composante soumet à son niveau propre n'ont aucune raison a priori d'aller dans le même sens. Si l'on prend l'image de la corde, c'est comme si chacun tirait de son côté; la plupart du temps, il y en a un qui tire plus fort que l'autre, ce qui fait que l'autre cède. Mais il peut aussi arriver, lorsque chacun a la même force, que la corde se casse: on dit alors qu'il y a *clash*. En concevant une architecture reposant sur des contraintes à satisfaire, on évite d'instancier une stratégie interprétative¹. Cela permet lorsque l'on étend le système de ne pas remettre en cause les contraintes déjà présentes. De manière générale, cela évite d'imposer une stratégie qui s'avère vite caduque. Nous pouvons d'ores et déjà dresser un organigramme relatif à notre démarche (cf. figure 7.1).

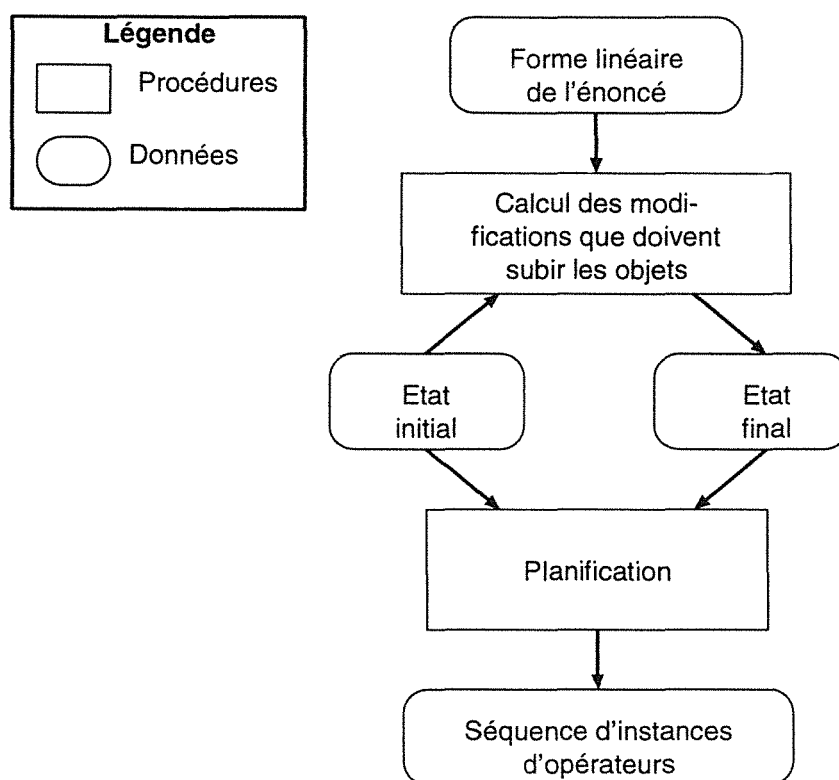


FIG. 7.1 - *Organigramme général*

Ainsi, en ce qui concerne la partie linguistique du traitement (calcul des modifications que doivent subir les objets), nous écartons toute architecture procédurale en série. Notre analyse se veut en effet guidée par les données issues de la combinaison des mots provenant de l'énoncé. De manière générale, rappelons que le paradigme de départ de la programmation orientée objet stipule qu'au niveau des procédures, ce qui se produit

1. Cette stratégie interprétative est en fait spécifiée dynamiquement par les mots [Rastier, 1987].

dépend de la classe des entités particulières qui sont impliquées. Un tel paradigme trouve notamment un écho direct dans diverses approches de la référence aux actions [Pouteau *et al.*, 1993], selon lesquels l'action à déclencher dépend du type des objets sur lesquels elle s'applique.

1.2 Analyse structurale de l'énoncé

Plutôt que de réécrire totalement un analyseur structural, notre choix s'est porté sur l'adaptation d'un analyseur du domaine public de type attribut-valeur suffisamment souple et générique pour autoriser diverses formes de grammaires. Il s'agit de l'analyseur *av-parser* de Mark Johnson [Johnson, 1988], écrit en Common LISP, incorporant un analyseur tabulaire et un mécanisme général d'unification.

1.2.1 Une analyse tabulaire

La principale différence d'approche entre une table et un arbre réside dans le fait que les arcs et les nœuds des structures sont échangés les uns par rapport aux autres ([Kaplan, 1979], [Sabah, 1989, p. 60]). Les arcs des arbres deviennent les nœuds des tables, correspondant alors aux séparateurs, et les nœuds des arbres deviennent les arcs (ou arêtes) des tables, correspondant ainsi aux mots ou groupes de mots (cf. figure 7.2). Un tel renversement de perspective apporte divers avantages, qui s'avèrent complémentaires de ceux provenant de la grammaire de dépendance que nous utilisons. Ainsi, la table code implicitement des relations de dominance alors qu'elles sont explicitées dans les grammaires de dépendance. En outre, les tables sont bien adaptées au traitement des forêts d'arbres syntagmatiques (c'est-à-dire un ensemble d'arbre disjoints). On a en effet vu qu'il n'y avait pas de catégorie *énoncé* proprement dite. On conçoit dès lors que l'on ne peut pas remonter à la racine (à l'axiome de la grammaire) et que l'on a ainsi affaire à une forêt d'arbres. La table permet de représenter une séquentialité entre les arbres sans imposer d'interconnexions. Enfin, dans une table, plusieurs arêtes peuvent relier deux nœuds, ce qui permet de représenter des ambiguïtés sans avoir à expliciter toutes les solutions en les développant combinatoirement [Constant, 1991]. L'avantage que cela procure est de toujours pouvoir renvoyer un ensemble d'arbres, sans nécessairement avoir pu les relier les uns aux autres.

1.2.2 Un mécanisme d'unification adapté aux grammaires de dépendance

Nous avons vu dans le chapitre précédent que chaque nœud d'une structure de dépendance représentait un mot ou une locution. Deux nœuds en dépendance seront reliés par un arc précisant la nature de la fonction de dépendance. En toute généralité, cette structure est un graphe, dès lors que l'on désire intégrer des mécanismes pour représenter la coordination. Dans ces différents cas, certaines structures sont mises en facteur, permettant de regrouper plusieurs arbres en un graphe. L'étude de cette factorisation conduisant à des problèmes épineux de portée, nous préférons nous restreindre aux cas où la structure est constituée d'un seul arbre de dépendance.

Dès lors que l'on représente une grammaire par un arbre de dépendance, il devient

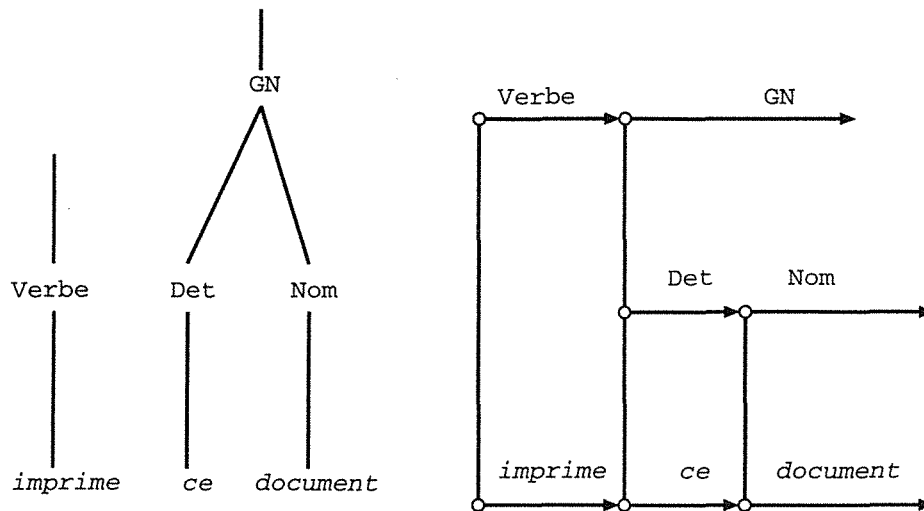


FIG. 7.2 - Comparatif entre représentation arborescente (syntagmatique) et tabulaire

pertinent d'utiliser un formalisme d'unification reposant sur des règles de production pour étiqueter les arcs. Cela s'effectue en rattachant des contraintes au niveau des termes intervenant dans les règles de production. Le programme de Johnson utilise un analyseur LALR permettant d'instancier un lecteur de grammaire correspondant à un formalisme lui-même décrit par la méta-grammaire fournie en paramètre (il peut s'agir aussi bien de grammaires d'application, de grammaires catégorielles, lexicales fonctionnelles, ou encore d'unification CAT-II, ...). Dans la figure qui suit, nous explicitons un extrait de notre propre grammaire de dépendance, qui utilise la notation suivante :

- *0 pour faire référence au terme « père » ;
- *i pour faire référence au terme « fils » d'ordre *i*.

```
E --> V SN SP:      label(*0) = ceiling,
                    dep1(*0) = *1,
                    fonct(dep1(*0)) = proposition,
                    dep1(dep1(*0)) = *2,
                    fonct(dep1(dep1(*0))) = comp-verbal,
                    dep2(dep1(*0)) = *3,
                    fonct(dep2(dep1(*0))) = comp-verbal,
                    indirect(dep2(dep1(*0))) = +.
```

```
E --> V SP:        label(*0) = ceiling,
                    dep1(*0) = *1,
                    fonct(dep1(*0)) = proposition,
                    dep1(dep1(*0)) = *2,
                    fonct(dep1(dep1(*0))) = comp-verbal,
                    indirect(dep1(dep1(*0))) = +.
```

```
SN1 --> Art N :    *0 = *2,
```

1. Implantation des principales composantes

```

                                dep1(*0) = *1,
                                fonct(dep1(*0)) = determinant.

SN1 --> Num N :                *0 = *2,
                                dep1(*0) = *1,
                                fonct(dep1(*0)) = quantifieur.

SN --> SN1 Adj :               pos-droite(*2) = +,
                                *0 = *1,
                                dep2(*0) = *2,
                                fonct(dep2(*0)) = comp-nom.

SN --> SN1 SP :                *0 = *1,
                                dep2(*0) = *2,
                                indirect(dep2(*0)) = +,
                                fonct(dep2(*0)) = comp-nom.

SN --> SN1      :               *0 = *1.

SP --> Prep SN :               *0 = *1,
                                dep1(*0) = *2,
                                fonct(dep1(*0)) = comp-prep.

SP --> Prep Adj :              *0 = *1,
                                dep1(*0) = *2,
                                fonct(dep1(*0)) = comp-prep.
```

Une différence de taille par rapport aux grammaires lexicales est que c'est ici le fils qui porte la fonction de dépendance par rapport au père, et non l'inverse. L'intérêt est de pouvoir se placer au niveau dominant du père, pour décider de la fonction à appliquer à un fils compte tenu de ses frères.

1.3 Schèmes de déroulement

1.3.1 La représentation des nœuds

Un nœud, quel qu'il soit, est défini par une collection de caractéristiques (*attribut, valeur*), une valeur pouvant être elle-même une caractéristique. Il comporte au minimum les caractéristiques suivantes (cf. chapitre 5, §1.4.3):

$$\text{Nœud de base} = \left[\begin{array}{l} \text{entrees} \\ \text{sorties} \\ \text{blocage-entree} \in \{V,F\} \\ \text{blocage-sortie} \in \{V,F\} \\ \text{situation} \in \{0,1,2,\dots\} \\ \text{specification} = \left[\begin{array}{l} \text{domaine} = \dots \\ \text{neg} \in \{V,F\} \\ \text{implicite} \in \{V,F\} \\ \dots \end{array} \right] \\ \text{forme} = \left[\begin{array}{l} \text{nom} = \dots \\ \text{fonction} = \dots \\ \dots \end{array} \right] \end{array} \right]$$

Pour des configurations, on a en plus :

$$\text{Nœud de configuration} = \text{Nœud de base} \cup \left[\begin{array}{l} \text{borne} \in \{V,F\} \\ \text{specification} = \left[\begin{array}{l} \text{identificateur} = \dots \\ \text{determination} = \dots \end{array} \right] \end{array} \right]$$

Et pour les nœuds de phase :

$$\text{Nœud de phase} = \text{Nœud de base} \cup \left[\begin{array}{l} \text{tension} \in \{V,F\} \\ \text{detension} \in \{V,F\} \end{array} \right]$$

Au niveau des promoteurs associés aux nœuds de phase, le jeu d'instructions contextuelles est représenté par une λ -expression. Il s'agit d'une façon commode de représenter l'automate d'instructions $\Pi_a = (\Sigma, \sigma_0, \cdot_a)$ (cf. chapitre 6, §1.4.3) dans un langage fonctionnel tel que Lisp.

1.3.2 Algorithme

L'algorithme suivant va permettre de superviser la construction du schème de déroulement où sont connectés les différents nœuds.

Conditions d'entrées

- *Un schéma initial* contenant des nœuds (non explicités dans l'énoncé) sur lesquels peut venir se greffer la construction du graphe de déroulement. Par exemple, dans le cas d'un énoncé à l'impératif, on a typiquement :

```
(make-noeud-configuration
 :situation 'maintenant
 :specification (make-objet
 :nom 'TU
 :ident '?toi
 :implicite T
 :designation 'absolu
 :quantite 1)
 :borne T)
```

FIG. 7.3 - *Le sujet implicite dans l'impératif*

1. Implantation des principales composantes

Cela peut également servir pour fournir le contexte prédicatif que nécessitent le traitement des ellipses sur des commandes.

- Une structure de dépendance *TopStructure* issue de l'énoncé correspondant à l'élément qui intègre les autres (un exemple typique est celui du verbe).

Table des principales variables

<i>Nom de variable</i>	<i>Description</i>
NdCourant	le nœud courant (init. la racine de l'arbre de dépendance)
PmtrCourant	le promoteur courant (initialement de base)
PmtrPrerequis	les prerequis relatifs au promoteur courant
lePrerequis	celui que l'on cherche à vérifier
num-prerequis	son numéro
StructuresDependantes	les structures filles de l'arbre de dépendance
DejaTraite	table de marquage des struct. incorporées ds le graphe
DejaEssaye	table de marquage des structures essayées
StrCandidate	la structure retenue
Ajout	indique si l'on doit effectuer un ajout dans le graphe
NdCandidat	le nœud candidat pour se connecter au nœud courant
LeLien	le type de lien entre les nœuds courant et candidat
LaDirection	la direction par rapport au nœud courant
lesPmtrVeille	liste des promoteurs en état de veille
LaissePourCompte	liste des structures non sollicitées

Algorithme TROMBONE(*Schema*, *TopStructure*)

1. **Création dans le schéma d'un nœud de phase *NdCourant*** relativement à *TopStructure*, et **association du promoteur de contrôle *PmtrCourant*** correspondant.

Soit *PmtrPrerequis* la liste des prérequis concernant le recrutement des candidats à l'activation de *PmtrCourant*. Initialiser *num-prerequis* à 0.

2. **Calcul de la liste *StructuresDépendantes*** des structures dominées immédiatement par *TopStructure*.

pour chaque $?s \in \text{StructuresDépendantes}$:

marquer $\text{DéjàTraité}(?s) \leftarrow \text{Faux}$ et $\text{DéjàEssayé}(?s) \leftarrow \text{Faux}^2$.

fin pour chaque

3. **Sélection d'un candidat** Soit *lePrerequis* le prérequis dans *PmtrPrerequis* de numéro *num-prerequis*.

Calcul de:

- liste-explicite: résultat de la recherche des structures $?s$ de *StructuresDépendantes* qui satisfont *lePrerequis* et non *DejaTraite(?s)*.

2. $\text{DejaTraite}(?s) \Rightarrow \text{DejaEssayé}(?s)$.

- liste-implicite : résultat de la recherche des nœuds du graphe *Schema* de statut implicite qui satisfont *lePrerequis*.
4. – **Éventualité 1 : Cas où il n’y a aucun candidat explicite ou implicite**
Si *num-prerequis* dépasse le dernier autorisé **alors** recours à un dialogue et resynchronisation **sinon**
- Incrémenter *num-prerequis*.
 - Aller en (3).
- fin si**
- **Éventualité 2 : Cas d’un candidat explicite dans la structure**
- Soit *StrCandidate* une structure prise dans *liste-explicite* telle que *DéjàEssayé(StrCandidate) = Faux*.
 - Marquer *DéjàEssayé(StrCandidate) ← Vrai*.
 - Marquer *Ajout ← Vrai*.
 - Transformer *StrCandidate* en nœud *NdCandidat*, correspondant à la spécification de l’entité ?E.
 - **S’il existe dans *Schema* un nœud dans *liste-implicite* correspondant à la même entité ?E **alors** effectuer une substitution par *NdCandidat* et marquer *Ajout ← Faux* (il n’y a donc pas finalement d’ajout).**
- fin si**
- **Éventualité 3 : Cas d’un candidat implicite dans le graphe**
- Soit *NdCandidat* un nœud pris dans *liste-implicite*.
 - Marquer *Ajout ← Faux*
- fin selon**
5. **Création d’un lien entre NdCandidat et NdCourant**
 Soit *leLien* le lien de direction *laDirection* (entrant ou sortant) issu des spécifications de *lePrerequis*.
Si *ajout* **alors** ajouter *NdCandidat* dans le graphe *Schema*, en effectuant la connexion à *NdCourant* par l’intermédiaire de *leLien* et *la Direction*
sinon créer un lien *leLien* de direction *laDirection* entre *NdCourant* et *NdCandidat*
fin si
6. **Test de propagation du lien *leLien* pour construire des zones temporelles**
Si le promoteur *PmtrCourant* parvient à s’activer (pas de conflits sémantiques entre les contraintes relatives à chaque domaine) **alors**
- Construire les zones temporelles à partir du gabarit associé à *leLien*.
 - **Si** *Ajout* **alors** marquer *DéjàTraité(?C) ← Vrai* **fin si**
- sinon**
- **Si** *Ajout* **alors** retirer *NdCandidat* du schéma,
sinon supprimer le lien *leLien* entre *NdCourant* et *NdCandidat*.
fin si

- Aller en (4) examiner un autre candidat pour le même prérequis.

fin si

7. Recherche si le déroulement a atteint son extension maximale³

Soit *lesPmtrEnVeille* la liste des promoteurs du graphe *Schema* qui sont à l'état *veille*.

Si *lesPmtrEnVeille* n'est pas vide, **alors**

- Soit *PmtrCourant* un élément de *lesPmtrEnVeille*.
- Soit *NdCourant* le nœud qu'il contrôle.
- Soit *PmtrPrerequis* la liste des prérequis concernant le recrutement des candidats à l'activation de *PmtrCourant*.
- Initialiser *num-prerequis* à 0.
- Aller en (3).

fin si

8. Terminaison : traitement des structures restantes en dépendance de *TopStructure*

Soit *LaissesPourCompte* la sous-liste des éléments de *StructuresDépendantes* vérifiant *DéjàTraité(?x) = Faux*⁴.

Si *LaissesPourCompte* n'est pas vide **alors**

- Soit *NdCourant* la transformation en nœud d'une structure de *LaissesPourCompte* (la première par exemple).
- Soit *PmtrCourant* le promoteur qui lui est associé.
- Soit *PmtrPrerequis* la liste des prérequis concernant le recrutement des candidats à l'activation de *PmtrCourant*.
- Initialiser *num-prerequis* à 0.
- Aller en (3).

fin si

1.3.3 L'interaction entre les diverses composantes

Notons qu'il n'est pas nécessaire de disposer dès le début de la liste entière des structures dominées immédiatement par *TopStructure*. La liste *StructuresDépendantes* décrite au point 2 peut alors correspondre au tampon d'un mécanisme de communication de type *producteur-consommateur* entre l'analyse tabulaire et le processus *trombone*.

Il serait en outre tout à fait envisageable d'interfacer cet algorithme avec une architecture qui repose sur un tableau noir, permettant à chaque composante de lire, d'écrire,

3. C'est-à-dire si aucun autre promoteur n'est prêt à prendre le relais.

4. Ce sont ceux qui n'ont pas encore été insérés dans le schéma, ne rentrant pas dans le cadre actanciel de *TopStructure*.

de modifier des données par rapport à son « terrain » à l'image du système de GLF programmé par Zweigenbaum [1991] qui utilisait une analyse tabulaire pour calculer des structures de constituants, parallèlement à un unificateur de structures fonctionnelles. Pour notre part, une telle architecture serait d'autant plus justifiée qu'il existe une véritable synergie entre les diverses composantes. Ainsi, dans notre cas, les spécifications véhiculées par les mots s'effectuent aux niveaux les plus généraux possibles (sémantico-pragmatique, lié au contexte) mais peuvent se raffiner au niveau même de la forme, pouvant ainsi valider ou invalider des hypothèses effectuées au niveau structural.

1.4 Processus inférentiels

On a tôt fait d'attribuer à la sémantique une caractéristique référentielle, faisant coïncider le signifié d'un signe linguistique à une réalité extérieure à la langue [Wittgenstein, 1961; Katz et Fodor, 1966], et une caractéristique différentielle, cherchant à opposer les classes de mots les unes aux autres au sein d'une même sur-classe [Greimas, 1966]. Mais c'est négliger là une caractéristique complémentaire de la sémantique constituée par son aspect inférentiel [Kayser, 1988; Rastier, 1987].

Dans notre système, l'inférence est utilisée à différents niveaux (lexical et pragmatique), en ayant toujours comme support un univers d'objets, renfermant des relations entre zones temporelles. Le premier niveau sert pour contourner la définition sémantique du lexique, alors que le second intègre des données pragmatiques liées à l'application.

1.4.1 Inférences liées à la sémantique lexicale

Prenons l'exemple d'un adjectif tel que *ouvert*, élément typiquement syncatégorématique (c'est-à-dire non suffisant à lui-même pour déterminer une catégorie sémantique). Si l'on veut pouvoir raisonner au niveau de l'application, il faut bien un moment effectuer le passage entre le domaine des identificateurs linguistiques et celui des identificateurs de l'application. La méthode naïve, à l'image de l'association « classique » *énoncé* → *fonction*, serait d'associer directement à l'adjectif une propriété. Or, une fenêtre n'est pas ouverte de la même façon qu'un fichier est ouvert. Il n'y a par conséquent aucune raison à ce que l'identificateur *ouvert* d'une fenêtre corresponde à l'identificateur *ouvert* d'un fichier (alors qu'il s'agit du même au niveau de la langue). Ainsi, il apparaît que le domaine sémantique, dans lequel se situe le nœud qui domine celui renfermant l'adjectif, est essentiel.

Sans recourir pour autant à une théorie du lexique, la méthode que nous proposons utilise un moyen simple pour tenir compte de ce domaine sémantique en s'appuyant sur les propriétés des objets. Il s'agit là du rôle du critère d'inclusion-dépendance que nous avons vu au chapitre précédent. Ci-dessous sont listées les notations que nous utilisons pour distinguer les spécifications de propriétés linguistiques (collectées au niveau structural) des propriétés sémantiques entre entités, telles qu'elles sont représentées dans les zones temporelles.

1. $Spec(e, p)$ désigne que p est la spécification linguistique d'une propriété de l'entité e .

1. Implantation des principales composantes

2. $Spec(e_1, e_2, r)$ désigne que r est la spécification linguistique d'une relation (orientée) entre les entités e_1 et e_2 .
3. $Type(e, T)$ désigne que l'entité e est de type T .
4. $Prop(e, P)$ désigne que P est une propriété de l'entité e .
5. $Rel(e_1, e_2, R)$ désigne que R est une relation (orientée) entre les entités e_1 et e_2 .

Seules les trois dernières seront représentées directement dans les zones temporelles :

- $Type(e, T)$: représenté par l'inclusion de $T(e)$ dans e ;
- $Prop(e, P)$: représenté par l'inclusion de $P(e)$ dans e ;
- $Rel(e_1, e_2, R)$: représenté par l'inclusion de $R(e_1, e_2)$ dans e_1 et e_2 .

La spécification peut se présenter le cas échéant sous forme de gabarits de zones temporelles à construire. Nous avons détaillé dans le chapitre précédent ceux correspondant à :

inchoatif(contexte-predicatif)

terminatif(contexte-predicatif)

rel(lieu)

jusqua(contexte-predicatif, lieu cognitif)

Autrement, on rendra compte d'un emploi possible de p dans plusieurs domaines à l'aide par exemple de clauses de Horn :

$$Prop(e, P_1) \leftarrow Type(e, T_1) \wedge Spec(e, p)$$

...

$$Prop(e, P_k) \leftarrow Type(e, T_k) \wedge Spec(e, p)$$

Lorsqu'aucune inférence n'a pu être effectuée, on pose par défaut :

$$Prop(e, P) \leftarrow Spec(e, P)$$

Un cas intéressant, plus général, est celui des verbes dits de production, qui introduisent de nouvelles entités qui ne sont pas différenciées linguistiquement. C'est par exemple le cas de *imprimer* qui implique l'existence d'un fichier x , mais introduit l'existence d'un support papier y , représentation de x . Pourtant, on dira tant de x que de y qu'ils sont imprimés ; mais cela ne correspondra pas à la même phase du déroulement. Nous constatons ainsi que deux entités distinctes peuvent correspondre au même complément d'objet direct du verbe.

$$Rel(y, x, SortiePapier) \leftarrow Type(x, fichier) \wedge Spec(x, imprime)$$

Précisons que l'identificateur *imprimé* est un raccourci d'écriture de :

$$\left[\begin{array}{l} nom : imprimer \\ tension : F \\ détente : V \end{array} \right]$$

1.4.2 Inférences d'ordre pragmatique

Un second niveau est celui de l'inférence pragmatique, qui opère directement au niveau du graphe de zones temporelles, en cherchant à le simplifier. Considérons par exemple les énoncés :

(7.1)

- (a) Laurent voit Bertrand dans le bureau de Fabrice. *vs.*
- (b) Laurent voit Bertrand depuis le bureau de Fabrice.

Dans l'énoncé (a), Bertrand se trouve nécessairement dans le bureau de Fabrice, alors que dans le (b), c'est Laurent qui s'y trouve nécessairement. Les inférences pragmatiques qui interviennent sont que si un individu est vu dans un lieu, alors il se trouve dans ce lieu.

$$Rel(x, L, in) \leftarrow Spec(x, L, dans) \wedge Spec(x, vu)$$

Précisons de même que l'identificateur *vu* est un raccourci d'écriture de :

$$\left[\begin{array}{l} nom : voir \\ tension : F \\ détente : V \end{array} \right]$$

Pour tenir compte de cet aspect, nous avons retenu au niveau de la réalisation pratique un simple moteur d'inférence à chaînage avant. Pour l'alimenter, il est nécessaire de recourir à des termes provenant d'une modélisation de l'application.

1.4.3 La prise en compte d'un modèle de l'application

Alors que *Spec* concerne la spécification d'une propriété au niveau sémantique, *Prop* et *Rel* se situent au niveau extra-linguistique. Comme nous le verrons plus en détail dans le §1.5, cela correspond à un langage formel de description d'un modèle de l'application. Les types des objets et leurs attributs, simples ou partagés, sont alors décrits. Formellement, cela correspond à définir un ensemble de termes, qui seront manipulés comme nous allons le voir par la composante de résolution de problèmes relative à l'application.

Un terme est une expression atomique décrite par la forme de Backus-Naur⁵ suivante :

```
<term> ::= (:not <term>)
<term> ::= (<predicate-name> <argument>*)
<argument> ::= <constant-name>
<argument> ::= <variable-name>
```

Ce que l'on souhaite, c'est qu'au niveau de *spec*, il n'y ait pas spécificité par rapport à l'application. Cela revient à revendiquer un niveau linguistique sémantique sur lequel interviennent des inférences permettant d'effectuer un recoupement avec le niveau extra-linguistique, destiné à la résolution de problèmes. L'étape suivante se situera donc au niveau extra-linguistique.

5. Nous utilisons des crochets pour entourer une clause optionnelle, l'étoile de Kleene (*) pour signifier « zero ou plus » et des chevrons pour faire référence à des noms.

1.5 Résolution de problèmes

Comme nous l'avons déjà élaboré dans le chapitre 3, nous intégrons la phase de raisonnement (purement extra-linguistique) sous la forme d'un planificateur. En reposant sur les termes que nous avons définis ci-dessous pour décrire l'application, nous empruntons à [Penberthy et Weld, 1993] le langage de description des actions qu'il utilise pour définir ses domaines de planification. Un domaine d'application sera donc constitué d'opérateurs, dont la forme de Backus-Naur est :

```
<operator> ::= (define (operator <operator-name>
                :parameters (<parameter>*)
                [:precondition <goal-description>]
                :effects (<effect-rule>*))

<parameter> ::= <variable-name>
<parameter> ::= <typed-var>
<typed-var> ::= (<parameter-type> <variable-name>)
```

La liste *parameters* correspond à une liste de variables sur lesquelles les règles opèrent (c'est-à-dire leurs arguments). La *precondition* est une description de but optionnelle devant être satisfaite avant que l'opérateur soit appliqué⁶. *Effects* recense les changements que l'opérateur impose sur l'état courant du monde.

Les descriptions de but sont utilisées pour spécifier les buts escomptés au niveau de la planification, mais aussi pour exprimer les préconditions pour un opérateur. Ils peuvent être décrits par une logique des prédicats du premier ordre, présentant la forme de Backus-Naur suivante :

```
<goal-description> ::= <term>
<goal-description> ::= (:and <goal-description>*)
<goal-description> ::= (:or <goal-description>*)
<goal-description> ::= (:not <goal-description>)
<goal-description> ::= (:forall (<typed-var>*) <goal-description>)
<goal-description> ::= (:exists (<typed-var>*) <goal-description>)
```

La définition d'effets conditionnels universellement quantifiés est également rendue possible pour décrire les opérateurs :

```
<effect-rule> ::= ([:forall (<parameter>*)]
                  [:when <goal-description>]
                  :effect <effect>)

<effect> ::= <term>
<effect> ::= (:and <term>*)
```

Nous employons le planificateur UCPOP [Penberthy et Weld, 1993] utilisant le langage de description des opérateurs explicité ci-dessus, qui fournit en résultat la séquence des opérateurs à exécuter pour rendre compte des actions auxquelles on réfère.

6. Si aucune précondition n'est spécifiée, cela signifie que l'opérateur est toujours applicable.

2 Tests et résultats

Même si l'architecture que nous proposons se veut indépendante du choix d'une application, une première nécessité est de se fixer un cadre applicatif, qui seul pourra donner une sémantique aux actions auxquelles réfèrera l'utilisateur. C'est seulement alors que nous pourrions évaluer les performances de la méthode de référence aux actions que nous proposons.

2.1 Choix du cadre applicatif

Notre but est d'être indépendant de l'application. La seule contrainte que nous imposons est que ce soit une application de type commande. Puisqu'il fallait malgré tout choisir une application, nous avons privilégié comme critères à la fois la validation de notre démarche (sur le plan théorique) et l'intérêt qu'il pouvait y avoir de recourir à un langage à consonance naturelle (sur le plan pratique). Notre choix s'est finalement porté sur une application informatique que possède tout ordinateur : un système d'exploitation.

Avant tout, ce sont des préoccupations d'ordre théorique et purement argumentatives qui nous ont conduits à choisir cette classe d'applications de commande plutôt qu'une autre :

- 1° Un shell ne reçoit que des commandes que l'on peut modéliser sous la forme (prédicat arguments), un argument pouvant représenter aussi bien un paramètre qu'un objet de l'application. Par rapport aux applications de commande de type *manipulation d'objets à l'écran*, cette classe d'applications apporte une plus grande richesse dans l'expression des verbes.
- 2° L'application se doit de posséder des fonctions (commandes) bien délimitées, de manière à bien pouvoir mettre en évidence ses points d'entrées déjà existant. Sur le plan illustratif, UNIX ou MS-DOS sont des systèmes suffisamment répandus pour pouvoir se faire une bonne représentation de l'application et de ses points d'entrées sans avoir à la décrire trop en détail. Tout lecteur informaticien peut ainsi se faire une idée d'une séquence d'actions élémentaires qui pourrait rendre compte d'un énoncé de commande⁷.
- 3° A des fins démonstratives, un moyen simple pour combiner des actions élémentaires devait être offert. Les shells présentent notamment la caractéristique de pouvoir créer eux-mêmes des enchaînements complexes de commandes (à base de tubes, de scripts de commandes).
- 4° Enfin, il est souhaitable de ne pas être dépendant de l'arbitraire des commandes liées à telle ou telle application. L'existence de différents types de systèmes d'exploitation réalisant les mêmes fonctions de base permet d'illustrer la généralité de notre approche. Par exemple, à la même commande :

(7.2) Imprime ce fichier en cinq exemplaires.

7. Toutefois, le risque pratique en s'adressant à des utilisateurs avertis pour tel ou tel système d'exploitation est que ceux-ci calquent leur manière de parler sur le format des commandes, sans chercher à exploiter la puissance d'expression d'un langage à consonance naturelle.

correspondent des séquences de commandes sans grand rapport les unes avec les autres, dans les quatre systèmes d'exploitation suivants⁸ :

(a) MS-DOS :

```
print rapport.ps /c:5
```

(b) VM/CMS :

```
CP SPOOL PRINTER  
COPY 5 PRINT RAPPORT PS A  
CP SPOOL PRINTER COPY 1
```

(c) VAX/VMS :

```
print/copies=5 rapport.ps
```

(d) Unix :

```
lpr -#5 rapport.ps
```

En outre, on peut trouver des arguments pour l'utilisation d'une telle interface. Sur le plan pratique, elle est surtout dédiée à des utilisateurs novices en systèmes d'exploitation, beaucoup plus enclins naturellement à utiliser un langage à consonance naturelle. Il existe toutefois des avantages à utiliser une interface de ce genre pour des utilisateurs moins occasionnels, qui peuvent être malgré tout déroutés par l'aspect cryptique des commandes. En effet, l'ordre et la nature des arguments ne suivant pratiquement aucune règle, cela peut entraîner grand nombre de confusions, d'une commande à une autre, voire d'un système d'exploitation à un autre. Pour illustrer ce dernier point, on peut envisager un exemple pratique dans le cadre de l'administration d'un réseau hétérogène [Xie *et al.*, 1989; Xie et Du, 1992]. Un intérêt pourrait alors être de rendre transparents les différents systèmes d'exploitation avec lesquels l'utilisateur communique, sans qu'il ait à se soucier de la syntaxe spécifique des commandes du système auquel il s'adresse (figure 7.4).

8. Exemple emprunté à [Manaris et Dominick, 1993].

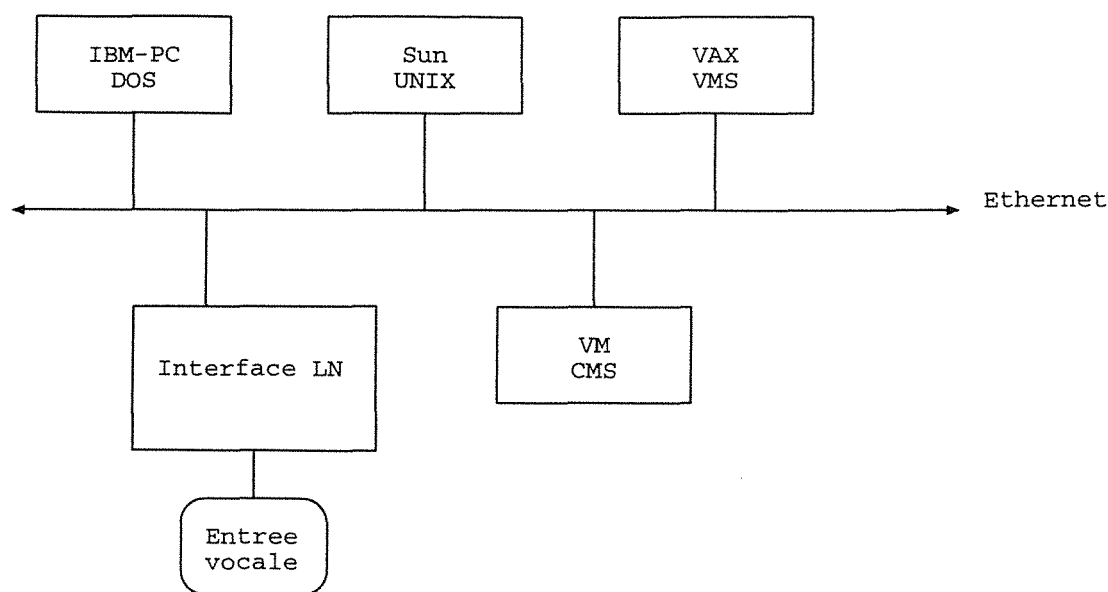


FIG. 7.4 - Administration d'un réseau hétérogène

Notre système entend traiter un jeu complet de commandes. Mais pour donner une vision globale des traitements, nous ne détaillerons qu'un seul exemple.

2.2 Un exemple détaillé complet

Considérons l'énoncé :

(7.3) Imprime ce document en deux colonnes.

Avec la grammaire dont on a présenté un fragment ci-dessus, l'analyse de cet énoncé amène à envisager deux tables (figures 7.5 et 7.6) :

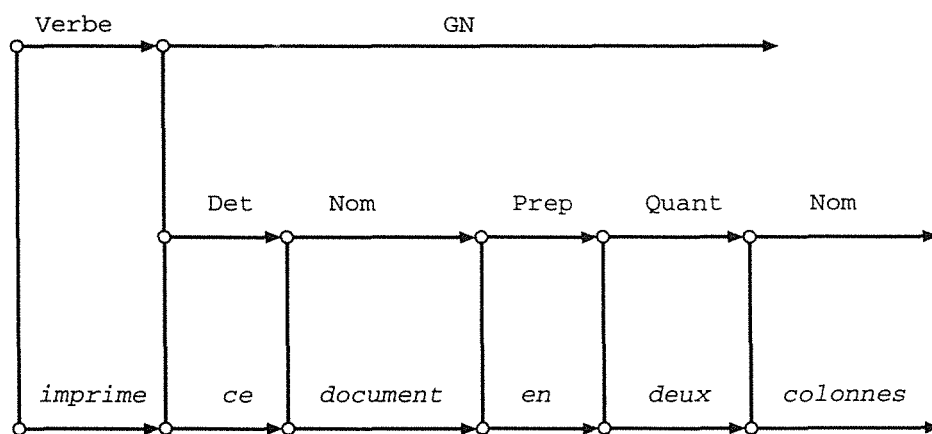


FIG. 7.5 - Table d'analyse (1)

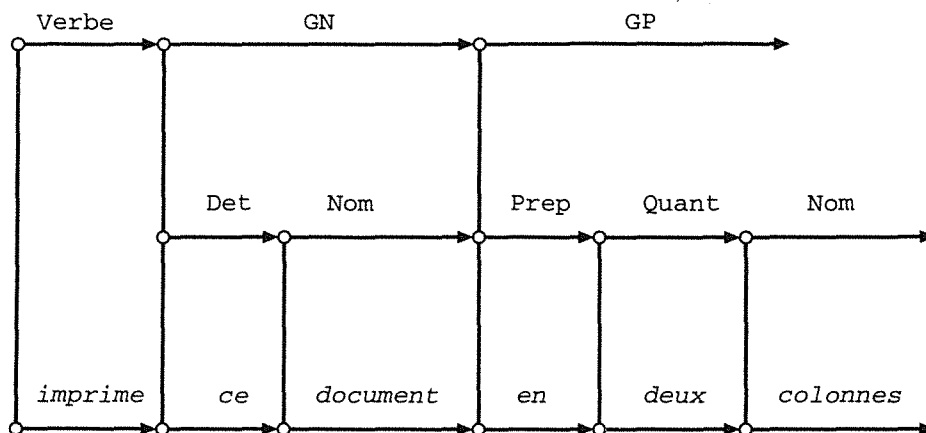


FIG. 7.6 - Table d'analyse (2)

A ces tables correspondront deux structures de dépendance :

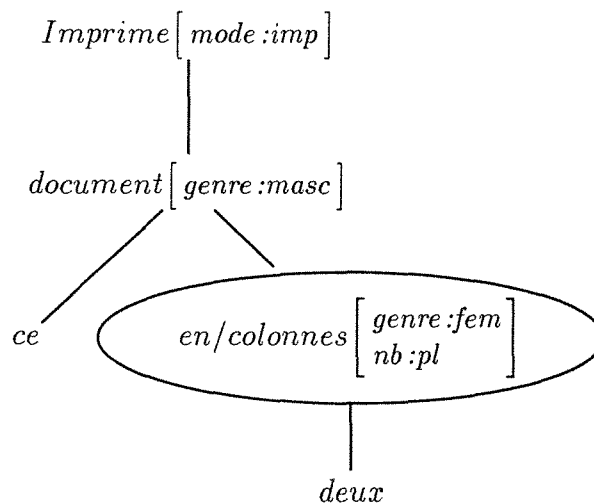


FIG. 7.7 - Structure de dépendance (1)

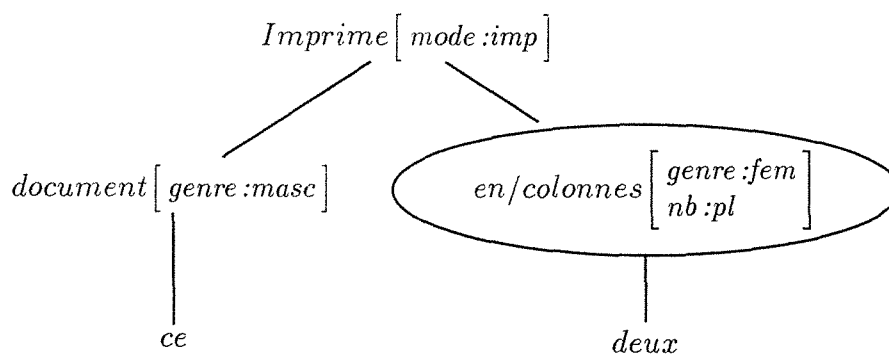


FIG. 7.8 - Structure de dépendance (2)

La différence entre les deux, est que dans le premier cas, *en deux colonnes* serait déterminatif de *ce document*, et, de ce fait, n'aurait pas d'incidence par rapport au verbe *imprimer*.

Dès ce niveau, on peut faire entrer le calcul référentiel lié aux objets pour trancher entre les deux. On montre en effet que *ce N* permet de re-catégoriser un référent saillant en contexte relatif à N [Kleiber, 1991; Gaiffe, 1992]. Or, *en deux colonnes* ne va pas dans le sens d'une telle recatégorisation (il n'y a pas de statut particulier pour les documents en deux colonnes). La solution qui s'impose est alors d'effectuer une séparation entre le *GN* et le *GP*. Cette hypothèse se trouvant ainsi privilégiée, le verbe va intégrer au niveau de la structure de dépendance le *GN* et le *GP* (figure 7.8). Les éléments de cette structure vont être utilisés comme données de base dans la construction du graphe de déroulement. La forme impérative de l'énoncé va conduire à insérer préalablement dans le graphe un nœud correspondant à la spécification du référent du sujet intégré que porte le verbe.

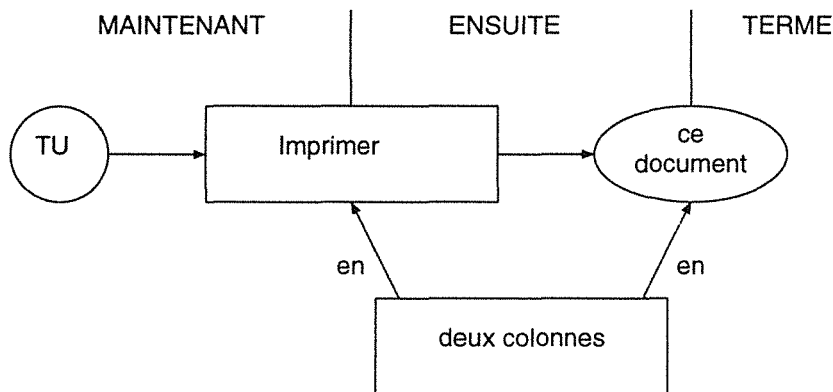


FIG. 7.9 - Schème de déroulement auquel on aboutit

A partir de là, l'activation avec succès des promoteurs (notamment avec le gabarit correspondant à l'arc *en* dans le schème de déroulement) permet d'aboutir au graphe de zones temporelles :

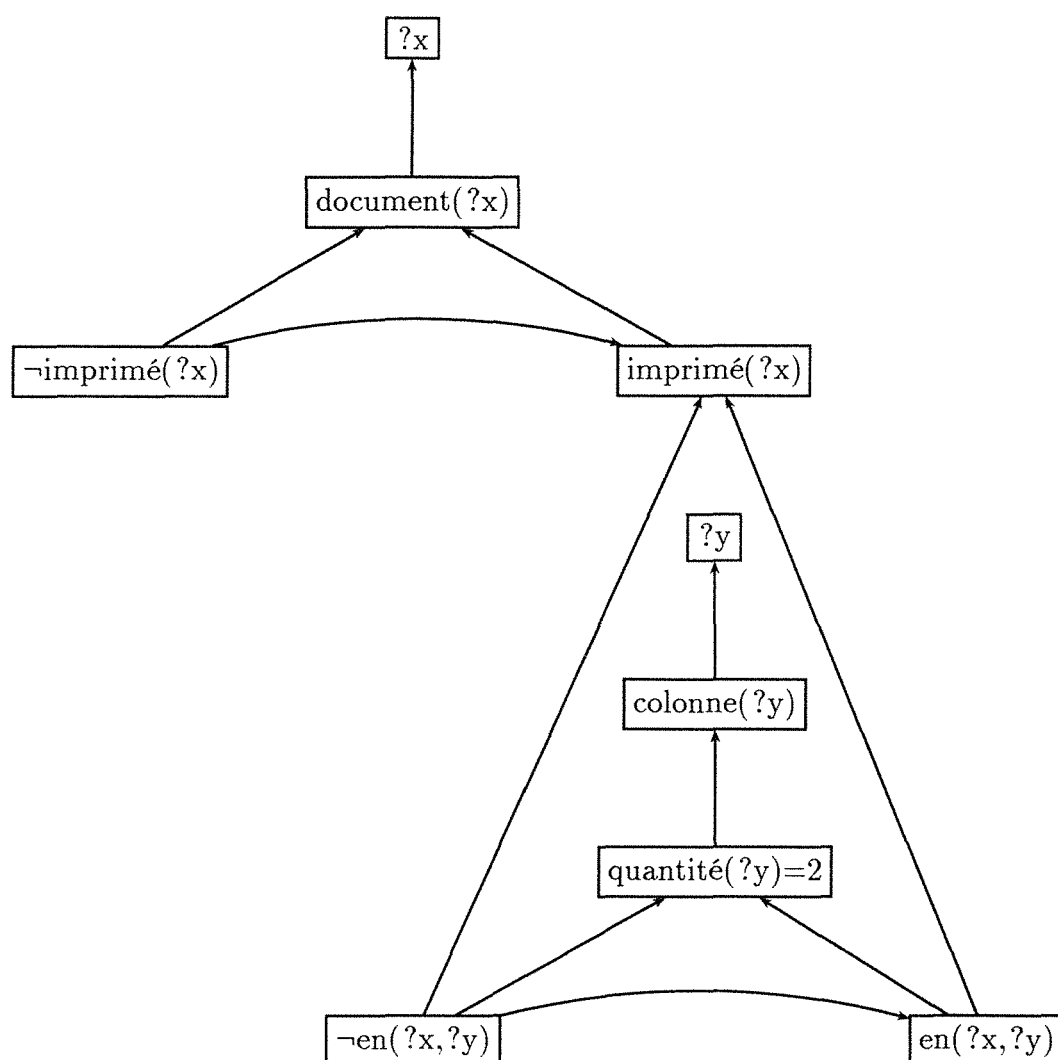


FIG. 7.10 - Représentation sous forme de zones temporelles (1)

Si l'on sort du domaine linguistique pour rentrer dans celui de l'application, on est amené à introduire de nouveaux référents (la sortie papier) ou à les ramener à des propriétés (les deux colonnes). Nous avons déjà mentionné que nous utilisons des clauses de Horn pour traduire ces données pragmatiques liées à la représentation de la logique de l'application.

$$Rel(y, x, SortiePapier) \leftarrow Type(x, fichier) \wedge Spec(x, imprime)$$

$$Prop(x, y, forme) \leftarrow Type(x, ?) \wedge Spec(x, y, en)$$

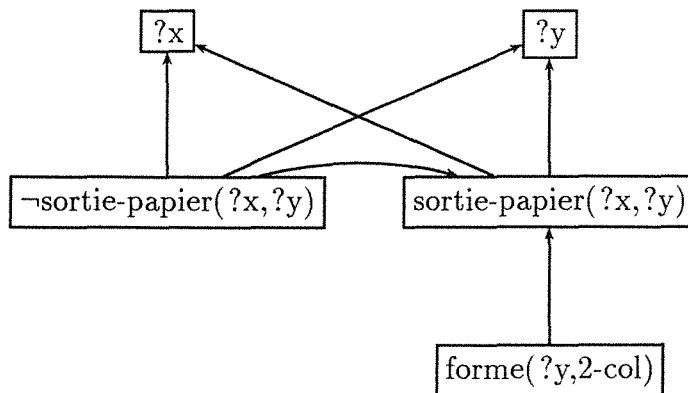


FIG. 7.11 - Représentation sous forme de zones temporelles (2)

On peut alors fournir une situation initiale et un état final au planificateur, sous la forme du problème suivant :

```
(make-problem
  :domain #'domaine-de-l-application
  :inits '((en-ascii TOTO) (fichier TOTO) (imprimante LJ1))
  :goal '(:and (hors-de SORTIE LJ1)
    (sortie-papier-de SORTIE TOTO)
    (forme SORTIE 2-col))
```

Le résultat fourni est alors la déduction de l'enchaînement `convertit-ascii-ps`, `extension-conversion-2-col`, `imprime`.

```
(define (operator convertit-ascii-ps)
  :parameters (?X ?Y)
  :precondition (en-ascii ?X)
  :effects
  ((:effect (:and (fichier ?Y)
    (est-une-conversion-de ?Y ?X )
    (en-postscript ?Y )))))
```

```
(define (operator extension-conversion-2-col)
  :parameters (?X ?Y)
  :effects
  ((:effect (:and (fichier ?Y)
    (est-une-conversion-de ?Y ?X)
    (forme ?Y 2-col)
    (en-postscript ?Y))))))
```

```
(define (operator imprime)
  :parameters ((fichier ?X) ?Y)
  :effects
  ((:effect (sortie-papier-de ?Y ?X))
   (:effect (forme ?Y 2-col)
            :when (:and (forme ?X 2-col)
                        (en-postscript ?X ))))))
```

On pourrait envisager de modéliser le tube (|) pour obtenir directement la commande Unix à effectuer.

3 Conclusion

Dans ce chapitre, nous avons mis en évidence une méthode pratique pour aboutir à un état final spécifié par l'énoncé, comme nous en avons montré l'intérêt au chapitre 3. Notamment, il devient possible d'envisager toutes sortes de combinaisons, comme par exemple :

(7.4) Imprime ce document en recto-verso sur deux colonnes.

sans rien rajouter au système. Cela ne serait pas possible avec les méthodes classiques, pour lesquelles il faudrait prévoir un nouveau script associé à l'énoncé. Il est en outre important de préciser que cette méthode n'ôte aucune fonctionnalité par rapport à ce que réalisent déjà les méthodes traditionnelles. Nous proposons ainsi une approche structurée qui permet d'aboutir à une bonne orthogonalité de la syntaxe sans pour autant compliquer le système.

Conclusion

1 Bilan

En posant le problème de la référence aux actions en les termes référentiels d'un état auquel on doit parvenir, on améliore les systèmes de dialogue de commande tant du point de vue de leur conception que de leur utilisation. Du point de vue de la conception, on aboutit à des composantes effectivement autonomes (linguistique, résolution de problème par un planificateur, dialogue) ainsi qu'à une simplification des modèles linguistiques et de l'application. Tout en mettant en évidence une méthodologie pour les construire, cela permet d'avoir une vision plus saine d'une conception de la sémantique comme lien inférentiel entre des identificateurs linguistique et ceux introduits par le modèle de l'application. Cela ouvre de nouvelles perspectives selon lesquelles on peut concevoir et exploiter un lexique relatif à un domaine d'application. Du point de vue de l'utilisation pratique, on aboutit à des systèmes qui se prêtent davantage à la prise en compte d'une régularité langagière, apportant ainsi une plus grande consonance naturelle. Enfin, un énoncé peut désormais être vu comme la description d'une tâche que prescrit un utilisateur. Dès lors, il devient envisageable de se situer par rapport à un niveau d'abstraction variable des tâches, et plus seulement par rapport à celles, élémentaires, que l'on peut associer à des fonctions de l'application. On peut ainsi espérer aboutir à un réel partage d'activité entre utilisateur et système.

2 Pour aller plus loin

2.1 Une meilleure prise en compte du déroulement des processus

On a montré que l'on ne pouvait considérer les verbes d'un point de vue monolithique, mais qu'il fallait les envisager selon les différentes phases du déroulement qu'ils dénotent. Dans le cadre que l'on s'est fixé, c'est à juste titre la partie « acquis de situation sur l'objet » que l'on a exploité : dans les dialogues de commande, il s'agit bien du phénomène essentiel. On peut toutefois rencontrer dans des dialogues homme-machine des énoncés où l'intention de l'utilisateur se rattache au déroulement lui-même. Par exemple :

(7.5) Tu trouveras la porte facilement.
opposé à

(7.6) Tu trouveras la porte ouverte.

Dans ce cas, « facilement » s'applique au processus de recherche qui précède la découverte de la porte en question. On retrouve le même phénomène dans des énoncés plus typiques de dialogue tels que dans :

(7.7) Veuillez patientez, s'il vous plait.

La subordination des verbes aux objets ne peut porter que sur une partie du déroulement temporel. Il serait intéressant de pouvoir préciser une phase transitoire avant d'en arriver à l'état final, typiquement par le biais des éléments relatifs à l'expression du moyen.

On a travaillé à acte de langage constant : l'acte directif. Il serait cependant intéressant d'examiner dans quelle mesure notre étude pourrait s'étendre à d'autres actes de langages. Cela revient donc à se demander si le modèle que l'on a proposé n'aurait pas une couverture plus large que le cadre des dialogues de commande que l'on s'était préalablement fixé.

2.2 Prise en compte d'autres actes de langages

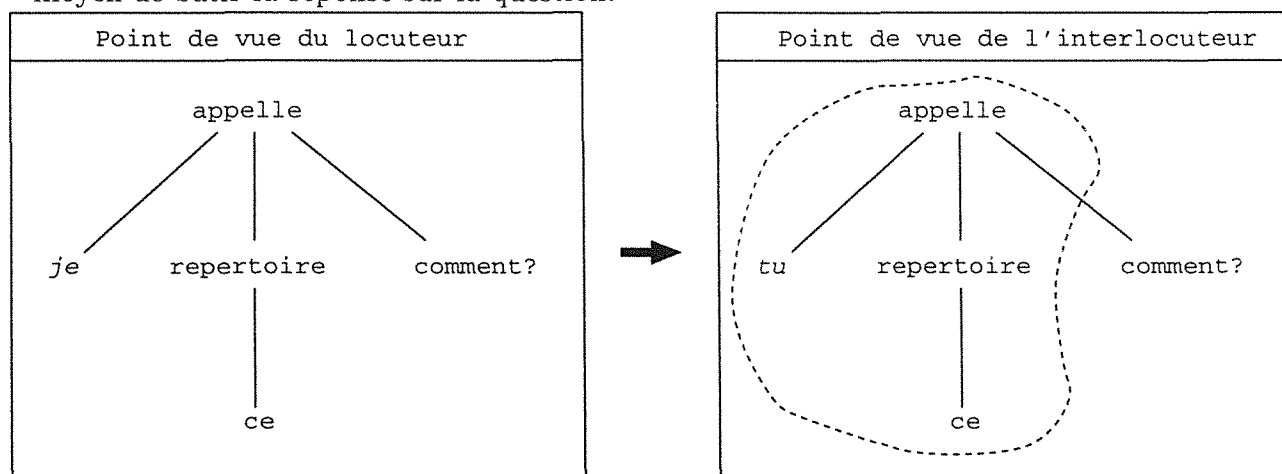
Quels que soient les actes de langage considérés, on retrouve classiquement les mêmes types de représentations d'énoncés qui sont utilisées (essentiellement sous forme arborescente sémantique ou logique) inspirés de théories faisant appel aux grammaires de cas. Par exemple, les grammaires lexico-fonctionnelles utilisent la même forme logique pour représenter une phrase déclarative ou interrogative, la seule différence apparaissant au niveau d'un trait exprimant le type de la phrase.

Or, nous avons montré que notre approche, représentant un énoncé dynamiquement sous forme d'un graphe de déroulement, répondait à certaines impasses rencontrées avec ces représentations. De plus, une telle conception trouvant écho dans certaines théories de linguistique générale [Guillaume, 1970; Pottier, 1985; Dervillez-Bastuji, 1982; Rousseau, 1993] ou de sémantique formelle [Dowty, 1979; Parsons, 1990] représentant le prédicat avec le déroulement qu'il induit, nous pourrions nous demander dans quelle mesure notre modèle de référence aux actions pourrait être étendu.

Ainsi, dans des cas de demande d'aide tels que :

(7.8) Comment j'appelle ce répertoire?

notre approche, outre le fait de représenter la question, présente l'intérêt de fournir le moyen de bâtir la réponse sur la question.



Au niveau du graphe de déroulement, cela reviendrait à incorporer la partie en pointillés avec le statut *implicite*. Cela a pour conséquence qu'il y aura prise en compte de la structure si et seulement si le contenu préalable du graphe n'est pas recouvert par l'explicite de l'énoncé qui suit.

3 Perspectives à long terme

Lorsque l'on s'intéresse aux phénomènes linguistiques, en l'occurrence l'expression d'une action, il est d'usage de considérer les divers signes linguistiques qui se trouvent impliqués selon deux angles⁹ :

- du point de vue de toute la potentialité qu'ils renferment en faisant abstraction de leurs emplois particuliers.
- du point de vue de leur utilisation concrète dans un discours.

Le second angle reflète l'aspect tangible des choses, tel qu'on le lit ou l'entend. C'est celui que Benveniste appelle la langue en action dans le discours, s'offrant à l'observation directe. En conséquence, il s'est trouvé fortement privilégié dans la linguistique traditionnelle. Il faut le voir comme le siège des effets de sens multiples qui résultent de l'emprise de la réalité sur le discours. Par contre, selon le premier point de vue, on ne s'intéresse plus au produit fini que constitue l'énoncé en contexte, mais aux facteurs qui l'ont précédé en conduisant à sa production. On est donc en mesure de mettre en évidence pourquoi cet énoncé a été construit. On dépasse ainsi le cadre de l'observation pour rentrer dans celui de l'explication. C'est dans cette optique que nous avons essayé de nous situer dans la démarche que nous proposons.

Or, même si l'on a tôt fait la différence entre langue et discours, c'est ce dernier, la portion d'énoncé produite, l'aspect visible des choses, qui a toujours été plus ou moins privilégié, excepté dans le courant issu de la psychomécanique du langage de Guillaume. Son mérite a été de se préoccuper des opérations de pensée qui conduisent à l'élaboration des formes linguistiques observables. Son défi a été de remonter « de la multiplicité des conséquences du signe linguistique dans le parler réel à son unité de condition dans la langue virtuelle ».

L'intérêt est de taille en ce qui concerne la linguistique : dès que l'on peut expliquer un phénomène, on peut en construire une théorie. A partir de là, il devient possible de comparer ce que prédit la théorie avec ce que l'on observe dans la réalité du discours et remanier la théorie en conséquence. De la sorte, on évite le traitement au cas par cas. On en voit aussi l'intérêt en ce qui nous concerne. Ce n'est en effet que si l'on fait émerger une théorie, que l'on peut espérer pouvoir l'informatiser.

9. Cette dichotomie est classiquement rendue depuis Saussure avec l'opposition Langue/Parole ou Langue/Discours, et s'est trouvée remaniée plus tard par l'école chomskienne en termes de Compétence/Performance.

Bibliographie

- [Ait-Kaci et Lincoln, 1989] H. Ait-Kaci et P. Lincoln. Life, a natural language for natural language. *T.A Informations, Bulletin semestriel de l'Atala*, 30, 1989.
- [Allen et Perrault, 1980] J.F. Allen et C.R. Perrault. Analyzing intentions in utterances. *Artificial Intelligence*, 15:143–178, 1980.
- [Allen, 1984] J.F. Allen. Toward a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
- [Asher et Sablayrolles, 1994] N. Asher et P. Sablayrolles. A typology and discourse semantics for motion verbs and spatial pps in french. *Journal of Semantics*, 1994. to be published.
- [Austin, 1962] J.L. Austin. *Quand dire, c'est faire*. Seuil, 1962.
- [Azémard et Macchion, 1993] F. Azémard et J.-P. Macchion. From an event to a conceptual graph to interpret in a multimodal context. Dans *Applications of Speech Technology*, pages 171–174, Lautrach Conference Center Bavaria, Germany, 16-17 September 1993. Joint ESCA-NATO/RSG10, NCSA.
- [Barton *et al.*, 198] G.E. Barton, R.C. Berwick, et E.S. Ristad. *Computational Complexity and Natural Language*. MIT Press, London, 198?
- [Bastien et Colas, 1991] C. Bastien et C. Colas. Modélisation de la tâche d'un opérateur. Dans *Science et Défense*, pages 329–340. Dunod, 1991.
- [Bellalem et Romary, 1993] N. Bellalem et L. Romary. Le dialogue homme-machine multimodal: vers la compréhension du geste de désignation. Dans *Interfaces pour monde virtuel et réel*, pages 217–228, Montpellier, 1993.
- [Bellalem, 1994] N. Bellalem. Interprétation de la désignation par entourage dans un dialogue homme-machine multimodal. Dans *Journées des jeunes chercheurs*, 1994.
- [Benveniste, 1966] E. Benveniste. Les niveaux de l'analyse linguistique. Dans *Problèmes de linguistique générale, Tome 1*, chapitre 10, pages 119–131. Gallimard, 1966.
- [Benveniste, 1974a] E. Benveniste. Les transformations des catégories linguistiques. Dans *Problèmes de linguistique générale, Tome 2*, chapitre 9, pages 126–136. Gallimard, 1974.

- [Benveniste, 1974b] E. Benveniste. Sémiologie de la langue. Dans *Problèmes de linguistique générale, Tome 2*, chapitre 3, pages 43–66. Gallimard, 1974.
- [Benveniste, 1974c] E. Benveniste. Structure des relations d'auxiliarité. Dans *Problèmes de linguistique générale, Tome 2*, chapitre 13, pages 177–193. Gallimard, 1974.
- [Berrendonner *et al.*, 1992] A. Berrendonner, M. Frejdi, F. Oquendo, et J. Rouault. Un système inférentiel orienté objet pour des applications en langue naturelle. Dans *COLING*, pages 461–467, Nantes (France), 23-28 août 1992.
- [Boe et Lienard, 1988] L.-J. Boe et J.-S. Lienard. La communication parlée est-elle une science? Eléments de discussion et de réflexion suivis de repères chronologiques. Dans *Actes des 17^{mes} Journées d'Etude sur la Parole*, pages 79–92, Nancy (France), 20-22 septembre 1988. Société Française d'Acoustique - Groupe Communication Parlée.
- [Borillo, 1991] A. Borillo. Temps notionnel, temps grammatical: quelques faits linguistiques concernant la durée. Dans *Approches formelles de la sémantique naturelle*, rédacteurs A. Borillo, M. Borillo, L. Fari nas del Cerro, et J. Virbel. C.N.R.S. - U.P.S. - U.T.M. - A.D.I., Toulouse, 31 août - 5 septembre 1991. 1^{re} école d'été de linguistique pour informaticiens.
- [Bourguet, 1992] M.-L. Bourguet. Icpplan: Dialogue multimodal pour la conception de plans architecturaux. pages 369–374, 1992. 19^{èmes} JEP Bruxelles.
- [Bresnan et Kaplan, 1981] J. Bresnan et R. Kaplan. *Lexical-functional grammar: a formal system for grammatical representation*, chapitre 4. MIT Press, Cambridge, 1981. in *The mental representation of grammatical relations*.
- [Burton, 1976] R. Burton. Semantic grammar. an engineering technique for constructing natural language understanding system. Rapport Technique 3353, BNN, Cambridge USA, 1976.
- [Cadiot, 1993] P. Cadiot. De et deux de ses concurrents: avec et à. Dans *La couleur des prépositions, Langages*, numéro 110, pages 68–106. Larousse, juin 1993.
- [Caelen *et al.*, 1991] J. Caelen, P. Garcin, J. Wretö, et E. Reynier. Interaction multimodale autour de l'application icp-draw. Dans *IHM'91*, Dourdan, 1991.
- [Caelen et Coutaz, 1991] J. Caelen et J. Coutaz. Interaction homme-machine multimodale: problèmes généraux. Dans *IHM'91*, Dourdan, 1991.
- [Caelen, 1991] J. Caelen. Interaction multimodale dans icp-draw. expérience et perspectives. *Greco PRC*, avril 1991.
- [Calliope, 1989] rédacteur Calliope. *La parole et son traitement automatique*. Collection Technique et Scientifique des Télécommunications. Masson, Paris, 1989.
- [Caron, 1993] J. Caron. *Les régulations du discours, psycholinguistique et pragmatique du langage*. P.U.F., Paris, 1993.

- [Carroll et Thomas, 1980] J.M. Carroll et J.C. Thomas. Metaphor and the cognitive representation of computing systems. RC 8303, IBM Watson Research Center, New York, 1980.
- [Cervoni, 1991] J. Cervoni. *La préposition - Etude sémantique et pragmatique*. Champs linguistiques. Duculot, Paris, 1991.
- [Charniak, 1978] E. Charniak. On the use of framed knowledge in language comprehension. *Artificial Intelligence*, 11 :225-265, 1978.
- [Charpentier *et al.*, 1992] F. Charpentier, F. Gavignet, K. Choukri, F. Andry, E. Bilange, et J.-Y. Magadur. Un système de dialogue oral pour une application de réservation téléphonique de billets d'avion. Dans *Actes des 19^{mes} Journées d'Etude sur la Parole*, pages 363-368, 1992.
- [Chignell et Hancock, 1988] M.H. Chignell et P.A. Hancock. Intelligent interface design. Dans *Handbook of Human-Computer Interaction*, rédacteur M. Helander, chapitre 46, pages 969-995. Elsevier Science Publishers B.V. (North-Holland), 1988.
- [Chomsky, 1957] Chomsky. *Syntactic Structures*. Mouton, 1957.
- [Chomsky, 1965] N. Chomsky. *Aspects of the theory of syntax*. MIT Press, Cambridge, Massachussets, 1965.
- [Clark, 1986] R. Clark. Predication and paronymous modifiers. *Notre Dame Journal of formal logic*, 27(3) :376-392, 1986.
- [Constant, 1991] P. Constant. *Analyse syntaxique par couche*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1991.
- [Coutaz, 1990] Joëlle Coutaz. *Interfaces Homme-Ordinateur*. Dunod, 1990.
- [Danlos *et al.*, 1985] L. Danlos, S. Guez, et S. Sabbagh. Conception d'un système d'aide intelligent. *Intellectica*, 1(1) :91-100, 1985.
- [Davidson, 1967] D. Davidson. The logical form of action sentences. Dans *Essays on Actions and Events*. Clarendon Press, Oxford, 1967.
- [Davis et Hersh, 1986] P.J. Davis et R. Hersh. *Descartes dream : The World according to Mathematics*. Orlando, Florida (United States), 1986.
- [Delouis, 1993] I. Delouis. *LISA : un langage réflexif pour la modélisation du contrôle dans les systèmes à base de connaissances. Application à la planification des réseaux électriques*. PhD thesis, Université d'Orsay, 1993.
- [Dervillez-Bastuji, 1982] J. Dervillez-Bastuji. *Structures des relations spatiales dans quelques langues naturelles*. Librairie DROZ Genève-Paris, 1982.
- [Desclès, 1990] J.-P. Desclès. *Langages applicatifs, langues naturelles et cognition*. Hermès, Paris, 1990.

- [Desclès, 1993] J.-P. Desclès. Le passif dans le système des voix du français. Dans *Sur le passif, Langages*, numéro 109, pages 73–102. Larousse, mars 1993.
- [Dowty, 1979] D. Dowty. *Word Meaning and Montague Grammar*. Reidel, Dordrecht, 1979.
- [Dubois, 1967] J. Dubois. *Grammaire structurale du français - Le verbe*. Larousse, 1967.
- [Ducrot et Todorov, 1972] O. Ducrot et T. Todorov. *Dictionnaire encyclopédique des sciences du langage*. Seuil, Paris, 1972.
- [Duermael et al., 1994] F. Duermael, B. Gaiffe, et J.-M. Pierrel. Un moteur à deux temps pour référer aux actions. Dans *Ergonomie et Informatique Avancée*, Bayonne (France), 26-28 octobre 1994.
- [Duermael et Gaiffe, 1993] F. Duermael et B. Gaiffe. Referring to actions in man-machine command dialogues. Dans *Eurospeech '93, 3rd European Conference on Speech Communication and Technology, Berlin, Germany*, pages 959–964. European Speech Communication Association, September 1993.
- [Duermael et Pierrel, 1992] F. Duermael et J.-M. Pierrel. Un système de dialogue multimodal pour poste de travail intelligent fondé sur une grammaire lexicale fonctionnelle. Dans *Actes des 19^{mes} Journées d'Etude sur la Parole*, Bruxelles, 19-22 mai 1992.
- [Eco, 1975] U. Eco. *Lector in Fabula*. Livre de Poche, 1975.
- [Falzon et al., 1991] P. Falzon, M.D. Wilson, D. Sedlock, et J.L. Binot. An architecture for multimodal dialogue. Dans *Pre-proceedings of the Second Venaco Workshop on the structure of multimodal dialogue*, Maratea (Italy), September 1991.
- [Falzon, 1984] P. Falzon. Les langages opératifs. Dans *Actes du séminaire GRECO sur le dialogue oral homme-machine*, Nancy (France), 1984.
- [Fikes et al., 1974] R.E. Fikes, P.E. Hart, et N.J. Nilsson. Learning and executing generalised robot plans. *Artificial Intelligence*, 3, 1974.
- [Fillmore, 1968] C. Fillmore. The case for case. Dans *Universals in linguistics theory*, rédacteurs Rinehart et Winston, pages 1–90. E. Bach and R.T. Harm, 1968.
- [Frege, 1971] G. Frege. *Ecrits logiques et philosophiques*. Seuil, Paris, 1971.
- [Gaiffe, 1992] B. Gaiffe. Référence et dialogue homme-machine: vers un modèle adapté au multi-modal, 1992. Thèse d'Université de Nancy I.
- [Gaiffe, 1993] B. Gaiffe. Ce matin, il pleuvait, ou la référence à l'épreuve du temps. *Langages*, (112):p. 74–92, 1993.
- [Gerlach et Onken, 1993] M. Gerlach et R. Onken. Speech input/output as interfaces devices for communication between aircraft pilots and the pilot assistant system "cassy". Dans *Applications of Speech Technology*, pages 87–90, Lautrach Conference Center Bavaria, Germany, 16-17 September 1993. Joint ESCA-NATO/RSG10, NCSA.

- [Green et Payne, 1984] T.R.G. Green et S.J. Payne. Organization and learnability in computer languages. *International Journal of Man-Machine Studies*, 21 :7-18, 1984.
- [Green, 1979] T.R.G. Green. The necessity of syntax markers: two experiments with artificial languages. *Journal of Verbal learnig and Verbal Behavior*, 18 :481-496, 1979.
- [Greimas, 1966] A. Greimas. *Sémantique structurale*. Larousse, 1966.
- [Grevisse, 1986] M. Grevisse. *Le bon usage*. Duculot, 1986.
- [Grice, 1979] H.P. Grice. Logique et conversation. *Communications*, 30 :57-72, 1979. Traduction française effectuée par O. Ducrot.
- [Guillaume, 1970] G. Guillaume. *Temps et verbe - Théorie des aspects, des modes et des temps*. Honoré Champion, Paris, 1970.
- [Haiman, 1980] J. Haiman. Iconicity of grammar: Isomorphism and motivation. *Language*, 56(3) :525-540, 1980.
- [Hartson et Hix, 1989] H.R. Hartson et D. Hix. Human-computer interface development : Concepts and systems for its management. *ACM Computing Surveys*, 21(1), March 1989.
- [Haton et Haton, 1989] J.-P. Haton et M.-C. Haton. L'intelligence artificielle. Dans *Que sais-je ?* Presses Universitaires de France, Paris, 1989.
- [Helander, 1988] rédacteur M. Helander. *Handbook of Human-Computer Interaction*. North Holland, 1988.
- [Hoc, 1980] J.-M. Hoc. L'articulation entre la description de la tâche et la caractérisation de la conduite dans l'analyse du travail. *Bulletin de psychologie*, 33(344) :207-212, 1980.
- [Hoc, 1988] J.-M. Hoc. *Cognitive Psychology of Planning*. Computers and People. AP, 1988.
- [Hopper et Thompson, 1980] P.J. Hopper et S.A. Thompson. Transitivity in grammar and discourse. *Language, Journal of the linguistic society of America*, 56(2) :251-299, June 1980.
- [Ishiguro, 1990] H. Ishiguro. *Leibniz's philosophy of logic and language (2nd edition)*. Cambridge University Press, Cambridge, 1990.
- [Jardino et Adda, 1993] M. Jardino et G. Adda. Language modelling for csr of large corpus using automatic classification of words. Dans *Eurospeech '93, 3rd European Conference on Speech Communication and Technology, Berlin, Germany*, pages 1191-1194, Berlin, 1993. ESCA.
- [Jespersen, 1971] O. Jespersen. *La philosophie de la grammaire*. Tel. Gallimard, 1971.

- [Johnson, 1988] M. Johnson. *Attribute-value logic and the theory of grammar*, volume 16 de *Lecture Notes*. Center for the study of language and information, Stanford, 1988.
- [Junqua et Morin, 1993] J.-C. Junqua et P. Morin. Towards successful and usable multi-modal applications using speech technology. Dans *Applications of Speech Technology*, pages 191–195, Lautrach Conference Center Bavaria, Germany, 16-17 September 1993. Joint ESCA-NATO/RSG10, ESCA.
- [Kamp, 1992] H. Kamp. The representation of tense and aspect in drt-theory (1). Support de cours de l'école d'été in *Linguistics Logic and Information* (Colchester, Essex), August 1992.
- [Kaplan, 1979] J. Kaplan. A general syntactic processor. Dans *Natural language processing*, rédacteur Rustin, pages 193–241. Algorithmic Press, New York, 1979.
- [Katz et Fodor, 1966] J.J. Katz et J. Fodor. Structure d'une théorie sémantique. *Cahiers de Lexicologie*, II:39–72, 1966.
- [Kaufmann, 1991] I. Kaufmann. Semantic and conceptual aspects of the preposition *durch*. Dans *The semantics of prepositions: from mental processing to natural language processing*, rédacteurs Correlia, Zelinsky, et Wibbelt, pages 221–247. Mouton de Gruyter, 1991.
- [Kayser, 1988] D. Kayser. Le raisonnement à profondeur variable. *Actes des journées du PRC IA Toulouse, Teknea*, pages 109–136, 1988.
- [Keenan, 1985] E. Keenan. Passive in the world's languages. Dans *Language, typology and syntax description*, rédacteur T. Shopen, pages 243–281. Cambridge University Press, 1985.
- [Kleiber, 1990] G. Kleiber. *La sémantique du prototype, catégories et sens lexical*. P.U.F, 1990.
- [Kleiber, 1991] G. Kleiber. Anaphore-deixis: où en sommes-nous? *L'information grammaticale*, (51), octobre 1991.
- [Kleiber, 1992] G. Kleiber. Cap sur les topiques avec le pronom *il*. *L'information grammaticale*, (54), juin 1992.
- [Kleiber, 1993] G. Kleiber. *Iconicité d'isomorphisme et grammaire cognitive*, pages 105–121. *Faits de langue*, 1993.
- [Klein, 1990] J. Klein. Contribution à la commande orale d'un robot doté d'un système de vision, 1990. Thèse d'Université de Nancy I.
- [Lakoff, 1965] G. Lakoff. *On the nature of syntactic irregularity*. PhD thesis, Indiana University, 1965.
- [Langacker, 1991] R.W. Langacker. *Concept, Image and Symbol; The Cognitive Basis of Grammar*. Mouton de Gruyter, 1991.

- [Laur, 1993] D. Laur. La relation entre le verbe et la préposition dans la sémantique du déplacement. Dans *La couleur des prépositions, Langages*, numéro 110, pages 68–106. Larousse, juin 1993.
- [Leplat et Hoc, 1983] J. Leplat et J.-M. Hoc. Tâche et activité dans l'analyse psychologique des situations. *Cahiers de Psychologie Cognitive*, pages 49–63, 1983.
- [Lyons, 1977] J. Lyons. *Semantics*. Cambridge University Press, 1977.
- [Manaris *et al.*, 1994] B.Z. Manaris, J.W. Pritchard, et W.D. Dominick. Developing a natural language interface for the unix operating system. *SIGCHI Bulletin*, 26(2):34–40, April 1994.
- [Manaris et Dominick, 1993] B.Z. Manaris et W.D. Dominick. Nalige: a user interface for the development of natural language interfaces. *International Journal of Man-Machine Studies*, 38:891–921, 1993.
- [McCarthy et Hayes, 1969] J. McCarthy et P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. Dans *Machine Intelligence*, rédacteurs D. Michie et B. Melzer, chapitre 4. Scotland University Press, Edinburg, 1969.
- [McCawley, 1968] J.D. McCawley. Lexical insertion in a transformational grammar without deep structure. *CLS*, 4:71–80, 1968.
- [McDermott, 1982] D. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–115, 1982.
- [Meinadier, 1991] J.-P. Meinadier. *L'interface utilisateur - Pour une informatique plus conviviale*. Informatique et stratégie. Dunod, Paris, 1991.
- [Mel'cuk et Pertsov, 1987] I. Mel'cuk et N.V. Pertsov. *Surface syntax of english: a formal model within the Meaning-Text framework*. J. Benjamin, Amsterdam, 1987.
- [Memmi, 1990] D. Memmi. Le symbole en intelligence artificielle. Dans *ARC90*, 1990.
- [Mignot *et al.*, 1993] C. Mignot, C. Valot, et N. Carbonell. An experimental study of future « natural » multimodal human-computer interaction. Dans *Proceedings INTERCHÍ93, Conference on Human Factors in Computing Science INTERACT'93 and CHÍ93*, Amsterdam (The Netherlands), April 1993.
- [Milner, 1982] J.-C. Milner. *Ordres et raisons de la langue*. Seuil, Paris, 1982.
- [Moeshler et Reboul, 1994] J. Moeshler et A. Reboul. *Dictionnaire encyclopédique de pragmatique*. Seuil, Paris, 1994.
- [Moignet, 1973] G. Moignet. Incidence verbale et transitivité. *Trav. ling. litt.*, 11(1):363–379, 1973.

- [Monclar *et al.*, 1994] F. R. Monclar, I. Delouis, et J.-P. Krivine. La coopération homme-machine dans les systèmes à base de connaissances : un problème de contrôle. Dans *Ergonomie et Informatique Avancée*, Bayonne (France), 26-28 octobre 1994.
- [Morgan, 1978] J.M. Morgan. Two types of convention in indirect speech acts. Dans *Syntax and Semantics 8: Pragmatics*, rédacteur P. Cole, pages 245-249. Academic Press, New York, 1978.
- [Morin et Pierrel, 1987] P. Morin et J.-M. Pierrel. Partner : un système de dialogue homme-machine. Dans *Cognitiva 87*, 1987.
- [Morki, 1984] K. Morki. Customer's requirements for natural language systems : Results of an inquiry. *International Journal of Man-Machine Studies*, 21 :401-414, 1984.
- [Parsons, 1990] T. Parsons. *Events in the semantics of English - A study in subatomics Semantics*, volume 19 de *Current Studies in Linguistics*. MIT Press, Cambridge (Mass), 1990.
- [Peirce, 1978] C. S. Peirce. *Ecrits sur le signe*. Seuil, 1978. Présentation et traduction de G. Deladalle.
- [Penberthy et Weld, 1993] J.S. Penberthy et D.S. Weld. Ucpop : A sound, complete, partial order planner for adl. ftp, 1993.
- [Pierrel, 1987] J.-M. Pierrel. *Dialogue Oral Homme-Machine, connaissances linguistiques, stratégies et architectures des systèmes*. Hermès, Paris, 1987.
- [Pitrat, 1990] J. Pitrat. *Métaconnaissance, futur de l'IA*. Hermès, 1990.
- [Poirier et Lefebvre, 1993] F. Poirier et P. Lefebvre. Une interface multimodale de dialogue homme-machine sous unix. Dans *Interfaces pour monde virtuel et réel*, pages 161-170, Montpellier, 1993.
- [Pottier, 1962] B. Pottier. *Systématique des éléments de relation*. Klincksieck, Paris, 1962.
- [Pottier, 1985] B. Pottier. *Linguistique générale, théorie et description*. Klincksieck, 1985.
- [Pottier, 1987] B. Pottier. *Théorie et analyse en linguistique*. Langue Linguistique Communication. Hachette, 1987.
- [Pouteau *et al.*, 1993] X. Pouteau, B. Gaiffe, et J.-M. Pierrel. A knowledge-based approach towards operative multimodal dialogues : Melodia experiment. Dans *Applications of Speech Technology*, pages 183-186, Lautrach Conference Center Bavaria, Germany, 16-17 September 1993. Joint ESCA-NATO/RSG10, NCSA.
- [Putnam, 1975] H. Putnam. *Mind, language and reality. Philosophical papers 2*. Cambridge University Press, Cambridge, 1975.
- [Quine, 1960] W.V.O. Quine. *Word and objects*. MIT Press, Cambridge (Mass.), 1960.

- [Rasmussen, 1986] J. Rasmussen. *Information Processing and Human-Machine Interaction. An approach to cognitive engineering*. System Science and Engineering. North-Holland, 1986.
- [Rastier, 1987] F. Rastier. *Sémantique interprétative*. Formes sémiotiques P.U.F, 1987.
- [Rastier, 1991] F. Rastier. *Sémantique et recherches cognitives*. Formes sémiotiques P.U.F, 1991.
- [Reboul, 1993] A. Reboul. Le poids des pères, le choc des fils : prédicats de phase, modificateurs et identification. *Cahiers de linguistique française*, 14 :229–246, 1993.
- [Reichenbach, 1947] H. Reichenbach. *Elements of symbolic Logic*. Mc Millan, New York, 1947.
- [Romary et al., 1991] L. Romary, B. Gaiffe, et J.-M. Pierrel. Eléments de convivialité dans un dialogue de commande multimodal. Dans *Préactes IHM'91*, Dourdan (France), 11-13 décembre 1991.
- [Romary, 1989] L. Romary. Vers la définition d'un modèle cognitif pour la représentation du temps dans un système de dialogue homme-machine, 1989. Thèse d'Université de Nancy I.
- [Romary, 1991] L. Romary. Integration of spatial and temporal information produced by a natural language discourse. Dans *Conference Knowledge Modelling and Expertise Transfert*, Sophia Antipolis, April 1991.
- [Rousseau, 1993] A. Rousseau. Espace, référence, représentation. réflexion sur quelques conceptualisations de l'espace. *Faits de langue*, (1) :151–162, 1993.
- [Sabah, 1988] G. Sabah. *L'intelligence artificielle et le langage*. Hermès, Paris, 1988. Vol 1.
- [Sabah, 1989] G. Sabah. *Processus de compréhension*. Hermès, Paris, 1989. Vol 2.
- [Sacerdoti, 1974] E.D. Sacerdoti. A structure for plans and behavior. Dans *Artificial Intelligence Series*. Elsevier, 1974.
- [Sandewall, 1992] E. Sandewall. Non-monotonic reasoning about time and action. Fourth European summer school in logic, language and information, August 1992.
- [Schank et Abelson, 1977] R. Schank et R. Abelson. *Scripts, Plans, Goals and Understanding. An inquiry into Human Knowledge Structures*. Lawrence Erlbaum associates, 1977.
- [Searle, 1972] J.R. Searle. *Les actes de langage, essai de philosophie du langage*. Collection Savoir. Hermann, Paris, 1972. Publié en anglais en 1969 par Cambridge University Press sous le titre *Speech Acts*. Traduction française par Hélène Pauchard.
- [Searle, 1983] J. R. Searle. *Intentionality*. Cambridge University Press, 1983.

- [Searle, 1990] J.R. Searle. Les limites de l'intelligence artificielle. *Pour la science*, 1990.
- [Serres, 1974] M. Serres. *Discours et parcours*. L'identité. Grasset, Paris, 1974. Séminaire interdisciplinaire dirigé par C. Lévi-Strauss.
- [Somers, 1987] H. L. Somers. *Valency and case in computational linguistics*. Edinburgh university press, 1987.
- [Souvay, 1992] G. Souvay. Diapason, un environnement de développement pour l'intégration d'une entrée vocale dans des applications de type commande de machine, 1992. Thèse de l'Université de Nancy I.
- [Tamba, 1991] I. Tamba. La phrase japonaise et son double dispositif d'intégration des noms : les particules dites relationnelles et casuelles. *Langages*, (104), 1991.
- [Teil et Bellik, 1991] D. Teil et Y. Bellik. Système de dialogue multimodal sur une station de travail de type pc. Dans *IHM'91*, 1991.
- [Tesnière, 1969] L. Tesnière. *Éléments de syntaxe structurale*. Klincksieck, Paris, 1969.
- [Vandeloise, 1986] C. Vandeloise. *L'espace en français*. Editions du Seuil, Paris, 1986.
- [Vendler, 1967] Z. Vendler. *Linguistics in philosophy*, chapitre 4-5, pages 97-146. Cornell University Press, New York, 1967.
- [Verkuyl, 1972] H. Verkuyl. *On the compositional nature of the aspects*. Reidel (Dordrecht), 1972.
- [von Wright, 1963] G.H. von Wright. *Norm and action*. Humanities Press, 1963.
- [Wehrli, 1987] E. Wehrli. Interaction of syntax and semantics in natural language processing. *Cognitive Science and Applied Epistemology (CC-AI), Artificial Intelligence and Discourse Analysis*, 4(2/3) :255-265, 1987.
- [White, 1990] G. White. Natural language understanding and speech recognition. *Communications of the ACM*, 33(8), August 1990.
- [Wilensky, 1991] R. Wilensky. *Sentences, situations and propositions*. Morgan Kaufmann Publishers, 1991.
- [Wilks, 1975a] Y. Wilks. An intelligent analyser and understander of english. *Communications of the ACM*, 18(5), 1975.
- [Wilks, 1975b] Y. Wilks. A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence*, 6 :53-74, 1975.
- [Wilks, 1978] Y. Wilks. Making preferences more actives. *Artificial Intelligence*, 11 :197-223, 1978.
- [Winograd, 1972] T. Winograd. *Understanding Natural Language*. PhD thesis, Academic Press, 1972.

- [Winston, 1977] P. H. Winston. *The meaning of meaning*, chapitre 6, pages 157–178. Addison-Wesley, 1977.
- [Wittgenstein, 1961] L. Wittgenstein. *Tractatus logico-philosophicus*. Gallimard, 1961.
- [Woods, 1970] Woods. Transition network grammars for natural language analysis. *CACM*, pages 591–606, 1970.
- [Xie *et al.*, 1989] L. Xie, X. Du, et Z. Sun. An approach to general intelligent interface for hci. Dans *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, rédacteurs Salvendy G. et Smith M.J., pages 853–860. Elsevier, Amsterdam, 1989.
- [Xie et Du, 1992] L. Xie et X. Du. General natural language for operating systems. *SI-GART Bulletin*, 3(4):28–32, 1992.

**AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL POLYTECHNIQUE
DE LORRAINE**

o o o

VU LES RAPPORTS ETABLIS PAR :

Madame SOQUIERES Jeanine, Professeur, Université Nancy II,

**Monsieur MOESCHLER Jacques, Professeur, Université de Genève
(Suisse),**

Monsieur SABAH Gérard, Directeur de Recherche CNRS, LIMSI Orsay.

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur DUERMAEL Fabrice

à soutenir devant l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE,
une thèse intitulée :

**"Référence aux actions dans des dialogues de commande homme-
machine".**

en vue de l'obtention du titre de :

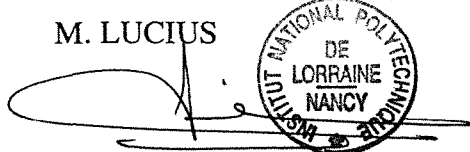
**DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE
LORRAINE**

Spécialité : **"INFORMATIQUE"**

Fait à Vandoeuvre le, **4 Octobre 1994**

Le Président de l'I.N.P.L.,

M. LUCIUS

A circular stamp of the Institut National Polytechnique de Lorraine Nancy is placed over the signature. The stamp contains the text 'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE NANCY' around the perimeter.

NANCY BRABOIS
2, AVENUE DE LA
FORET-DE-HAYE
BOITE POSTALE 3
F - 5 4 5 0 1
VANDŒUVRE CEDEX

Résumé

Dès que l'on propose une interface langagière pour commander une application informatique, il se pose la question des fonctions à activer en réponse à un énoncé de l'utilisateur. L'identification de ces fonctions constitue le problème de la référence aux actions. Classiquement, l'action (simple ou sous la forme d'un script de fonctions) à effectuer est dérivée directement de l'analyse syntaxico-sémantique, et les modifications que subissent les objets en sont déduites. Dans notre thèse, nous défendons le point de vue que cette identification suppose la prise en compte d'une temporalité inhérente aux énoncés de commande. Cela amène à considérer un énoncé comme la spécification d'un état final à atteindre, à travers les ruptures de propriétés qu'il prédique sur des objets. Dans cette optique, nous proposons un modèle de construction d'objets du discours pour référer aux actions en introduisant des graphes de déroulement et des éléments de contrôle (des promoteurs) associés aux noeuds représentant les phases du déroulement. L'utilité de ces promoteurs est de prendre en compte le contexte dynamique des portions de graphe qu'ils contrôlent, de restructurer ce graphe en fonction des propriétés aspectuelles mises en présence et d'alimenter le processus de référence. Cette approche de la référence aux actions permet de traiter plus simplement un langage qui reprend certaines régularités de construction propres au langage naturel, ce qui a pour conséquence de simplifier le modèle de l'application.

Mots-clés: Dialogue homme-machine, langue naturelle, référence aux actions.

Abstract

When designing an human-computer interface to control an application, the question may be raised of what functions are to be called, in response to the interpretation of a given user's utterance. The identification of these functions corresponds the general problem of the reference to actions. Classically, the action to be performed is derived from the parsing, and the modifications undergone by the objects are deduced. In our thesis, we argue that this identification involves taking into account a temporality inherent in command utterances. That leads to consider an utterance as the specification of a final state to be reached. We then propose a model of building of discourse objets by introducing some so-called development graphs and promotors to control them dynamically. This approach of the reference to actions enables us to process more easily a language that uses construction regularities of building which characterize natural languages. This results in a simplification of the application model.

Keywords: Man-Machine Dialogue, Natural Language, Reference to Actions.