POLYPUBLIE

Génération d'explications basées sur un algorithme de planification de texte à mémoire d'intentions

André Godin

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	1996
Type	Mémoire ou thèse / Dissertation or Thesis Mémoire de maîtrise/Master's thesis
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/8994/
Conditions d'utilisation Terms of use	© Tous droits réservés / All rights reserved
Référence Citation	Godin, A. (1996). Génération d'explications basées sur un algorithme de planification de texte à mémoire d'intentions [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8994/

Détails de la thèse
Thesis details

Directeurs de recherche Advisors	Jean-Charles Bernard, & Christian Langheit
Programme Program	Non spécifié

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

GÉNÉRATION D'EXPLICATIONS BASÉES SUR UN ALGORITHME DE PLANIFICATION DE TEXTE À MÉMOIRE D'INTENTIONS

ANDRÉ GODIN
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
OCTOBRE 1996

© André Godin, 1996.

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-26476-9

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

GÉNÉRATION D'EXPLICATIONS BASÉES SUR UN ALGORITHME DE PLANIFICATION DE TEXTE À MÉMOIRE D'INTENTIONS

présenté par: GODIN André
en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de:

M. ROBERT Jean Marc, Doctorat, président
M. BERNARD Jean-Charles, Ph.D., membre et directeur de recherche
M. LANGHEIT Christian, Ph.D., membre et codirecteur de recherche
M. CORRIVEAU Philippe, M.Sc.A., membre

à mes parents

et Chantal

REMERCIEMENTS

J'aimerais d'abord remercier M. Jean-Charles Bernard d'avoir accepté d'agir à titre de directeur de recherche.

Je remercie sincèrement mon codirecteur, M. Christian Langheit, pour ses conseils, son appui et ses encouragements tout au long de ce projet de recherche.

Je tiens également à remercier M. Michel Manseau, Chef de division, Informatique scientifique, à l'IREQ, de m'avoir donné l'opportunité de travailler sur ce projet.

Finalement, merci à tous ceux qui m'ont manifesté leur appui, particulièrement les membres du projet SATT.

RÉSUMÉ

Ce projet s'inscrit dans le cadre de recherches visant à améliorer les mécanismes d'explications que l'on retrouve dans les systèmes experts, les systèmes à base de connaissances ou tout autre système appelé à fournir des explications à un utilisateur.

L'étude de l'évolution des techniques en matière de génération d'explications a permis de choisir une technique grâce à laquelle un mécanisme d'explications est capable de générer un texte explicatif par la réalisation d'un but de communication ayant pour objectif de créer un effet sur l'état mental de l'utilisateur. Cette technique repose sur un algorithme de planification de texte ayant la particularité de conserver à la fois la structure des relations rhétoriques et la structure des intentions d'un texte explicatif. Cela confère à un mécanisme d'explications des capacités de réaction lui permettant d'entretenir un véritable dialogue explicatif avec l'utilisateur. De cette technique est alors émise l'hypothèse selon laquelle un mécanisme d'explication doté de capacités de réaction peut satisfaire un utilisateur en matière d'explication sans avoir recours à un modèle complet et précis de cet utilisateur.

Un prototype a été réalisé de façon à vérifier la faisabilité d'implantation de l'algorithme de planification de texte à mémoire d'intentions, à vérifier les capacités de réaction d'un mécanisme d'explications doté d'un tel algorithme et à valider l'hypothèse émise. Le domaine de connaissances utilisé pour le prototype concerne la tarification de l'électricité à Hydro-Québec.

Ce prototype a démontré qu'il est faisable d'implanter l'algorithme de planification de texte, particulièrement avec un langage comme Prolog. Les capacités de réaction du mécanisme d'explications et l'hypothèse émise ont été vérifiées à l'aide de la participation de cinq utilisateurs à un dialogue explicatif avec le prototype, utilisateurs dont on n'avait aucun modèle.

L'expérience a fait ressortir quelques limites et contraintes davantage imputables aux ressources, comme la base de connaissances explicatives, qu'à l'algorithme de planification de texte lui-même. Des voies de recherche pour pallier à ces limites et contraintes sont d'ailleurs proposées.

ABSTRACT

This research is related to explanations in expert systems, knowledge based systems or any system that must produce explanations for a user.

After a study of the evolution of the techniques used to generate explanations, one of these techniques was chosen. This technique allows an explanation system to produce an explanation by achieving a communicative goal that create an effect on the mental state of the user. The core of this technique is a text planning algorithm. In this algorithm, both rhetorical structure and intentional structure of an explanation are represented. By this, an explanation system is able to react and to participate in explanatory dialogues. An assumption is also made and it state that an explanation system can satisfy a user without a complete and accurate model of this user.

A first prototype has been developed to verify that it is feasible to implement the text planning algorithm. It also served to verify the reactive capabilities of the explanation system and to validate the assumption made. The field of knowledge used was the electricity rates at Hydro-Quebec.

The implementation has demonstrated that it is feasible to implement the text planning algorithm, especially with the Prolog programming language. An experience was also performed with five users who have participated in explanatory dialogues with the system. The conclusion of this experience show that an explanation system composed of a text planning algorithm with intentional structure has reactive capabilities. These reactive capabilities eliminate the need of a complete and accurate user model.

TABLE DES MATIÈRES

DÉDICACE iv

REMERCIEMENTS v

RÉSUMÉ vi

ABSTRACT viii

TABLE DES MATIÈRES ix

LISTE DES TABLEAUX xiii

LISTE DES FIGURES xiv

INTRODUCTION 1

CHAPITRE 1 PROBLÉMATIQUE 4

1.1 Besoins d'Hydro-Québec 4
1.2 Objectif de recherche 6

CHAPITRE 2 REVUE HISTORIQUE 7

2.1 Qualités requises d'un mécanisme d'explications 7
2.2 Trace d'exécution 9
2.3 Texte préfabriqué 11
2.4 Paraphrasage 11

2.5 Problèmes avec les mécanismes de première génération 12
2.5.1 Une représentation à trop bas niveau 12
2.5.2 Des connaissances manquantes 13
2.5.3 Des connaissances de stratégie implicites 14
2.6 Enrichissement de la base de connaissances 15
2.6.1 Ajout de nouveaux types de connaissances 16
2.6.2 Dissociation de la ligne d'explication et de la ligne de raisonnement 17
2.7 Planification de texte 18
2.7.1 Tirer profit de la structure de connaissances 20
2.7.2 Imposition d'une structure 20
2.8 Le fait sur mesure 21
2.8.1 Adapter selon le niveau d'expertise de l'utilisateur 22
2.8.2 Adapter selon les buts et préférences de l'utilisateur 24
2.9 Interaction 25

CHAPITRE 3 PLANIFICATION DE TEXTE 27

3.1 Taxonomie des relations entre les segments de discours 27
3.2 Planification statique de texte: schémas de McKeown 30
3.3 Planification dynamique de texte 31
3.3.1 Planificateur de Hovy 33
3.4 Schéma versus plan de relation 35

CHAPITRE 4 CHOIX D'UN TYPE DE MÉCANISME D'EXPLICATIONS 38

4.1 Critères à considérer 38
4.2 Deux écoles de pensée 41
4.3 Type de mécanisme choisi 43
4.3.1 Préalables à la capacité de réaction et d'adaptation 43

CHAPITRE 5 CHOIX D'UNE MÉTHODE DE PLANIFICATION DE TEXTE 45

5.1 Analyse des premières approches 45
5.2 Problème de représentation des intentions 46
5.3 Planificateur de Moore et Paris 49
5.3.1 Aperçu du processus de planification de texte 51

CHAPITRE 6 MÉCANISME DE PLANIFICATION DE TEXTE 53

6.1 Buts de communication 53
6.2 Buts linguistiques 54
6.2.1 Actes de discours 54
6.2.2 Buts rhétoriques 54
6.3 Opérateurs de planification de texte 55
6.4 Modèle de l'utilisateur 57
6.5 Historique du dialogue 59
6.6 Architecture 60
6.7 Algorithme de planification de texte 61
6.8 Génération du texte explicatif 68
6.8.1 Techniques de génération de texte 69
6.9 Réaction au feed-back de l'utilisateur 70
6.9.1 Réaction suite à une question 71
6.9.2 Réaction suite à une incompréhension 77

CHAPITRE 7 PROTOTYPE 83

7.1 Domaine d'application 83
7.2 Exemples de dialogues 84
7.3 Analyse des dialogues 87
7.3.1 Détermination des intentions et des relations rhétoriques 87

7.4 Implantation 91
7.4.1 Outils, langages et environnement de développement 91
7.4.2 Représentation des connaissances explicatives 93
7.4.3 Représentation des buts de communication et des buts linguistiques . 94
7.4.4 Représentation du modèle de l'utilisateur 96
7.4.5 Représentation de l'historique du dialogue 97
7.4.6 Représentation des opérateurs de planification de texte 98
7.4.7 Génération d'un plan de texte 99
7.4.8 Génération d'un texte explicatif 102
7.4.9 Communication interprocessus 103
7.4.10 Création des questions relatives au texte explicatif 104
7.4.11 Traitement du feed-back de l'utilisateur 105

CHAPITRE 8 ANALYSE DES RÉSULTATS 110

8.1 Type de tests 110
8.2 Constatations 111
8.3 Analyse 114

CHAPITRE 9 CONCLUSION ET VOIES DE RECHERCHE FUTURES 119

BIBLIOGRAPHIE 127

ANNEXE 132

LISTE DES TABLEAUX

Tableau 3.1: Taxonomie des relations de segment de discours 29

Tableau 3.2: Exemples de prédicats rhétoriques 30

Tableau 3.3: Exemples de relations rhétoriques 32

Tableau 3.4: Comparaison Schémas de McKeown versus Plans de relation de Hovy ... 36

Tableau 3.5: Méthodes d'explication en fonction des critères de qualité 37

Tableau 6.1: Algorithme de planification de texte 67

Tableau 7.1: Dialogue humain-humain #1 85

Tableau 7.2: Dialogue humain-humain #2 85

Tableau 7.3: Dialogue humain-humain #3 86

Tableau 7.4: Dialogue humain-humain #4 86

Tableau 7.5: Liaisons intentions/relations rhétoriques 90

Tableau 9.1: Ordonnancement Satellite/Noyau de relations rhétoriques 124

LISTE DES FIGURES

Figure 2.1 Types de connaissances explicatives 16
Figure 2.2 Stéréotypes d'utilisateur 22
Figure 3.1 Exemple d'arborescence d'une planification de texte 34
Figure 5.1 Arbre de planification de texte d'un fragment de dialogue 47
Figure 6.1 Exemple d'opérateur de planification de texte de Moore et Paris 57
Figure 6.2 Architecture du mécanisme d'explications 61
Figure 6.3 Plan de texte et l'explication produite 72
Figure 7.1 Numérotation des noeuds d'un plan de texte 101
Figure 7.2 Texte explicatif et liste de questions 105

INTRODUCTION

Au cours des deux dernières décennies, les systèmes experts sont passés de "simple intérêt scientifique" à une technologie de plus en plus prometteuse. Durant cette évolution, une constante est toutefois demeurée, soit le besoin d'explications des utilisateurs, si bien, que munir les systèmes experts[1] d'un mécanisme capable de produire de bonnes explications est aujourd'hui considéré comme étant aussi important que de fournir la bonne réponse [18]. Étant donné que les systèmes experts sont passés du stade de recherche à leur application dans la vie de tous les jours, le profil de l'utilisateur a lui aussi progressé, passant de l'auteur du système à l'expert du domaine, puis à l'utilisateur débutant.

Les utilisateurs ont donc commencé à demander à ce que les mécanismes d'explications des systèmes experts fournissent des textes explicatifs mieux adaptés à leurs besoins, car on a constaté que la présentation de la trace d'exécution d'un système à titre d'explication demeure très éloignée d'une explication au sens où nous, humains, l'entendons.

Bien que plusieurs travaux de recherche aient fait en sorte que les mécanismes d'explications d'aujourd'hui sont capables de répondre à plusieurs types de questions et aussi de produire des textes explicatifs plus naturels, il n'en demeure pas moins que le

---

1. On inclut également les systèmes à base de connaissances ou tout autre système appelé à fournir des explications à un utilisateur.

problème avec la plupart de ces mécanismes est qu'ils considèrent la production d'une explication comme un processus sans suite [24]. Ainsi, si un système n'a qu'une seule chance de produire un texte explicatif qui rencontre les buts et attentes de l'utilisateur, et ce, sans surcharger ni trop alléger le texte, sans ennuyer ou rendre l'utilisateur confus, alors le système doit posséder une mine d'informations à propos de l'utilisateur.

C'est ce qui a amené bon nombre de chercheurs à penser que la qualité des explications fournies par les systèmes experts s'améliorera le jour où l'on sera capable de construire un modèle complet et précis de l'utilisateur (ses buts, ses connaissances, le niveau de détails qu'il faut lui présenter, etc.). Toutefois, il est à se demander s'il est possible d'obtenir et de gérer un tel modèle de l'utilisateur. De plus, en centrant l'attention sur ce que l'on croyait être une nécessité, on a totalement ignoré le fait que l'utilisateur puisse servir de guide dans l'élaboration des explications. C'est donc en s'éloignant de l'idée de générer des explications sans suite que l'on pourra se servir de l'utilisateur comme guide. En fait, il n'est peut-être pas si important de fournir l'explication parfaite dès le départ. Il faut plutôt faire en sorte que le système puisse ajuster ses explications suivant le feed-back de l'utilisateur, un peu comme le feraient deux interlocuteurs humains.

Le but de la présente recherche est de valider l'hypothèse selon laquelle il est possible de satisfaire l'utilisateur en matière d'explication sans avoir recours à un modèle complet et précis de ce dernier. L'approche retenue consiste à faire participer l'utilisateur à un dialogue explicatif avec le système expert, de profiter du feed-back de l'utilisateur et de s'en servir comme guide pour les explications subséquentes. Inspirée des travaux de

Moore et Paris [19,20,23], la méthode choisie repose sur la théorie de la planification de texte et conduit à l'implantation d'un algorithme de planification de texte à mémoire d'intentions permettant de mémoriser l'effet de chaque composant de l'explication sur l'utilisateur.

Afin de valider cette hypothèse, un prototype a été réalisé. Ce prototype implante l'algorithme de planification de texte à mémoire d'intentions et utilise une base de connaissances dont le domaine de connaissances porte sur la tarification de l'électricité à Hydro-Québec. À cet effet, on a aussi effectué une analyse empirique, analyse dans laquelle on compare les résultats fournis par le prototype en terme de dialogues explicatifs à ceux obtenus lors de simulations entre deux personnes.

CHAPITRE 1
PROBLÉMATIQUE

De nombreux faits démontrent que les entreprises se heurtent à ce que l'on pourrait appeler "le mur de la connaissance". Cela se traduit par des difficultés à acquérir le savoir, à l'assimiler, à le maîtriser et à le produire. Dans le contexte d'Hydro-Québec, il existe des délais souvent importants entre d'une part, le développement des connaissances et du savoir-faire touchant divers domaines tels la tarification et la compatibilité électromagnétique, et d'autre part, leur utilisation par la force de vente, c'est-à-dire, les représentants commerciaux.

Dans la structure actuelle de la diffusion de la connaissance à Hydro-Québec, les représentants commerciaux constituent le point de convergence, vers leurs clients, d'une multitude d'informations présentées sous différentes formes et différents points de vue par différentes personnes (programmeur, chercheur, formateur, etc.). Ce flot d'informations sans structure ni uniformisation peut s'avérer un obstacle au travail du représentant qui n'est alors plus en mesure d'aider son client. Mais plus important encore, les connaissances de l'entreprise se trouvent à être éparpillées dans de nombreux documents et dans la tête de plusieurs personnes susceptibles de quitter l'entreprise à plus ou moins court terme et emportant avec eux une partie du savoir-faire de l'entreprise.

1.1 Besoins d'Hydro-Québec

Compte tenu de ce qui précède, Hydro-Québec a décidé de démarrer un projet

dont les objectifs sont les suivants:

- mettre à la disposition des employés le savoir-faire des experts de l'entreprise;
- faire en sorte que le savoir-faire devienne un actif tangible;
- réduire le délai entre le développement du savoir-faire et son transfert aux employés.

Ce que l'on propose pour y arriver est de faire en sorte que les connaissances cheminent directement de la source de connaissances (expert du domaine) vers l'utilisateur (représentant commercial) avec pour seul intermédiaire un système d'aide au transfert de connaissances.

L'approche de formation classique dans laquelle un formateur diffuse des connaissances à une classe d'élèves ne répond pas aux besoins puisque rien ne garantit que les connaissances diffusées vont réellement être utilisées lors de l'exécution d'une tâche.

En fait, ce dont l'entreprise a besoin, c'est d'un système d'aide au transfert de connaissances qui soutient l'utilisateur, d'une part, dans ses actions en facilitant et en accélérant l'atteinte de résultats de qualité, et d'autre part, dans ses réflexions en lui fournissant toutes les connaissances disponibles d'un domaine de connaissances pour l'aider à analyser, comprendre et planifier ses tâches d'action.

Ce système informatisé d'aide au transfert de connaissances repose essentiellement sur l'utilisation de techniques et méthodes en intelligence artificielle, plus particulièrement les systèmes experts et les sciences cognitives.

1.2 Objectif de recherche

Les besoins exprimés par Hydro-Québec sont à la fois ambitieux et stimulants, et l'idée de vouloir soutenir le représentant commercial dans ses tâches de réflexion offre des défis intéressants.

Le présent projet de recherche cadre dans cet objectif de soutien à la réflexion. En effet, il vise à créer un mécanisme d'explications qui permettrait à un utilisateur d'obtenir des explications quant à la méthode de résolution du système et quant aux connaissances mêmes du domaine d'expertise. Plus précisément, l'objectif de ce travail de recherche est de démontrer que la satisfaction de l'utilisateur en matière d'explication peut être obtenue en le faisant participer à un dialogue explicatif avec le système, et ce, sans que ce dernier possède un modèle précis et complet de cet utilisateur.

CHAPITRE 2
REVUE HISTORIQUE

Avant d'entreprendre la description du mécanisme d'explications implanté ainsi que l'analyse des résultats obtenus, on présente dans un premier temps les qualités requises des mécanismes d'explications et leur évolution en regard de ces critères de qualité.

2.1 Qualités requises d'un mécanisme d'explications

Basés sur l'étude des explications humaines et sur leur propre expérience en matière de conception de systèmes, Moore et Paris [22], de même que Swartout [32] ont dressé une liste d'exigences qui représentent les qualités qu'un bon mécanisme d'explications et les explications qu'il génère devraient posséder. Plusieurs de ces critères de qualité sont aussi corroborés par une étude menée auprès des utilisateurs de systèmes experts [35].

Les explications devraient être:

- **Fidèles**: Les explications doivent correspondre très précisément aux connaissances de la base de connaissances, de même qu'au raisonnement du système pour que les utilisateurs fassent confiance à ces explications.
- **Naturelles**: Les explications, en particulier celles qui comportent plusieurs phrases, doivent être structurées selon des modèles du discours humain pour

produire des textes cohérents et naturels.

- **Sensibles**: Les explications doivent tenir compte des connaissances et des buts de l'utilisateur, de même que du dialogue antérieur.

Quant au mécanisme d'explications, il devrait être ou avoir:

- **Capacités de réaction**: Le mécanisme d'explications doit pouvoir gérer les requêtes sur ses propres explications en répondant à d'autres questions s'y rapportant ou en proposant des explications de rechange si l'utilisateur n'est pas satisfait d'une explication.
- **Flexible**: Le mécanisme d'explications doit pouvoir avoir recours à différentes stratégies d'explication pour répondre à une même question.
- **Suffisant**: Le mécanisme d'explications doit pouvoir répondre à un large éventail de questions grâce à une base de connaissances diversifiée de laquelle il peut extraire les informations nécessaires à la réalisation des réponses aux questions des utilisateurs.
- **Capacités d'adaptation**: Le mécanisme d'explications doit pouvoir modifier ses stratégies ou en créer de nouvelles en se basant sur les interactions avec l'utilisateur.
- **Efficace**: Le mécanisme d'explications ne doit pas affecter les performances du système.
- **Extensible**: Il doit être possible d'ajouter des types de questions ou de

nouvelles stratégies d'explication qui n'ont pas été prévus au moment de la conception initiale.

- **Minimum de surcroît de travail:** L'ajout d'un mécanisme d'explications à un système ne doit pas amener trop de travail de conception supplémentaire, ou du moins, il faudrait pouvoir récupérer l'équivalent de ce travail lors de la maintenance ou de l'évolution du système.

Les sections qui suivent présentent l'évolution des mécanismes d'explications, et chacune des approches préconisées aux diverses étapes de l'évolution sera confrontée aux critères de qualité précédemment mentionnés.

2.2 Trace d'exécution

La plupart des systèmes experts dits de première génération, tel le pionnier MYCIN [27], appartiennent à la catégorie des systèmes experts dont le mécanisme d'explications repose sur l'utilisation de la trace de raisonnement, ou trace d'exécution, pour rendre compte de leur comportement.

La méthode d'explication basée sur la trace d'exécution est principalement utilisée pour montrer la manière dont le système s'y est pris pour arriver au résultat (le comment), ou encore, pour expliquer la raison d'une demande d'information de la part du système (le pourquoi). Ainsi, dans le système MYCIN, la commande COMMENT permettait à l'utilisateur d'explorer en descendant les différentes branches de l'arbre de raisonnement, tandis que la commande POURQUOI permettait à l'utilisateur d'explorer toute la chaîne de raisonnement de bas en haut. Ce type d'explication était très apprécié

des concepteurs du système qui y voyaient une façon de valider leur système. Quant aux utilisateurs en bout de ligne, ce mécanisme d'explications permettait à tout le moins de favoriser l'acceptation des réponses et suggestions du système.

La méthode basée sur l'affichage de la trace d'exécution est simple et elle requiert très peu d'efforts à l'implantation. Cependant, elle possède plusieurs inconvénients non négligeables comme le mentionne Gilbert [6]:

- Étant donné que la trace d'exécution est le résultat des étapes du mécanisme d'inférence, il en résulte souvent une explication difficile à comprendre pour l'utilisateur. Si le raisonnement est complexe, l'utilisateur se voit noyé dans une mer d'informations reflétant le processus de résolution du système.
- Ce type d'explication tend à inclure trop de détails, tels des expressions arithmétiques, et à être répétitif: si une déduction est effectuée plus d'une fois lors du processus d'inférence, elle sera alors présentée autant de fois dans l'explication.
- Le niveau d'abstraction est uniforme en ce sens que l'utilisateur ne peut espérer obtenir qu'une vue d'ensemble car tout lui est présenté.
- Le langage est très éloigné du langage naturel parce que la trace d'exécution est souvent constituée d'une liste de règles du type SI...ALORS.
- Les explications ne tiennent compte en aucun temps des attentes et des connaissances de l'utilisateur.

2.3 Texte préfabriqué

Afin de rendre les explications plus naturelles, certains systèmes emploient la technique des textes préfabriqués. Cette technique consiste à prévoir toutes les questions que l'utilisateur pourrait poser et à encapsuler les réponses dans des gabarits où se marient texte et variables à être instanciées au moment de fournir l'explication.

Bien que cette technique présente l'avantage d'être simple à implanter, elle a le défaut majeur de créer un mécanisme d'explications qui n'est pas fidèle, puisque la base de connaissances et les texte préfabriqués peuvent être modifiés indépendamment l'un de l'autre. Également, cette technique offre peu de flexibilité, puisqu'à chaque question est associée une et une seule réponse pour une situation donnée. On pourrait être tenté, pour donner plus de flexibilité, d'incorporer des éléments caractérisant l'utilisateur et ses réactions, et de représenter chaque cas possible par un texte préfabriqué. Malheureusement, on se bute rapidement à une explosion combinatoire des cas possibles.

2.4 Paraphrasage

De façon à régler le problème de fidélité de l'approche précédente, un autre type de mécanisme d'explications a vu le jour: le paraphrasage du code et de la trace d'exécution. Il s'agit d'une technique dans laquelle les règles sont transcrites en langage naturel par des transformations simples. De cette façon, les explications sont produites à même les connaissances utilisées par le moteur d'inférence, ce qui a pour effet de simplifier la tâche de maintenance et de garantir la fidélité des explications. Cependant, le

fait d'envelopper l'explication avec un langage plus humain qu'informatique ne lui confère pas une allure beaucoup plus naturelle. L'explication demeure, comme avec toutes les techniques présentées jusqu'ici, le reflet de la trace d'exécution qui n'est en soi pas nécessairement ce à quoi l'utilisateur s'attend pour bien comprendre.

2.5 Problèmes avec les mécanismes de première génération

Les mécanismes d'explications présentés jusqu'à maintenant font partie de la première génération et ils dépendent essentiellement de la trace d'exécution. Ils sont très peu suffisants et ils produisent des explications non naturelles, non sensibles et non flexibles en n'offrant aucune capacité de réaction et d'adaptation.

2.5.1 Une représentation à trop bas niveau

L'une des causes des limitations énoncées ci-haut est due au fait que la trace d'exécution est une représentation à trop bas niveau qui ne peut capturer toutes les informations nécessaires et qui ne peut distinguer le rôle joué par chaque type de connaissance dans l'explication [33].

En premier lieu, la trace d'exécution est une représentation à trop bas niveau parce que les règles présentées ne permettent pas à un utilisateur de saisir la stratégie globale de résolution de problème. L'utilisateur pourra saisir la signification de chacune des règles localement, mais pourra difficilement extraire l'approche globale de résolution de problème.

Également, l'utilisateur peut difficilement inférer le rôle de chaque clause

d'une règle parce que ce rôle n'est pas explicitement représenté. Ainsi, il n'est pas possible de différencier les connaissances de contrôle des connaissances de stratégie ou des connaissances de résolution. Par exemple, supposons que l'on exprime la règle suivante pour définir le concept *petite puissance* du domaine de la tarification de l'électricité:

```
Si puissance à facturer < 100 kW
Alors conclure à petite puissance
```

Lors de la génération d'une explication concernant la définition du concept *petite puissance*, cette règle pourra être utilisée pour dire qu'une petite puissance correspond à une puissance à facturer inférieure à 100 kW.

Cependant, en supposant que l'on doive ajouter une clause de contrôle qui tienne compte du fait qu'au nord du 53^{e} parallèle, il ne peut y avoir de petite puissance (fait fictif), on obtient alors la règle suivante:

```
Si puissance à facturer < 100 kW et
Pas au nord du 53e parallèle
Alors conclure à petite puissance
```

La génération d'une explication concernant la définition du concept *petite puissance* inclura alors une mention au fait qu'il ne faut pas être situé au nord du 53^{e} parallèle, ce qui en soi ne devrait pas être relié directement au concept *petite puissance*.

2.5.2 Des connaissances manquantes

Comme autre cause des lacunes des mécanismes d'explications de première génération, on peut mentionner le fait que la formalisation des connaissances sous forme

de règles tend à faire disparaître des connaissances qui pourtant ont été utilisées par l'ingénieur de la connaissance lors de cette formalisation. Par exemple, dans le domaine de la tarification de l'électricité, on a la notion de facteur d'utilisation qui est définie comme suit:

$$FacteurUtilisation = \frac{PuissanceMoyenneDeLaPeriode}{PuissanceMaximaleDeLaPeriode}$$

Supposons que l'ingénieur de la connaissance ait décidé de représenter cette connaissance de la façon suivante:

```
SI PuissanceMaximaleDeLaPeriode est Stable
ET PuissanceMoyenneDeLaPeriode Augmente
ALORS FacteurUtilisation Augmente
```

On s'aperçoit que pour générer une telle règle, il a fallu avoir recours à des connaissances sur l'opérateur de division dans une relation d'égalité, c'est-à-dire, si le dénominateur du quotient reste stable et que le numérateur augmente, alors le quotient augmente. Cette connaissance n'apparaît nullement dans la base de connaissances et est donc irrécupérable pour le mécanisme d'explications.

2.5.3 Des connaissances de stratégie implicites

Dans d'autres cas, on assiste à une dilution des connaissances de stratégie dans les règles de la base de connaissances. Par exemple, une stratégie qui consiste à privilégier

certaines choses avant d'autres peut se refléter dans la base de connaissances par des opérateurs de priorité ou par l'ordre de définition des règles. Dans un tel cas, les connaissances de stratégie ne sont pas représentées explicitement et ne sont donc pas accessibles au mécanisme d'explications.

Ainsi, les mécanismes d'explications de première génération sont incapables d'expliquer le fondement de leurs connaissances et de justifier leur comportement. En effet, les connaissances permettant de générer de telles explications ne sont pas représentées, ou si elles le sont, leur représentation est faite de façon implicite de telle sorte qu'elles sont inaccessibles pour le mécanisme d'explications. Entre autres, l'incapacité de justifier la méthode de résolution du système tient du fait que la connaissance permettant de dire quelle action doit être faite et quand elle doit être faite n'est pas représentée dans le système parce que, n'étant pas utile au processus de résolution de problème, elle a été omise lors de la phase de conception.

2.6 Enrichissement de la base de connaissances

Tant que l'on considérait l'utilisateur comme étant l'un des concepteurs du système, le problème des connaissances manquantes ou implicites ne se posait pas vraiment. En effet, tout ce qui importait pour l'utilisateur était de pouvoir examiner le contenu des règles. Mais face à d'autres types d'utilisateurs, les besoins en matière d'explication ont fait en sorte qu'il devenait nécessaire d'avoir accès à d'autres sources de connaissances dans lesquelles il n'y a pas eu d'inférence, mais à partir desquelles on peut générer des explications pour justifier le comportement du système, expliquer le choix

d'une stratégie, donner des définitions, expliquer des concepts, etc. (voir Figure 2.1).

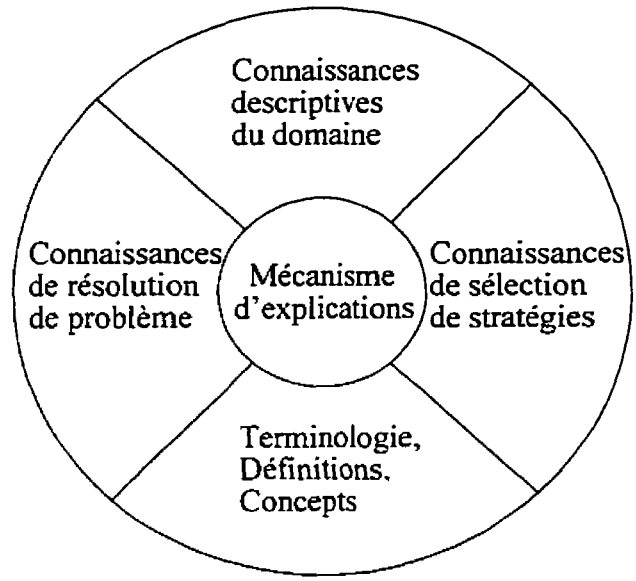


Figure 2.1 Types de connaissances explicatives

Il est clair que pour fournir de bonnes explications en ce qui a trait à la résolution de problème, on a besoin d'une représentation riche en stratégies de résolution de problème ainsi que des connaissances du domaine qui s'y rattachent. Des recherches ont corroboré ces faits et ont démontré la nécessité de représenter explicitement différents types de connaissances d'un domaine [4] et de capturer toutes les connaissances de résolution de problème lors du processus de conception [31]. Ces constatations ont pavé la voie à une seconde génération de mécanismes d'explications plus suffisants.

2.6.1 Ajout de nouveaux types de connaissances

D'une part, il y a des systèmes tels NEOMYCIN [5] et XPLAIN [31] qui

incorporent des méta-connaissances (connaissances sur les connaissances) et des connaissances à différents niveaux d'abstraction pour expliquer le comportement d'un système.

Le système NEOMYCIN, extension de MYCIN, fut l'un des premiers systèmes à considérer des connaissances autres que les connaissances de résolution de problème. Ce système incorpore, sous forme de méta-règles, des connaissances de stratégie qui contrôlent le choix et l'ordre des tâches à exécuter. Cette représentation explicite des connaissances de stratégie permet au système de construire des explications quant à la stratégie de résolution de problème, donc de justifier le comportement du système.

Le système XPLAIN, quant à lui, repose sur une architecture munie d'un générateur de programme qui crée le système expert à la place du programmeur. Durant la création, le générateur de programme enregistre les alternatives rencontrées ainsi que les décisions prises pour constituer un historique du développement. Également, en plus des connaissances de résolution de problème, des connaissances descriptives du domaine y sont représentées. Ainsi, lors de l'exécution du système expert, cet historique du développement, de concert avec la base de connaissances enrichie et la trace d'exécution, permet de présenter à l'utilisateur non seulement ce qu'a fait le système expert mais aussi pourquoi il l'a fait.

2.6.2 Dissociation de la ligne d'explication et de la ligne de raisonnement

D'autre part, il y a des systèmes, tel le système REX (*Reconstructive*

EXplainer) [36], qui eux proposent l'ajout d'une autre base de connaissances pour générer des explications présentées sous un point de vue différent de celui de la résolution de problème. Cela est rendu possible en dissociant la ligne d'explication de la ligne de raisonnement. C'est une approche intéressante qui repose sur le principe selon lequel un expert n'explique pas sa solution de la même façon qu'il a procédé pour la trouver. L'idée ici est que la ligne de raisonnement représente la séquence d'inférences partant des faits vers la solution alors que la ligne d'explication représente la séquence d'inférences constituant l'explication fournie. On considère donc la tâche d'explication comme une tâche en soi, comme une nouvelle inférence à faire sur la ligne de raisonnement permettant d'en montrer les étapes importantes, de la simplifier et de l'adapter à l'utilisateur.

Dans les faits, le système REX procède à une correspondance de la ligne de raisonnement vers des schémas d'explications dans une seconde base de connaissances. Cela permet d'extraire les connaissances explicatives des connaissances d'implantation et de construire à l'aide de la seconde base de connaissances une explication à différents niveaux d'abstraction qui est différente de la ligne de raisonnement.

2.7 Planification de texte

Le fait que certains mécanismes d'explications soient plus suffisants grâce à l'ajout de connaissances explicatives n'en fait pas pour autant des mécanismes capables de générer des explications cohérentes et naturelles.

Pour réussir à générer des textes explicatifs cohérents, naturels et composés de plusieurs phrases, un mécanisme d'explications doit avoir recours à des modèles et

conventions de discours. En fait, il faut que l'explication fournie à l'utilisateur concorde avec l'idée préconçue qu'il a de l'explication à laquelle il s'attend pour que celle-ci semble naturelle. En effet, les gens ont souvent une bonne idée du genre de discours nécessaire pour répondre à leurs attentes. Ainsi, pour produire une explication, un mécanisme d'explications doit être en mesure de sélectionner l'information pertinente pour atteindre un objectif de discours précis et d'organiser cette information pour qu'elle produise un texte cohérent et naturel. C'est ce qu'on appelle la planification de texte.

Comme on l'a mentionné, le problème avec le paraphrasage de la trace d'exécution est que le texte explicatif qui en résulte est le reflet du mode de raisonnement du moteur d'inférence du système expert. Or, il est juste de croire qu'une explication doit pouvoir refléter autre chose pour être bien comprise: l'utilisateur et le système expert n'utilisent pas toujours les mêmes connaissances ni le même mode de raisonnement pour résoudre un problème. Bien sûr, l'enrichissement des bases de connaissances par des connaissances explicatives a permis d'étendre le spectre des types de questions auxquelles les mécanismes d'explications peuvent répondre, mais ce n'est pas ce qui garantit d'obtenir des explications cohérentes et naturelles. Le fait de dissocier la ligne d'explication de la ligne de raisonnement ne le garantit pas non plus. C'est pourquoi, les responsables des travaux de recherche portant sur les mécanismes d'explications des systèmes experts ont commencé à s'intéresser aux techniques utilisées et aux résultats obtenus dans les domaines de la linguistique par ordinateur et du traitement en langage naturel. Entre autres, ils ont constaté qu'il existe essentiellement deux types de textes explicatifs: ceux dont la structure est étroitement liée à la structure des connaissances du

domaine de connaissances et ceux dont l'organisation n'est pas propre au domaine de connaissances.

2.7.1 Tirer profit de la structure de connaissances

Il existe des types de textes explicatifs pour lesquels la structure des connaissances explicatives correspond à la structure d'une explication cohérente et naturelle pour ces connaissances, par exemple, les relations cause à effet. Dans ce cas, l'une des meilleures façons de produire une explication est d'utiliser la technique de la traversée de graphe [30]. Cette technique utilise des procédures qui déterminent les liens à suivre dans la base de connaissances pour produire une explication, un peu comme le fait la technique du paraphrasage. Ainsi, non seulement la trace d'exécution peut être parcourue, mais également des réseaux sémantiques représentant des concepts de la base de connaissances peuvent aussi l'être. Donc, l'idée maîtresse de la technique de la traversée de graphe est que le parcours d'un ensemble déterminé de liens produit une explication cohérente et naturelle à la condition que les connaissances du domaine soient représentées de la même façon dont les gens parlent entre eux de ces connaissances. Il est évident que cela restreint les stratégies d'explication possibles, car elles se doivent d'être à l'image de la structure des connaissances du domaine, et c'est pourquoi d'autres travaux ont porté sur des techniques indépendantes d'une telle structure.

2.7.2 Imposition d'une structure

Pour certains types de textes, le contenu du texte explicatif doit être construit

à partir d'une structure qui n'est pas inhérente à la base de connaissances. Dans ces cas, les textes doivent être produits à partir de considérations du discours humain. Il faut rechercher les informations pertinentes dans la base de connaissances et ensuite les structurer de façon cohérente pour produire l'effet voulu sur l'utilisateur. Ce type de planification de texte est au coeur du présent travail de recherche et c'est pourquoi le prochain chapitre lui est entièrement consacré.

Mais au delà du fait qu'il existe maintenant de meilleures architectures (architectures qui font en sorte que le processus de résolution de problème est plus explicite, que le raisonnement de conception est capturé et que plusieurs types de connaissances y sont représentés) et que l'on emploie des méthodes de planification de texte (méthodes qui font que les explications sont plus cohérentes et plus naturelles), un problème important demeure: les explications que ces systèmes fournissent ne correspondent pas vraiment aux types d'explications que les gens ont entre eux. En fait, les explications produites par ces systèmes experts peuvent être qualifiées de '*one shot, fits all*' [33] ou, traduction libre, *du premier coup, pour tous. Du premier coup* parce qu'elles ne permettent pas aux utilisateurs de poser des questions sur l'explication elle-même, l'explication fournie étant une fin en soi. *Pour tous*, car elles ne tiennent pas compte de l'utilisateur, ce qu'il connaît, ses buts, ses préférences, etc. Les prochaines sections abordent ces points.

2.8 Le fait sur mesure

Il va de soi qu'une explication sera vraisemblablement mieux comprise si elle

est adaptée aux connaissances de l'utilisateur et si elle prend en compte ses buts et ses préférences. Deux principales approches ressortent de différents travaux afin de munir les systèmes d'un mécanisme d'explications plus sensible, à savoir, l'adaptation des explications en fonction de l'expertise de l'utilisateur et aussi celle en fonction de ses buts.

2.8.1 Adapter selon le niveau d'expertise de l'utilisateur

La première approche est la plus populaire et elle consiste à créer des stéréotypes d'utilisateurs basés sur leur niveau d'expertise (Figure 2.2) et à marquer les faits et règles de la base de connaissances afin d'indiquer à quel niveau d'expertise ils sont associés.

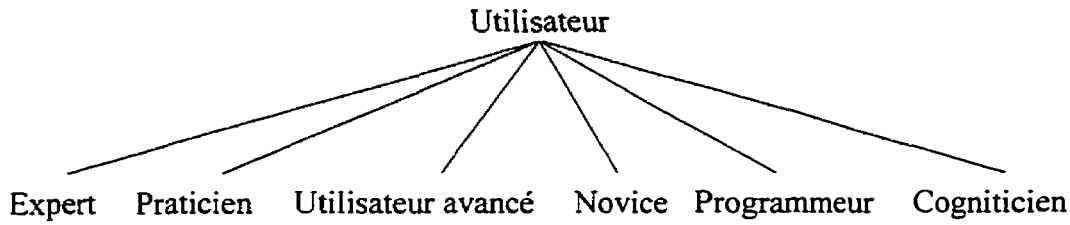


Figure 2.2 Stéréotypes d'utilisateur

À titre d'exemple, Wallis et Shortliffe [34] ont développé un système dans lequel un facteur de complexité de 1 à 10 est associé à chaque règle d'inférence et à chaque concept, ainsi qu'une valeur d'importance à chaque concept. Lors de la génération d'une explication, le facteur de complexité des règles d'inférence et des concepts est comparé au niveau d'expertise de l'utilisateur qui sert alors de limite supérieure à la complexité des concepts pouvant faire partie de l'explication.

Cette technique semble simple à première vue, parce que d'une part, il est facile de réaliser un module informatique pour utiliser et gérer le marquage dans la base de connaissances, et parce que d'autre part, il appartient à l'utilisateur d'indiquer son niveau d'expertise. Ainsi, le système ne requiert pas de mécanisme complexe pour inférer le niveau d'expertise de l'utilisateur.

Cependant, le marquage peut être une tâche longue et ardue, et il peut nécessiter une grande connaissance du domaine. Mais l'inconvénient majeur de cette technique est qu'elle repose sur l'hypothèse que l'expertise de l'utilisateur est uniforme sur l'ensemble du domaine de connaissances, ce qui n'est pas nécessairement le cas. Ainsi, une explication est plus ou moins adaptée si cette explication fait intervenir des connaissances pour lesquelles l'utilisateur est novice et d'autres pour lesquelles il est expert.

À cet effet, les travaux de Paris [25] se sont attaqués au rôle de la connaissance de l'utilisateur dans la génération de textes explicatifs. L'objectif visé par ces travaux était de permettre la génération de textes adaptés à des utilisateurs ayant des niveaux de connaissances très variés, et ce, sans avoir recours à un ensemble de stéréotypes définis a priori. Donc, au lieu d'assumer que l'utilisateur a un profil qui correspond à l'un des stéréotypes (novice, expert, etc.), on considère que l'utilisateur possède des expertises du domaine de connaissances dites locales. Donc, l'utilisateur peut être considéré comme un expert pour certaines parties du domaine de connaissances et novice pour d'autres.

De plus, il est non seulement important de considérer la quantité

d'informations contenues dans un texte explicatif mais également le type de ces informations. Ainsi, selon les connaissances de l'utilisateur pour une partie donnée du domaine de connaissances, différentes stratégies d'explication peuvent être utilisées. Il s'agit certes d'une approche qui permet la génération d'explications mieux adaptées à l'utilisateur que l'approche par stéréotypes d'expertise, mais cette approche requiert l'acquisition et la gestion d'un modèle de l'utilisateur qui reflète ce qu'il connaît du domaine de connaissances.

2.8.2 Adapter selon les buts et préférences de l'utilisateur

D'autres diront qu'une bonne explication est une explication qui sait prendre en compte les buts et préférences de l'utilisateur. Par exemple, McKeown [17] a développé le système ADVISOR qui sert à fournir l'information nécessaire pour conseiller un étudiant dans son choix de cours. Dans ce système, l'utilisateur, qui est un étudiant, peut avoir différents buts lors de la sélection de ses cours. Il peut vouloir rencontrer les exigences de son programme de cours le plus rapidement possible, il peut vouloir choisir des cours pour suivre le programme à la lettre, ou bien il peut choisir les cours qu'il trouve les plus intéressants pour un domaine d'intérêt donné. Selon le but de l'utilisateur, le système produit des explications qui en tiennent compte. Par exemple, si le but de l'utilisateur est de terminer le plus rapidement possible son programme de cours, le système pourra justifier le choix d'un cours en signalant que ce cours est un préalable à plusieurs autres cours.

Mais pour être en mesure de fournir des explications qui tiennent compte des

buts de l'utilisateur, le système doit les connaître dès le départ. Cela peut se faire en associant des buts par défaut aux différents stéréotypes d'utilisateurs. Autrement, le système doit être capable de les inférer durant l'exécution. Cette dernière façon de faire semble plus appropriée puisque les buts de l'utilisateur peuvent varier beaucoup et subitement durant un échange avec le système, contrairement à une modélisation selon l'expertise où l'expertise ne varie pas rapidement.

2.9 Interaction

Jusqu'à maintenant, les mécanismes d'explications présentés n'offrent aucune capacité de réaction. Un mécanisme d'explications ayant des capacités de réaction est un mécanisme capable de gérer les requêtes sur ses propres explications en répondant à d'autres questions s'y rapportant ou en proposant des explications de rechange si l'utilisateur n'est pas satisfait d'une explication.

La capacité de réaction est une caractéristique importante d'un mécanisme d'explications puisqu'elle permet de considérer la génération d'explications comme un processus coopératif entre le système et l'utilisateur plutôt que comme un processus unidirectionnel où l'utilisateur fournit des données et le système fournit des réponses. D'ailleurs, Karsenty et Brézilon [12] mentionnent que l'étude des dialogues humains entre un expert et un débutant montre que les deux interlocuteurs doivent négocier à la fois le problème à résoudre ainsi que la solution afin que le débutant puisse bien comprendre.

Également, un mécanisme d'explications pourvu de capacités de réaction permet de contrebalancer le fait qu'un modèle précis et complet de l'utilisateur ne soit pas

disponible. La génération d'une explication bonne du premier coup, c'est-à-dire, qui satisfait dès le départ aux attentes de l'utilisateur, n'est alors plus une obligation. À la place, le système peut se fier à l'utilisateur pour que ce dernier lui fournisse le feed-back nécessaire afin d'éclaircir certains points de son explication ou carrément produire une explication de rechange.

Mais pour ce faire, un mécanisme d'explications doit être en mesure de comprendre ses propres explications. Lorsque l'utilisateur ne comprend pas totalement une explication, le mécanisme d'explications doit être capable de déterminer la partie du texte explicatif qui n'a pas été comprise et d'élaborer sur cette partie. Cela ne peut se faire que si le mécanisme d'explications garde en mémoire l'information nécessaire sur l'intention de chacune des parties du texte explicatif et la manière dont ces intentions sont atteintes. De cette façon, le mécanisme d'explications peut raisonner sur son propre texte explicatif et déterminer une façon de remédier au fait qu'une explication ou partie d'explication n'ait pas été comprise.

CHAPITRE 3
PLANIFICATION DE TEXTE

Comme il a été mentionné au chapitre précédent, la génération de textes explicatifs cohérents, naturels et composés de plusieurs phrases nécessite l'emploi d'une technique de planification de texte. Ce chapitre présente deux des principales techniques de planification de texte: les schémas de McKeown et les plans de relation de Hovy.

3.1 Taxonomie des relations entre les segments de discours

D'après Hovy [8], on peut affirmer ceci à propos de la structure du discours ou des textes, et ce, indépendamment du langage et du type de texte:

- Un discours ou un texte est une structure regroupant un ensemble de segments composés d'autres segments ou de propositions. La structure du discours s'exprime donc par une imbrication de segments selon des relations spécifiques. Un discours peut alors être représenté par une structure arborescente dans laquelle chaque noeud gère le sous-arbre, c'est-à-dire, le segment sous lui. Au niveau le plus haut, le discours est géré par un seul noeud-racine. Quant aux feuilles de l'arbre, elles représentent des segments non décomposables contenant des propositions servant à produire le texte.
- À chaque segment du discours est associé un objectif de discours. Cet objectif est présent dans chaque noeud. Un discours atteint son but si chaque segment réussit son objectif de discours.

Ainsi donc, de nombreux auteurs, dont Mann et Thompson avec leur théorie de la structure rhétorique [14], se sont alors mis à la tâche d'identifier l'ensemble des relations pouvant exister entre les segments d'un discours. Ces relations sont également appelées relations rhétoriques. De ces recherches, Hovy [8] en a fait ressortir une taxonomie comportant trois catégories principales: au niveau des idées, interpersonnelle et textuelle[1]. Le Tableau 3.1 à la page suivante présente les relations les plus communes de chacune des catégories.

Les relations au niveau des idées permettent d'exprimer les connaissances que l'on a à propos d'un domaine de connaissances donné. Les relations de ce type n'ont besoin que d'une base de connaissances assez riche en connaissances du domaine pour être réalisées. Aucune connaissance à propos de l'utilisateur n'est requise.

Les relations interpersonnelles sont des relations par lesquelles on tente de créer un effet sur les croyances, les attitudes, les désirs, etc. de la personne à laquelle s'adressent ces intentions. L'effet ainsi créé est d'amener à croire, ou de convaincre, de motiver, de rendre capable, etc. Puisque cela a un lien étroit avec les connaissances, croyances, désirs et intérêts d'une personne, ce type de relation est généralement utilisé de concert avec un modèle de l'utilisateur.

Quant aux relations textuelles, ce sont des relations qui relient deux segments de texte dont la relation n'est ni au niveau des idées, ni interpersonnelle. Il s'agit plutôt

---

1. Mann et Thompson [14] ont identifié deux catégories, "*presentationnal*" et "*subject matter*", qui correspondent respectivement aux catégories suivantes de Hovy: interpersonnelle et au niveau des idées.

d'une relation qui n'est là que pour supporter ou renforcer le mode de présentation: et, ou, premièrement, deuxièmement, etc.

Tableau 3.1: Taxonomie des relations de segment de discours

Au niveau des idées	Élaboration	Objet-Attribut
		Objet-Fonction
		Ensemble-Membre
		Processus-Étape
		Général-Spécifique
		Définition
		Exemple
		Conclusion
		Résumé
	Circonstancielle	Lieu
		Temps
		Manière
	Cause/Effet	
	Condition	
	Exception	
	Illustration	
	Comparative	Équivalence
		Contraste
		Comparaison
		Analogie
Interpersonnelle	Interprétation	
	Support	Préalable
		Évidence
		Justification
		Motivation
	Habilitation	
Textuelle	Conjonction	
	Disjonction	
	Présentation-Séquence	
	Joint	

3.2 Planification statique de texte: schémas de McKeown

Une première méthode de planification de texte fut développée en se basant sur l'analyse d'explications apparaissant dans des documents (encyclopédie, manuel de formation, etc.). Cette analyse a permis de constater qu'il existe des prédicats rhétoriques indépendants du domaine de connaissances et qui caractérisent la structure des textes cohérents et naturels. Les prédicats rhétoriques sont les moyens utilisés pour dire les choses. Le Tableau 3.2 présente quelques exemples de prédicats rhétoriques.

Tableau 3.2: Exemples de prédicats rhétoriques

Prédicats rhétoriques	
définition logique	"Un client est une personne, une société, une corporation ou un organisme, titulaire d'un ou de plusieurs abonnements."
constitution	"La structure tarifaire comporte des tarifs de base et des tarifs optionnels."
contrainte	"Vous ne pouvez pas changer votre puissance souscrite avant 12 mois."

Par exemple, McKeown [16] a analysé la structure de différents textes explicatifs sur la description d'entités et en a dégagé des organisations de prédicats rhétoriques qui suivent un certain ordre. Ainsi, l'auteure a montré qu'un texte explicatif pouvait être décomposé en une structure hiérarchique de contenus explicatifs formés de prédicats rhétoriques dans laquelle les prédicats rhétoriques sont utilisés comme moyen

pour décrire une information. Par exemple, faire une analogie, décrire les parties constituantes et donner une définition sont autant de façons de décrire un objet, mais via trois prédicats rhétoriques différents.

Un schéma de McKeown est un canevas constitué d'un certain nombre de prédicats rhétoriques pouvant être instanciés de façon sélective. La génération d'une explication est donc guidée par la structure des schémas. Les schémas peuvent aussi être récursifs puisque chaque prédicat d'un schéma peut mener soit vers un contenu textuel, soit vers un autre schéma. La structure résultante du parcours d'un schéma est donc une arborescence dans laquelle chaque sous-arbre représente un prédicat menant à un autre prédicat, et où chaque feuille de l'arbre représente un prédicat menant à un contenu textuel. Des informations supplémentaires, telles le nombre de fois que le prédicat peut apparaître dans le paragraphe ou bien le fait qu'un prédicat peut être optionnel, sont également incluses dans le schéma. Pour utiliser un schéma lors de la génération d'un texte explicatif, il suffit de parcourir le schéma en instanciant chacun des prédicats terminaux (noeud-feuille de l'arborescence) par le contenu approprié extrait de la base de connaissances, puis de fournir ce contenu à un générateur de texte[1].

3.3 Planification dynamique de texte

Les schémas de McKeown permettent une planification de texte que l'on pourrait qualifier de statique en ce sens que toute l'organisation d'un texte est

1. L'utilisation de textes préfabriqués peut s'avérer une solution de rechange pour le cas où il n'y a pas de générateur de texte. Cependant, le texte explicatif peut s'avérer moins naturel en surface, principalement en ce qui a trait aux charnières reliant les phrases.

prédéterminée par des schémas explicatifs qui décrivent un agencement cohérent de différents contenus explicatifs.

Une autre approche consiste à créer un agencement cohérent de contenus explicatifs de façon dynamique. Pour cela, on a recours à des relations rhétoriques. Les relations rhétoriques sont des abstractions sur la façon dont les parties adjacentes d'un texte sont reliées l'une à l'autre. Le Tableau 3.3 illustre quelques exemples de relations rhétoriques.

Tableau 3.3: Exemples de relations rhétoriques

Relations rhétoriques	
motivation	"Vous devriez choisir le tarif G. La facture sera moins élevée!"
contraste	"Le tarif G est admissible. Par contre, le tarif M ne l'est pas."
condition	Vous pouvez opter pour le tarif G. Pour cela il faut que votre puissance soit inférieure à 100 kW."
moyen	"J'essaie de calculer le montant de la facture par le calcul, entre autres, du coût de l'énergie."

Ainsi, un texte cohérent est un texte dans lequel tous les contenus explicatifs adjacents sont reliés par des relations rhétoriques.

3.3.1 Planificateur de Hovy

Hovy [7] fut le premier à mettre en opération les relations rhétoriques en se fondant sur les travaux de Mann et Thompson. Dans cette mise en opération, chaque relation rhétorique est représentée par un plan de relation composé d'un noyau et de zéro ou plusieurs satellites. Le noyau représente la partie du texte essentielle à l'objectif de discours alors que les satellites viennent renforcer l'effet du noyau. Par exemple, soit la phrase suivante:

> Vous devriez choisir le tarif G. La facture sera moins élevée!

En supposant que l'objectif de discours est de faire en sorte que la personne choisisse le tarif G, la première partie du texte représente le noyau et la seconde le satellite. Dans ce cas-ci, la relation rhétorique présente entre le noyau et le satellite est une relation de motivation: la partie satellite vient motiver (*ça va coûter moins cher*) le fait de faire telle action (*choisir le tarif G*).

Chaque texte explicatif peut alors être représenté par une structure composée d'un ensemble de plans de relation, le plus commun de ces plans de relation étant composé d'une partie noyau et d'une partie satellite reliées par une relation rhétorique. Les plans de relation sont également récursifs puisque leur noyau ou leurs satellites peuvent eux-mêmes être décrits par un plan de relation qui peut à son tour être décomposé, et ainsi de suite jusqu'à l'atteinte d'unités terminales (un noyau ou un satellite) qui ne se décomposent pas.

Seules les unités terminales servent à produire du texte, bien que certaines

unités non terminales peuvent également en produire à des fins de charnière de discours (telles que 'parce que', 'comme', 'contrairement', etc.) pour expliciter la relation rhétorique qu'elles représentent. La Figure 3.1 donne un exemple d'arborescence résultant d'une planification de texte accompagné du texte explicatif produit.

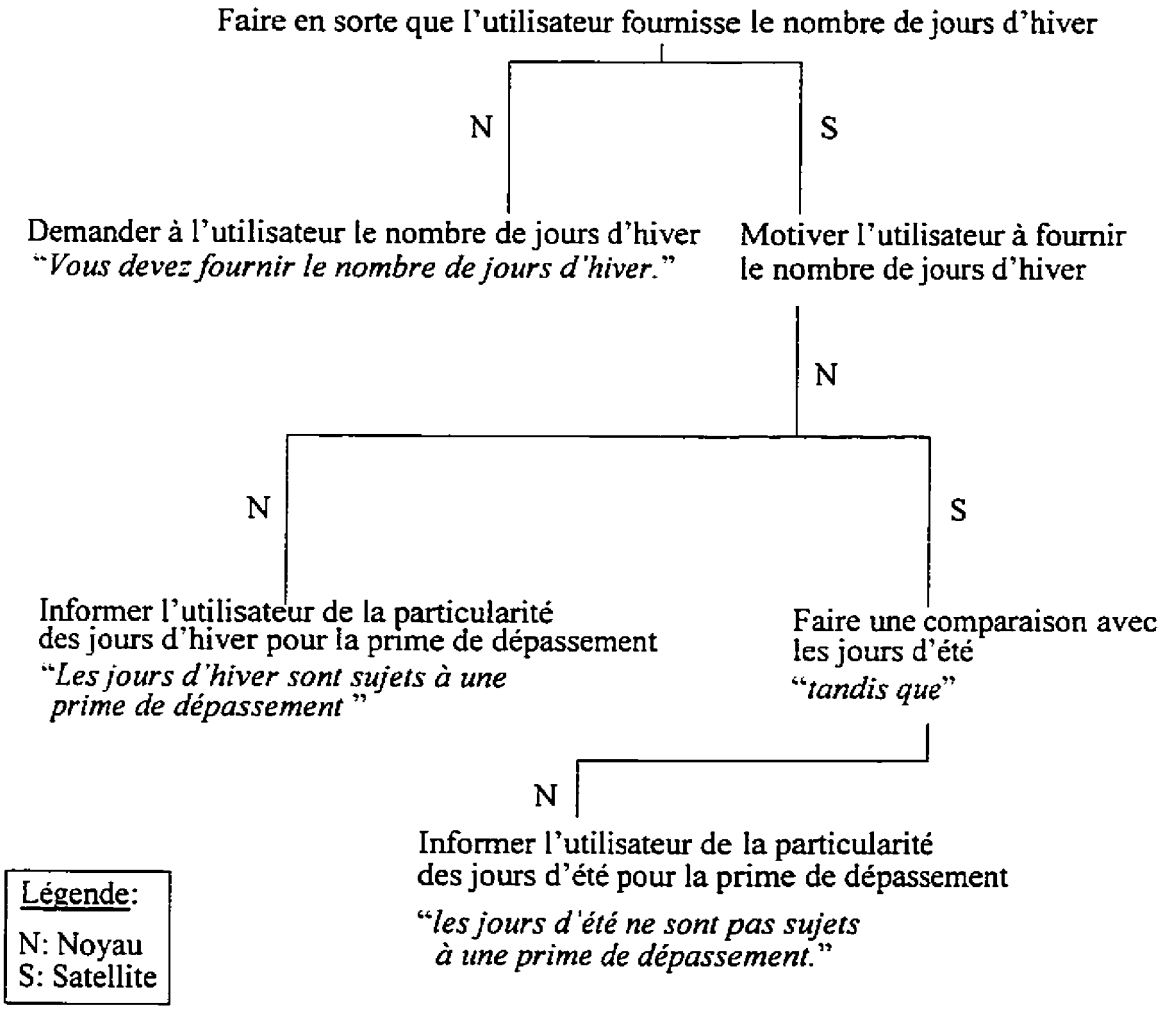


Figure 3.1 Exemple d'arborescence d'une planification de texte

3.4 Schéma versus plan de relation

Les sections précédentes ont présenté deux méthodes de planification de texte: les schémas de McKeown et les plans de relation de Hovy. Chacune de ces méthodes possède des avantages et des inconvénients. Voyons ce qu'il en est.

Les schémas ont l'avantage d'être faciles à construire et à utiliser. Il suffit, pour une application donnée, de définir un schéma pour chaque type de texte explicatif. Cette structure statique leur confère toutefois peu de variabilité. Les plans de relation, eux, ont une structure plus dynamique et permettent donc plus de flexibilité.

Un autre désavantage des schémas est qu'ils ne contiennent pas le rôle rhétorique de chacune des parties d'un texte explicatif. Cela peut être un inconvénient majeur pour tout système devant adapter ses explications dynamiquement face aux réactions de l'utilisateur. Les plans de relation n'ont pas ce problème puisque la structure arborescente construite lors d'une planification de texte contient la relation rhétorique entre chacune des parties du texte explicatif qui en résulte.

Par contre, la structure des schémas explicatifs étant définie d'avance, cela leur confère l'avantage d'être plus efficace en terme d'exécution par rapport à une planification dynamique avec les plans de relation. En effet, il est plus difficile, donc plus coûteux en temps d'exécution, de créer l'arbre du texte explicatif à partir de relations indépendantes que ce ne l'est pour instancier un schéma.

Le Tableau 3.4 résume la comparaison entre les schémas de McKeown et les plans de relation de Hovy.

Tableau 3.4: Comparaison Schémas de McKeown versus Plans de relation de Hovy

	Schémas de McKeown	Plans de relation de Hovy
Structure	statique	dynamique
Flexibilité	peu flexible	très flexible
Temps d'exécution	plus efficace	moins efficace
Structure rhétorique	absente	présente
Capacité de réaction	faible	bonne

En somme, il est intéressant de constater que la gestion des schémas explicatifs et la gestion des plans de relation permettent actuellement à un système de produire des explications composées de plusieurs phrases organisées sous forme d'un texte cohérent.

Le Tableau 3.5 représente une synthèse de l'étude de l'évolution des diverses méthodes d'explication que l'on a présentées en fonction des critères de qualité énumérés à la section 2.1, *"Qualités requises d'un mécanisme d'explications"*. On constate entre autres que des méthodes d'explications permettent maintenant de concevoir des mécanismes d'explications plus suffisants, flexibles et extensibles, dotés de capacités de réaction et d'adaptation. En contrepartie, cela amène un surcroît de travail lors de la conception des systèmes. De même, plus le mécanisme d'explications est complexe, plus l'efficacité des systèmes peut en être effectée.

Tableau 3.5: Méthodes d'explication en fonction des critères de qualité

	Explication			Mécanisme d'explications						
Méthodes	Fidèle	Naturelle	Sensible	Capacité de Réaction	Flexible	Suffisant	Capacité d'adaptation	Efficace	Extensible	Surcroît de travail
Trace d'exécution	Oui	Non	Non	Aucune	Non	Non	Aucune	Excellent	Non	Aucun
Texte préfabriqué	Non	Peu[1]	Non	Aucune	Non	Non	Aucune	Excellent	Oui	Moyen
Paraphrasage	Oui	Peu[1]	Non	Aucune	Non	Non	Aucune	Excellent	Non	Faible
Ajout de types de connaissances	Non[3]	Potentiel[2]	Non	Non	Potentiel[2]	Oui	Potentiel[2]	Très bon	Oui	Moyennement élevé
Planification de texte statique	Non[3]	Oui	Non	Faible	Peu	Oui	Oui	Très bon	Oui	Élevé
Planification de texte dynamique	Non[3]	Oui	Non	Bonne	Beaucoup	Oui	Oui	Moyen	Oui	Élevé
Le fait sur mesure	Non[3]	Non	Oui[4]	Non	Non	Oui	Non	Moyen[5]	Non	Élevé[5]
Avec structure des intentions	Non[3]	Oui	Oui	Excellente	Oui	Oui	Oui	Moyen	Oui	Élevé

(1) Ça demeure le reflet de la trace d'exécution.

(2) Doit être combinée avec une autre méthode capable d'exploiter les différents types de connaissances.

(3) Nécessite un support supplémentaire pour assurer la fidélité des explications.

(4) Dépend de la qualité du modèle de l'utilisateur.

(5) Dépend de la complexité du modèle de l'utilisateur.

CHAPITRE 4
CHOIX D'UN TYPE DE MÉCANISME D'EXPLICATIONS

La revue historique présentée au chapitre précédent permet d'aiguiller notre choix quant au type de mécanisme d'explications à choisir dans le cadre de ce projet de recherche.

4.1 Critères à considérer

Bien sûr, il serait souhaitable d'implanter un mécanisme d'explications ayant toutes les qualités d'un bon mécanisme d'explications énumérées au chapitre 2, à savoir:

des explications;

- Fidèles
- Naturelles
- Sensibles

et un mécanisme d'explications;

- Flexible
- Suffisant
- Extensible
- Efficace
- Requérant un minimum de surcroît de travail
- Avec capacités de réaction
- Avec capacités d'adaptation

Cependant, il faut considérer l'objectif du présent travail de recherche qui consiste à développer un mécanisme d'explications permettant à l'utilisateur d'obtenir des explications sur la méthode de résolution du système et sur les connaissances mêmes d'un domaine d'expertise (en l'occurrence la tarification de l'électricité à Hydro-Québec). De plus, il ne suffit pas d'être capable de fournir de telles explications, il faut aussi que l'utilisateur en soit satisfait. Il y a donc parmi cette liste de qualités, des qualités à favoriser par rapport à d'autres. Il ne faut pas oublier qu'il s'agit ici d'effectuer un transfert de connaissances, donc de s'assurer que l'utilisateur s'approprie de nouvelles connaissances ou consolide celles déjà acquises.

Ainsi, la nécessité d'avoir un mécanisme d'explications efficace en terme d'exécution et qui requiert un minimum de surcroît de travail pour les concepteurs des systèmes experts ne cadre pas directement dans l'objectif à atteindre. C'est pourquoi, l'optimisation des algorithmes et processus de traitement de même que l'implantation d'une méthode qui tient compte de la tâche du cogniticien ne feront pas ici l'objet d'une grande attention.

Le mécanisme d'explications devra cependant être suffisant de façon à ce que l'utilisateur puisse obtenir des explications non seulement sur la méthode de résolution du système mais également sur les connaissances mêmes du domaine d'expertise. Les connaissances explicatives devront donc être explicitement représentées.

Cet aspect de suffisance engendre cependant une autre préoccupation, soit celle de la fidélité. Puisque l'on favorise l'existence de deux catégories de connaissances

dans une même base de connaissances (connaissances explicatives et connaissances servant au moteur d'inférence pour la résolution des problèmes), il faut s'assurer de la concordance entre ces deux catégories de connaissances. Il faut donc que les textes explicatifs fournis correspondent à ce que le système expert fait réellement. On pourrait bien sûr s'attarder à concevoir un mécanisme capable d'extraire les deux types de connaissances à même une seule source (tel le générateur de programme XPLAIN de Swartout) de façon à assurer la fidélité. Cependant, il ne s'agit pas ici d'un des éléments essentiels à l'atteinte de l'objectif du présent travail. Bien sûr, la fidélité des explications importe, mais c'est la façon dont on va s'y prendre pour l'assurer qui l'est moins. Ainsi, dans un premier temps, on relayera cette responsabilité au concepteur du système qui devra s'assurer de la concordance entre les deux catégories de connaissances.

On s'assurera de créer un mécanisme d'explications flexible et extensible de façon à permettre la génération de textes explicatifs selon différentes stratégies d'explication ainsi que l'accroissement de son potentiel explicatif par l'ajout de nouvelles stratégies d'explication.

Il faudra également que les textes explicatifs générés soient naturels, donc cohérents et reflétant la structure du discours humain. En ce sens, il faudra être capable de générer des textes explicatifs composés de plusieurs phrases liées par des relations rhétoriques et reflétant autre chose que la trace d'exécution.

Finalement, l'un des critères importants à considérer sera de concevoir un mécanisme d'explications sensible. Pour ce, il faut un mécanisme qui tienne compte de

l'utilisateur de façon à satisfaire les attentes de l'utilisateur en matière d'explication. Ce critère de sensibilité est au coeur de deux axes de recherche différents.

4.2 Deux écoles de pensée

Selon Carcagno [2], la valeur explicative d'une réponse n'est pas une qualité intrinsèque au texte, mais elle est plutôt accordée par l'interlocuteur en fonction de ses buts, ses attentes et ses connaissances. La question qui se pose maintenant est de savoir s'il est possible, à partir d'une requête de l'utilisateur, de justement connaître ses buts, ses attentes et ses connaissances et ainsi déterminer ce que doit contenir une explication. Cela entraîne nécessairement de poser des hypothèses sur ce à quoi l'utilisateur s'attend en matière d'explication. Mais comment valider ces hypothèses? On a déjà mentionné que l'utilisation d'un modèle de l'utilisateur permet de représenter ses connaissances et ses buts. Ainsi, il serait possible de valider les hypothèses dès le départ, mais encore faut-il que ce modèle soit précis et complet. Mais on pourrait également valider les hypothèses, une fois l'explication fournie, en analysant les réactions de l'utilisateur. En effet, en permettant au système et à l'utilisateur d'avoir un véritable dialogue entre eux, il est alors possible pour l'utilisateur de négocier l'explication qu'il désire. On considère alors que l'utilisateur est le seul juge de la qualité de l'explication. Donc, il faut choisir: valider les hypothèses a priori ou a posteriori?

De façon à faciliter le choix, voici les observations qui sont ressorties d'une analyse de dialogues entre deux humains (un expert-conseiller et un requérant) effectuée par Moore et Paris [23]:

- Il arrive fréquemment que le requérant ne comprenne pas totalement une explication de l'expert. L'expert doit donc dans ces cas définir certains termes qu'il a employés ou fournir d'autres connaissances préalables à la compréhension de l'explication.
- Le requérant pose souvent des questions dans le but d'obtenir des éclaircissements, d'autres détails ou carrément une autre explication.
- Il est fréquent que le requérant ne sache pas ce qu'il ne comprend pas. Il a souvent des difficultés à formuler une question claire: il hésite, répète l'explication de l'expert ou dit tout simplement qu'il n'a pas compris.
- L'expert n'a pas un modèle complet et précis du requérant. Au mieux, il peut lui assigner un stéréotype au début de la conversation, puis raffiner son modèle tout au long du dialogue.

Considérant que: 1- l'une ou l'autre des approches de modélisation de l'utilisateur (selon le niveau d'expertise de l'utilisateur ou selon ses buts) suppose une parfaite connaissance de l'utilisateur, car autrement les explications produites ne seraient pas totalement adaptées; 2- la conception et la gestion d'un modèle parfait et complet de l'utilisateur ne sont pas réalistes [28]; 3- rarement une explication satisfait entièrement son destinataire du premier coup (selon les observations précédemment mentionnées), alors on en arrive au constat suivant: le mécanisme d'explications doit posséder des capacités de réaction et d'adaptation.

4.3 Type de mécanisme choisi

En fonction de ce qui précède, on peut affirmer que le mécanisme d'explications à réaliser devra être *flexible* et *extensible*, et capable de générer des explications *naturelles* construites à partir de connaissances explicatives explicitement représentées dans la base de connaissances.

On favorise également un mécanisme d'explications possédant des *capacités de réaction et d'adaptation* pour satisfaire l'utilisateur plutôt qu'un mécanisme basé uniquement et entièrement sur un modèle de l'utilisateur. Ce qui veut dire qu'il faut être capable d'instaurer un véritable dialogue entre le système et l'utilisateur. Cela n'empêche en rien d'utiliser un modèle de l'utilisateur incomplet, ne serait-ce que pour s'assurer de ne pas dire ou répéter des choses que l'utilisateur connaît déjà.

4.3.1 Préalables à la capacité de réaction et d'adaptation

On peut supposer que toute personne qui désire fournir une explication à une autre personne a des buts ou des intentions qui servent à créer un effet sur cette autre personne. Pour y arriver, elle doit nécessairement avoir recours à des ressources linguistiques et sélectionner les bonnes.

Selon Moore [19], si l'on désire être capable de faire participer un ordinateur à un dialogue explicatif avec un utilisateur, l'ordinateur doit être capable de raisonner sur son propre discours explicatif, c'est-à-dire, comprendre ses propres explications. Cela requiert d'implanter une méthode de planification de texte capable de conserver

l'information reflétant les intentions de l'ordinateur et la manière dont ces intentions ont été réalisées. En ayant cela, l'ordinateur sera en mesure d'analyser son propre discours explicatif pour déterminer son prochain texte explicatif dans le cas où une explication ou une partie d'explication n'a pas bien été comprise par l'utilisateur.

CHAPITRE 5
CHOIX D'UNE MÉTHODE DE PLANIFICATION DE TEXTE

L'objectif du présent chapitre est de choisir une méthode de planification de texte qui permet de conserver les informations nécessaires pour qu'un mécanisme d'explications utilisant une telle méthode soit doté de capacités de réaction.

5.1 Analyse des premières approches

Comme il a été mentionné, l'une des premières approches de planification de texte utilisait les schémas explicatifs de McKeown. Bien que les schémas explicatifs offrent une certaine souplesse en permettant l'ajout de prédicats rhétoriques optionnels pouvant être instanciés selon le contexte, selon ce qui a déjà été dit et selon ce que l'utilisateur connaît, leur utilité est limitée par un désavantage important, soit le manque de représentation du but de chacune des parties d'un texte explicatif. En effet, les schémas explicatifs n'ont aucune représentation explicite de l'effet de chacune des parties d'un texte sur l'utilisateur et sans cette information, il est impossible d'expliquer à nouveau une partie de texte non comprise, ni de justifier pourquoi telle chose a été dite. Ce qui fait qu'en l'absence de toute structure des buts de chacune des parties d'un texte explicatif, un mécanisme d'explications ne peut espérer offrir des capacités de réaction et d'adaptation. En effet, une fois son explication produite, il n'est pas en mesure de raisonner sur celle-ci puisqu'il n'a aucune information lui permettant de savoir pourquoi une telle partie de texte ou un tel segment de discours existe dans l'explication. Tout système devant supporter un

dialogue explicatif avec l'utilisateur doit pouvoir raisonner sur ses propres explications, et à cet égard, les schémas explicatifs s'avère inadéquats.

Le développement de méthodes de planification de texte dynamiques, comme celle employée par le planificateur de Hovy et présentée au chapitre 3, a permis de démontrer l'applicabilité de la théorie de la structure rhétorique de Mann et Thompson pour la production de textes explicatifs cohérents. Ces méthodes permettent de mémoriser le rôle rhétorique de chacun des segments d'un texte explicatif. Cela leur confère jusqu'à un certain point des capacités de réaction, puisqu'à chaque relation rhétorique peut être déduit un but ou une intention. Par exemple, à une relation rhétorique de motivation peut être associée une intention de persuasion de faire quelque chose. On dit bien jusqu'à un certain point seulement car comme on va le voir à la prochaine section, il y a un problème à considérer que les ensembles de relations rhétoriques et d'intentions sont en relation bijective.

5.2 Problème de représentation des intentions

Bien que les plans de relation permettent de représenter chaque relation rhétorique employée pour chacun des segments d'un texte explicatif, il subsiste toutefois un problème que l'on peut illustrer à l'aide d'un exemple.

Soit le fragment de dialogue suivant et l'arbre de planification de texte correspondant (Figure 5.1):

Système: Vous devez fournir votre numéro de compte. Prenez votre facture. Le numéro de compte y est indiqué. OK?

Utilisateur: Non!

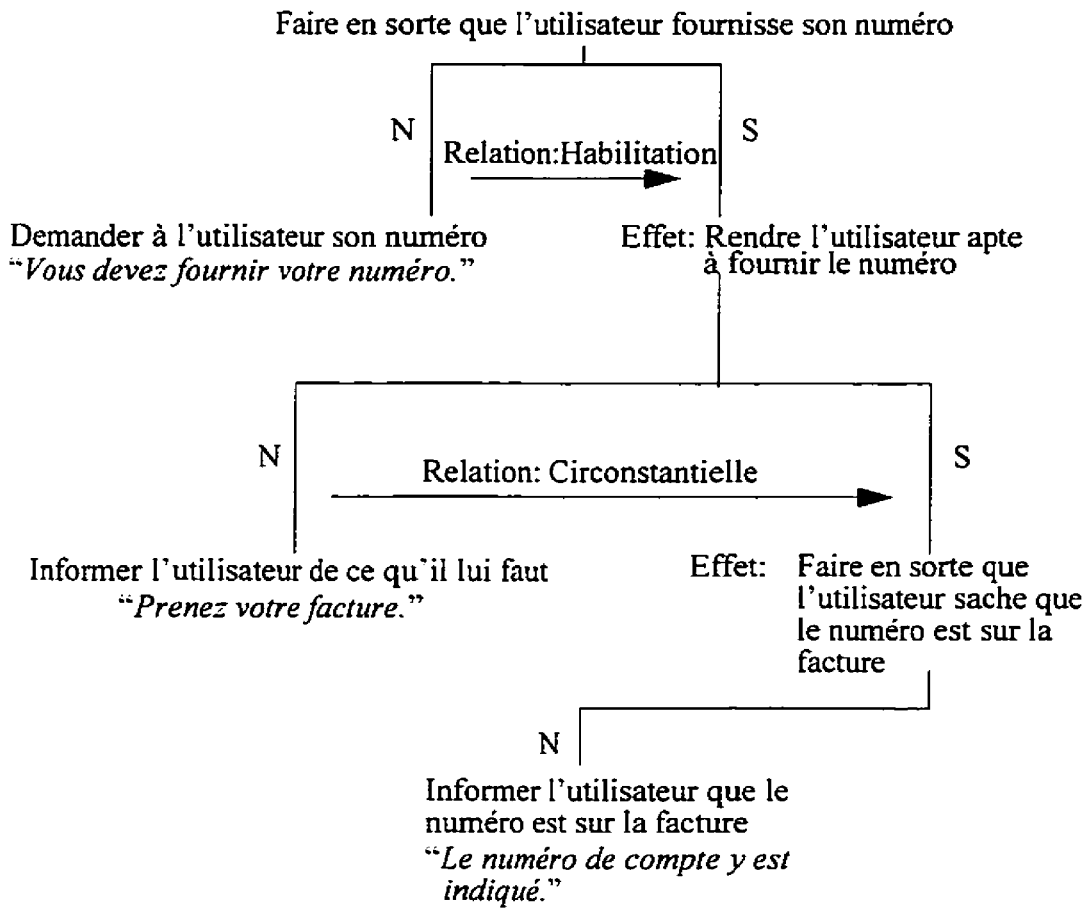


Figure 5.1 Arbre de planification de texte d'un fragment de dialogue

Dans ce dialogue, le système demande d'abord à l'utilisateur de fournir son numéro de compte. Pour l'aider, le système emploie une relation d'*habilitation* en lui indiquant que sa facture peut lui être utile. Puis, l'expert termine en utilisant une relation *circonstancielle* (de lieu) pour mentionner que le numéro de compte est sur la facture. Chacune de ces relations rhétoriques est également accompagnée d'un effet à créer sur l'utilisateur.

Comment le système peut-il réagir suite à la réponse négative de l'utilisateur

signifiant qu'il n'est pas capable de fournir son numéro? Partant de l'hypothèse que l'utilisateur ne peut fournir son numéro parce qu'il ne le trouve pas sur la facture (l'autre hypothèse serait que l'utilisateur ne trouve tout simplement pas sa facture), le système doit être capable d'identifier le segment du texte explicatif qui est en cause. Il s'agit ici du texte généré par le satellite de la relation circonstancielle. Cette tâche d'identification est ardue, mais supposons qu'il en soit capable. Que peut-il faire alors? Il peut essayer de trouver d'autres plans de relation qui ont le même effet (faire en sorte que l'utilisateur sache que le numéro est sur la facture), mais qui sont accomplis différemment. Il peut aussi utiliser d'autres informations circonstancielles au sujet du numéro de compte, mais il est fort à parier que ces autres informations circonstancielles ne sont pas pertinentes (par exemple, depuis quand le numéro existe).

Le problème ici est qu'il est difficile pour le système de réagir en réponse à l'utilisateur en ne sachant pas pourquoi l'information circonstancielle est présentée. Ce qu'il manque dans la planification de texte de la Figure 5.1, c'est une information indiquant l'intention derrière la présentation de la relation circontancielle. Dans ce cas-ci, on peut affirmer que cette intention en est une visant à permettre l'identification du numéro de compte par l'utilisateur. Connaissant cette intention, il serait alors possible pour le système d'utiliser une autre relation rhétorique ayant la même intention. Par exemple, une relation *élaboration-objet-attribut* (le numéro de compte commence par les quatre premières lettres de votre nom) ou une relation *comparative-contraste* (le numéro de compte n'est pas celui apparaissant en haut à gauche de la facture).

On peut constater que, contrairement aux relations rhétoriques

interpersonnelles, les relations rhétoriques au niveau des idées ne sont pas en relation bijective avec les intentions. Les relations rhétoriques interpersonnelles étant des relations par lesquelles on tente de créer un effet sur l'utilisateur, il est facile de voir qu'à chaque relation de ce type correspond une et une seule intention et vice-versa. Par exemple, à une intention de persuasion du bien fondé d'une action de la part du système correspond une relation rhétorique de justification, et l'inverse est vrai.

Dans le cas des relations rhétoriques au niveau des idées, ceci n'est pas vrai puisqu'en général il existe différentes relations rhétoriques pour réaliser une intention. Par exemple, si l'intention est de s'assurer que l'utilisateur connaisse un concept, on peut utiliser des relations rhétoriques circonstancielles, comparatives, d'élaboration, etc. Inversement, une relation rhétorique peut servir à diverses intentions. Par exemple, une relation rhétorique comparative peut être utilisée pour définir un nouveau concept, donner les différences entre deux concepts, convaincre l'utilisateur d'un choix, etc.

Pour qu'un mécanisme d'explications puisse avoir des capacités de réaction et d'adaptation, il ne suffit donc pas de conserver la structure rhétorique du texte explicatif. Il faut également conserver la structure des intentions.

5.3 Planificateur de Moore et Paris

Comme on le sait maintenant, un bon moyen pour un système d'offrir des capacités de réaction et de participer à un dialogue explicatif avec l'utilisateur est d'être en mesure de comprendre ses propres explications. Pour y parvenir, il doit pouvoir

raisonner sur les intentions incluses dans le texte explicatif, non seulement pour le but global de l'explication, mais également pour chaque sous-but de l'explication. Il doit en plus pouvoir connaître les moyens rhétoriques employés pour atteindre ces buts.

Le planificateur de texte de Moore et Paris [21] est le seul parmi un ensemble de planifacateurs de texte (Hovy [7], Maybury [15], Cawsey [3], Suthers [30]) à considérer le problème de réaction et d'adaptation en dissociant structure rhétorique et structure des intentions. La plupart des planificateurs de texte sont dépourvus de toute capacité de recouvrement local, c'est-à-dire, pouvoir fournir une autre explication sur une partie de texte explicatif non comprise ou qui demande des éclaircissements. Leur mécanisme de recouvrement est plutôt global en ce sens que leur seul moyen de réaction, lorsqu'une explication ou une partie d'explication n'est pas comprise (bien que leur problème réside justement dans l'impossibilité d'identifier cette partie d'explication non comprise), consiste à employer une autre stratégie pour réaliser le but de discours de tête, c'est-à-dire, le noeud-racine de l'arbre de planification de texte.

En plus de représenter la structure rhétorique et la structure des intentions lors de la planification d'une explication, le planificateur de texte de Moore et Paris mémorise chaque hypothèse faite à propos des buts et connaissances de l'utilisateur. Ceci est important puisqu'étant donné que le système repose sur un modèle de l'utilisateur qui n'est ni complet ni précis, il doit faire de telles hypothèses de temps à autre, de façon à pouvoir utiliser ses stratégies d'explication (une stratégie pouvant nécessiter que l'utilisateur connaisse un concept donné par exemple). En mémorisant ces hypothèses, elles deviennent alors disponibles plus tard pour le cas où une partie du texte explicatif reposant

sur l'une des hypothèses n'aurait pas été comprise. Le système peut alors décider de rendre vraie une hypothèse selon laquelle l'utilisateur connaît un concept donné en élaborant sur ce concept.

5.3.1 Aperçu du processus de planification de texte

Le processus de planification de texte s'amorce avec un but de communication résultant soit d'un traitement effectué par le système, soit d'une requête de l'utilisateur. Ce but de communication indique l'objectif du texte explicatif à présenter à l'utilisateur. Lorsqu'un but de communication est envoyé au planificateur de texte, celui-ci consulte sa librairie de stratégies pour déterminer lesquelles s'appliquent au but de communication à réaliser. Ces stratégies sont encodées sous forme d'opérateurs de planification de texte qui ont sensiblement la même fonction que les plans de relation de Hovy, mais leur structure diffère quelque peu. Des contraintes, sur la base de connaissances ou sur le modèle de l'utilisateur, sont également associées aux stratégies et sont utilisées comme filtre de sélection lorsque plusieurs stratégies sont candidates. Le choix d'une stratégie peut mener à décomposer l'opérateur de planification de texte correspondant, et la planification se poursuit alors jusqu'à ce que chaque sous-but ne se décompose plus. Le produit de cette décomposition est appelé *plan de texte*. La méthode de décomposition hiérarchique employée ici est très semblable à la méthode de décomposition hiérarchique que l'on a présentée à la section 3.3, *"Planification dynamique de texte"*.

Pendant le processus de planification de texte, on mémorise toutes les hypothèses à propos de ce que l'utilisateur connaît, de même que toutes les stratégies de

rechange qui auraient pu être choisies à un noeud (sous-but) donné.

Une fois la planification de texte terminée, celle-ci est envoyée à un générateur de texte pour produire le texte explicatif à présenter à l'utilisateur. Ensuite, le système attend un feed-back de l'utilisateur. Ce feed-back peut nécessiter l'envoi d'un nouveau but de communication au planificateur de texte si ce feed-back indique que l'utilisateur désire poser une question en rapport avec le texte explicatif qui vient de lui être présenté, ou s'il indique que l'utilisateur n'a pas compris l'explication. L'envoi de ce nouveau but de communication déclenche alors le début d'un nouveau processus de planification de texte qui cette fois nécessite la consultation de ce qu'il convient d'appeler l'historique du dialogue. En effet, cet historique du dialogue contient les plans de texte produits durant le dialogue avec l'utilisateur qui peuvent être consultés pour déterminer de quelle façon répondre lorsqu'une explication n'est pas comprise. En sachant quelle stratégie a été employée et dans quel but, le mécanisme d'explications est à même de déterminer une stratégie de rechange pouvant accomplir le but ayant échoué.

Le mécanisme d'explications du présent travail de recherche s'inspire du planificateur de texte de Moore et Paris. Le coeur de ce planificateur de texte est régi par un algorithme de planification de texte qui permet au système de participer à un dialogue explicatif avec l'utilisateur puisqu'il a cette particularité que l'on a mentionnée: il représente explicitement l'intention de chaque segment ou partie de l'explication. C'est pourquoi on peut qualifier cet algorithme d'algorithme à mémoire d'intentions. L'algorithme est aussi hiérarchique parce qu'il décompose chaque but de haut niveau en sous-buts de plus bas niveau, le résultat étant une structure arborescente.

CHAPITRE 6
MÉCANISME DE PLANIFICATION DE TEXTE

L'algorithme de planification de texte est au coeur même de la méthode de planification de texte adoptée. Il utilise des opérateurs de planification de texte qui servent à accomplir les buts du planificateur de texte. De façon à obtenir des plans de texte qui représentent explicitement les structures d'intentions et des relations rhétoriques, on répertorie les buts selon deux classes: les buts de communication et les buts linguistiques.

6.1 Buts de communication

Les buts de communication sont représentés en terme d'effets que le système entend créer sur les croyances de l'utilisateur (ce qu'il sait, connaît), de même que ses buts (ce qu'il désire faire, atteindre). Voici quelques exemples:

- Persuader l'utilisateur de faire telle chose.
- S'assurer que l'utilisateur connaît certaines choses à propos de tel concept.
- S'assurer que l'utilisateur croit telle proposition.

Les buts de communication présents dans les plans de texte ne génèrent aucun texte. Seuls les buts linguistiques ont cette propriété. À noter que le noeud de tête (ou noeud-racine) d'un plan de texte est toujours un but de communication.

6.2 Buts linguistiques

Les buts linguistiques correspondent aux moyens linguistiques disponibles au système pour lui permettre de réaliser les buts de communication. Ils servent à produire le texte de l'explication et ils constituent des actes de discours ou bien des buts rhétoriques.

6.2.1 Actes de discours

Les actes de discours sont des buts linguistiques à partir desquels sont produits directement les segments de texte d'une explication. Parmi les actes de discours, on dénote l'acte *informer*, l'acte *demander* et l'acte *recommander*. Par exemple:

- Informer l'utilisateur à propos de quelque chose.
- Demander à l'utilisateur de faire une action.
- Recommander à l'utilisateur de faire tel choix.

Dans un plan de texte, les actes de discours représentent les noeuds-feuilles de ce plan. Par conséquent, les actes de discours ne sont pas décomposables.

6.2.2 Buts rhétoriques

Les buts rhétoriques sont des buts linguistiques servant à établir les liens rhétoriques entre les segments d'un texte explicatif. Par exemple:

- Décrire un concept en donnant une définition (élaboration par définition).
- Décrire un concept en spécifiant ses attributs (élaboration par attributs).

- Énumérer les différences avec un concept connu (comparaison par contraste).

Les buts rhétoriques, contrairement aux actes de discours, ne sont pas réalisés directement. Ils sont décomposables en d'autres sous-buts, ces sous-buts étant soit des buts de communication, soit des actes de discours.

Le noeud-racine d'un plan de texte étant représenté par un but de communication et les noeuds-feuilles par des actes de discours, les buts rhétoriques sont donc des noeuds internes au plan de texte (mais pas nécessairement tous les noeuds internes puisque d'autres buts de communication peuvent émaner d'un but rhétorique).

6.3 Opérateurs de planification de texte

Pour réaliser les buts de communication et les buts linguistiques, qui sont en fait les noeuds du plan de texte d'une explication, le planificateur de texte a recours à un ensemble d'opérateurs appelés *opérateurs de planification de texte*. Ces opérateurs ont une structure qui leur confère un rôle de décideur pour l'algorithme de planification de texte. En effet, les opérateurs de planification de texte contiennent suffisamment d'informations leur permettant de déterminer si une stratégie d'explication donnée peut s'appliquer dans un contexte donné. Concrètement, les stratégies d'explication sont représentées par les buts rhétoriques encodés dans des opérateurs de planification de texte. Voici donc ce que contient la structure de ces opérateurs:

- Un effet: sert à caractériser le but recherché par l'opérateur. Il peut s'agir d'un effet de communication, émanant d'un but de communication, ou d'un effet linguistique, émanant d'un but linguistique. Les effets de

communication sont exprimés en termes d'état mental de l'utilisateur à atteindre. Par exemple, un effet de communication peut être du genre "faire en sorte que l'utilisateur soit persuadé de devoir faire telle chose", ou encore, "faire en sorte que l'utilisateur connaisse l'existence de tel concept". Pour ce qui est des buts linguistiques, on peut mentionner que "établir une évidence pour la proposition qui vient d'être mentionnée", "élaborer un concept" ou encore "informer l'utilisateur de quelque chose", en sont quelques exemples.

- Des contraintes: faits qui doivent exister dans la base de connaissances explicatives ou dans le modèle de l'utilisateur pour que l'opérateur puisse être utilisé. La liste de contraintes peut également être vide.
- Un noyau: sous-but nécessaire à l'atteinte de l'effet à créer par l'opérateur.
- Un satellite: sous-but qui contribue à l'atteinte de l'effet à créer par l'opérateur. Il vient renforcer le propos tenu au niveau du noyau. Le satellite n'est pas obligatoire. Aussi, s'il en existe un, il peut être optionnel, auquel cas le mécanisme d'explications l'ignore s'il est dans un mode où il doit produire des textes explicatifs laconiques. Si par contre il est en mode verbeux, le satellite ne sera pas ignoré.

La Figure 6.1 présente un exemple d'opérateur de planification de texte de Moore et Paris [21].

```
EFFECT:
    (PERSUADED ?hearer (DO ?hearer ?act))
CONSTRAINTS:
    (AND  (GOAL ?hearer ?goal))
    (STEP ?act ?goal)
NUCLEUS:
(FORALL ?goal (MOTIVATION ?act ?goal))
SATELLITE:
nil
```

Figure 6.1 Exemple d'opérateur de planification de texte de Moore et Paris

Cet opérateur contient l'information quant au fait que le but de communication du système visant à persuader l'utilisateur (*hearer*) de faire une action (*act*) peut être réalisé en motivant cette action en termes d'un but (*goal*) de l'utilisateur. Cet opérateur contient deux contraintes. La première stipule que l'utilisateur doit avoir un but, et la seconde, que l'action que l'utilisateur doit poser est une étape pour atteindre ce but, chacune de ces contraintes étant respectivement une contrainte sur le modèle de l'utilisateur et une contrainte sur la base de connaissances explicatives.

6.4 Modèle de l'utilisateur

Bien qu'il ait été dit que l'acquisition et la gestion d'un modèle de l'utilisateur complet et précis soit irréaliste (section 4.2, *"Deux écoles de pensée"*), il faut tout de

même considérer qu'un modèle de l'utilisateur, aussi restreint soit-il, s'avère un élément à ne pas négliger lors des échanges avec l'utilisateur, ne serait-ce que pour s'assurer de ne pas redéfinir inutilement des concepts que l'utilisateur connaît déjà par exemple. Mais également, le modèle de l'utilisateur peut être fort utile pour déterminer la meilleure stratégie à employer là où plusieurs stratégies sont candidates. Chacune de ces stratégies, qui se transposent en fait à un opérateur de planification de texte, peut être confrontée aux autres par le biais des contraintes de l'opérateur auquel elles sont associées. Ainsi, on pourrait privilégier une stratégie, donc un opérateur, pour laquelle toutes les contraintes en rapport avec le modèle de l'utilisateur sont satisfaites.

Mais il ne faut pas tomber dans le piège de rejeter toutes stratégies possédant une contrainte du modèle de l'utilisateur non satisfaite. Cette façon de faire tend à considérer le modèle de l'utilisateur comme étant précis et complet. Il faut plutôt exploiter le fait que le mécanisme d'explications a des capacités de réaction et qu'il sera en mesure de satisfaire l'utilisateur si la stratégie employée ne porte pas fruit.

Le modèle de l'utilisateur peut contenir principalement six types d'information [23]:

- Les concepts que l'utilisateur connaît, c'est-à-dire, des concepts dont il connaît les attributs, les fonctions, etc. Par exemple, l'utilisateur connaît le concept *tarif M* ainsi que le fait que ce tarif n'a pas de frais de redevance, a une puissance souscrite, etc.
- Les concepts dont l'utilisateur a une idée, c'est-à-dire, des concepts dont

l'utilisateur connaît une description, mais ne connaît pas nécessairement ses attributs, fonctions, etc. Par exemple, l'utilisateur connaît l'existence du tarif M, que c'est un tarif général, mais sans plus.

- Des faits que l'utilisateur croît, comme par exemple, des relations entre les concepts: le fait que par exemple les jours d'hiver, contrairement aux jours d'été, sont sujets à une prime de dépassement.
- Les buts de l'utilisateur. Par exemple, l'utilisateur veut connaître les tarifs applicables dans son cas.
- Les connaissances de l'utilisateur en matière de méthode pour atteindre ses buts. Par exemple, l'utilisateur sait qu'il doit déterminer son appel maximal de puissance pour l'aider à connaître les tarifs applicables.
- Les connaissances de l'utilisateur en matière de méthode pour faire une action. Par exemple, l'utilisateur sait comment calculer le nombre de jours d'hiver d'une période de consommation.

6.5 Historique du dialogue

L'historique du dialogue est une composante du mécanisme d'explications servant à conserver ce que les deux interlocuteurs du dialogue se sont dit. Il contient donc les requêtes de l'utilisateur ainsi que les réponses du système. Loin d'être limité à ne conserver que le texte d'une réponse, l'historique du dialogue contient en fait chaque plan de texte généré par le planificateur de texte. Ceci est important quand vient le temps pour

le mécanisme d'explications d'interpréter une question de l'utilisateur. Il consulte alors l'historique du dialogue de façon à pouvoir déterminer à quoi se rapporte la question. Et puisque chaque plan de texte contient à la fois la structure des intentions et la structure des relations rhétoriques, le mécanisme d'explications a accès à toute l'information nécessaire pour réagir.

6.6 Architecture

Avant de présenter le fonctionnement de l'algorithme de planification de texte, il serait bon de schématiser l'ensemble des composants nécessaires au fonctionnement de cet algorithme. Ces composants sont, en plus des deux participants (le système et l'utilisateur): l'analyseur des demandes, le planificateur de texte, les opérateurs de planification de texte, le modèle de l'utilisateur et l'historique du dialogue. La Figure 6.2 présente l'architecture du mécanisme d'explications dans laquelle interagissent tous ces composants.

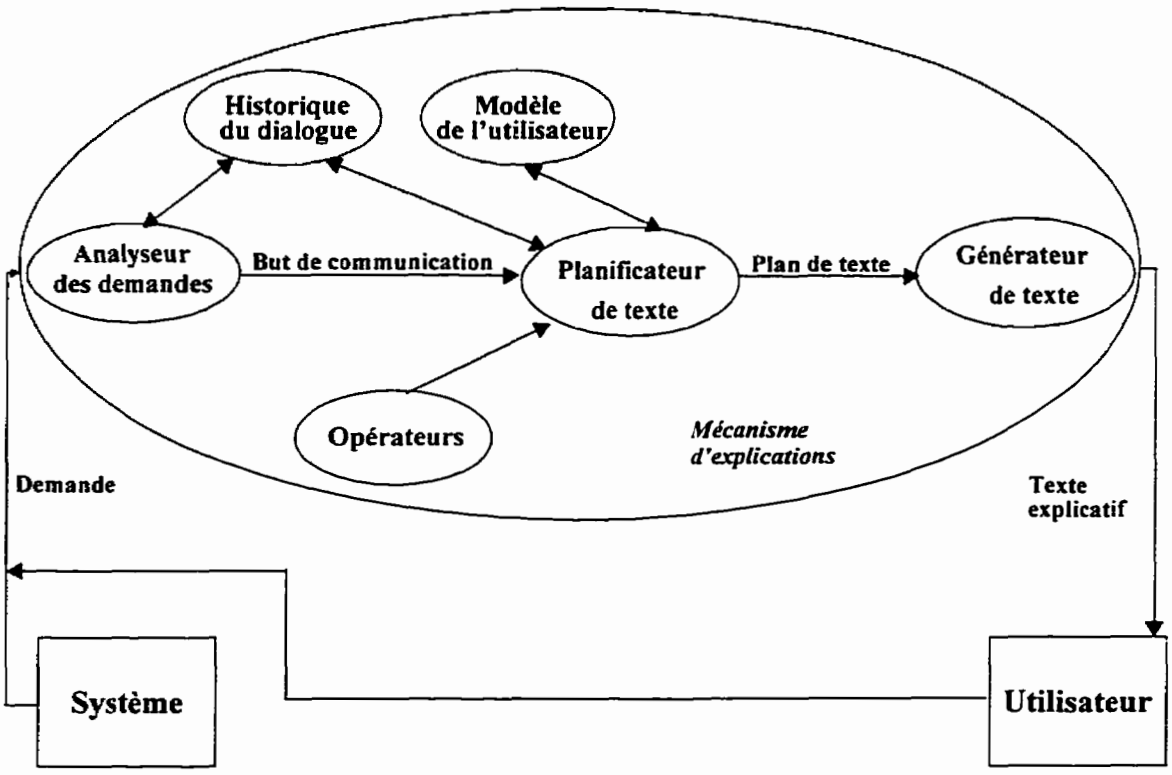


Figure 6.2 Architecture du mécanisme d'explications

Les modalités d'interaction entre les différents composants sont exposées à la section suivante où l'algorithme de planification de texte est présenté.

6.7 Algorithme de planification de texte

L'algorithme de planification de texte à mémoire d'intentions entre en scène au moment où une requête, parvenant de l'utilisateur ou du système, est envoyée au mécanisme d'explications, plus précisément à son analyseur des demandes. Chacune des requêtes est convertie par l'analyseur des demandes en un but de communication. Rappelons qu'un but de communication est représenté en termes d'effet que le système

entend créer sur l'état mental de l'utilisateur. Par exemple, si l'analyseur des demandes reçoit du système une requête visant à demander à l'utilisateur de faire une action, le but de communication qui en résulte devra amener l'utilisateur à avoir pour but d'effectuer cette action. Il en est de même si la requête provient de l'utilisateur. Par exemple, si la requête de l'utilisateur est une question du type *Qu'est-ce que tel concept?*, le but de communication résultant devra faire en sorte que l'utilisateur connaisse des choses à propos de ce concept.

Le but de communication produit par l'analyseur des demandes est ensuite acheminé au planificateur de texte, dont la responsabilité est de produire un plan de texte. La première étape de la construction d'un plan de texte consiste à identifier tous les opérateurs de planification de texte dont l'effet correspond au but de communication. Il est effectivement possible, et cela est souhaitable, qu'il existe plus d'un opérateur qui satisfasse ce critère. Par exemple, un but de communication ayant pour objectif d'amener l'utilisateur à avoir certaines connaissances à propos d'un concept peut être réalisé soit par un opérateur qui élabore les attributs de ce concept, soit par un opérateur qui compare ce concept à un concept connu de l'utilisateur, ou encore par un opérateur qui élabore par un exemple, etc.

Tous les opérateurs retenus suite à leur confrontation au but de communication représentent des candidats potentiels pour faire partie du plan de texte. Cependant, pour devenir de véritables candidats, chacune de leurs contraintes doit être satisfaite. Par exemple, l'une des contraintes de l'opérateur d'élaboration d'un concept par un exemple est qu'il doit exister dans la base de connaissances explicatives un exemple se rapportant

à ce concept. S'il un tel exemple n'existe pas, l'opérateur est alors rejeté comme candidat. À l'inverse, il se peut qu'un même opérateur puisse produire plus d'un candidat, car le fait de vérifier les contraintes de cet opérateur amène à instancier les variables contenues dans les contraintes. Dans le cas de l'opérateur d'élaboration d'un concept par un exemple, il se peut très bien qu'il existe plus d'un exemple pour un concept donné, créant ainsi autant de candidats qu'il y a d'exemples. Il est important de remarquer que déjà, lors de la vérification des contraintes d'un opérateur, le contenu du texte explicatif commence à se préciser puisque l'instanciation des variables des contraintes effectue une sélection d'informations dans la base de connaissances explicatives.

On se rappellera que les contraintes d'un opérateur sont de deux types: les contraintes sur la base de connaissances explicatives et les contraintes sur le modèle de l'utilisateur. Bien qu'un opérateur soit rejeté par le fait qu'une contrainte sur la base de connaissances explicatives n'est pas satisfaite, il en va tout autrement en ce qui a trait aux contraintes sur le modèle de l'utilisateur. Étant donné que l'on considère le modèle de l'utilisateur comme étant ni complet ni précis, il serait absurde de rejeter un opérateur sur le simple fait qu'une ou plusieurs de ses contraintes sur le modèle de l'utilisateur ne sont pas satisfaites. On conserve plutôt l'opérateur comme candidat avec une mention indiquant que telle ou telle contrainte n'est pas satisfaite. On verra plus loin que ce type d'information est utile lors du processus de recouvrement servant à produire une nouvelle explication lorsqu'une explication n'est pas comprise.

À ce stade-ci, l'algorithme doit prendre une décision quant à l'opérateur candidat à choisir parmi l'ensemble des candidats disponibles. Ce choix est effectué à

l'aide d'une heuristique de sélection, laquelle peut se baser sur des facteurs tels que: ce que l'utilisateur connaît, ce qui a été dit jusqu'à maintenant, ce que l'utilisateur préfère ou toute autre considération pouvant apporter une aide pour effectuer un choix. Une fois le candidat choisi, ce choix est enregistré, de même que tous les autres candidats n'ayant pas été choisis. Le fait de conserver la liste des candidats une fois la sélection d'un candidat réalisée s'avère nécessaire pour permettre au mécanisme d'explications de réagir si l'utilisateur n'est pas satisfait d'une explication. Ce faisant, il est alors possible d'employer une autre stratégie comme par exemple, donner un exemple si le fait de fournir une définition n'a pas produit l'effet escompté.

L'étape suivante consiste à développer le noyau et le satellite de l'opérateur retenu. Le développement d'un noyau ou d'un satellite s'effectue de la même façon que l'on vient de décrire, c'est-à-dire, en recherchant les opérateurs dont l'effet correspond au but du noyau ou du satellite. Ce but peut être soit un but de communication, soit un but linguistique (but rhétorique ou acte de discours). Le développement d'un satellite n'est pas nécessairement effectué contrairement au développement d'un noyau, la raison étant qu'un satellite peut être optionnel. Selon que le mécanisme d'explications est dans un mode verbeux ou non, cela amène le satellite à être soit considéré, soit ignoré. Cependant, l'algorithme ne développera pas nécessairement un satellite optionnel, même si le mécanisme d'explications est dans un état verbeux. En effet, pour qu'un satellite optionnel soit développé, il ne faut pas qu'il mène à la production d'un segment de texte dont le contenu est déjà connu par l'utilisateur. Par exemple, si l'objectif d'un satellite optionnel est de donner des instructions quant à la façon d'exécuter une action et que le modèle de

l'utilisateur indique que l'utilisateur connaît la façon de faire, alors le satellite ne sera pas développé.

Le plan de texte est complété une fois que chaque sous-but a été affiné en acte de discours. Une fois complété, le plan de texte est acheminé au générateur de texte. Cependant, il serait faux de croire que la génération du plan de texte est un processus d'une seule itération sans aucun retour arrière. En réalité, il est possible durant la génération du plan de texte que l'algorithme ne trouve aucun opérateur pour satisfaire un but. Cela peut se produire en plusieurs circonstances:

- L'effet d'aucun opérateur ne correspond au but à accomplir.
- Les contraintes des opérateurs qui sont candidats potentiels ne sont pas satisfaites.
- Tous les opérateurs candidats ont déjà été utilisés. Cela se produit lorsque le mécanisme d'explications en est à une nième tentative pour accomplir un but de communication.

Lorsque cela se produit, le planificateur de texte doit effectuer un retour arrière d'un niveau dans l'arborescence du plan de texte, puis tenter de réaliser le but de ce noeud-parent en utilisant une autre stratégie, laquelle se trouve parmi les opérateurs candidats non utilisés pour ce noeud-parent. Si aucun de ces candidats ne parvient à réaliser le but (pour cela, il faut aussi que tout l'arbre sous-jacent puisse s'accomplir), alors le planificateur remonte d'un autre niveau. Si le planificateur remonte jusqu'au noeud-racine et que toutes les stratégies de ce noeud ont été testées, l'algorithme ne peut produire un plan de texte.

La complexité de l'algorithme de planification de texte peut être évaluée en terme du nombre d'opérateurs candidats testés lors de l'élaboration d'un plan de texte. Ainsi, soit n, le nombre de noeuds d'un plan de texte, et p, le nombre d'opérateurs candidats par noeud, il existe alors p^n combinaisons possibles d'opérateurs dans un plan de texte. Dans le pire cas, il se peut qu'une seule de ces combinaisons permette de réaliser le plan de texte et qu'elle soit trouvée en dernier. La complexité de l'algorithme de planification de texte est donc $O(p^n)$ dans le pire cas. Cela peut donc résulter en un très grand nombre de tests, mais dans les faits, plusieurs facteurs viennent atténuer ce nombre:

- Les noeuds-feuilles d'un plan de texte qui sont des actes de discours et qui ne possèdent qu'un seul noeud candidat (diminue n).
- Un nombre moyen d'opérateurs candidats égal à 2 ou 3 (p petit).
- Un nombre de contraintes limité pour chaque opérateur qui diminue les chances qu'un opérateur soit rejeté.
- Une base de connaissances explicatives assez riche pour être en mesure de satisfaire les contraintes des opérateurs, et diminuer par le fait même le nombre d'opérateurs rejetés à cause de contraintes non satisfaites.

Le Tableau 6.1 montre les différentes étapes de l'algorithme de planification de texte.

Tableau 6.1: Algorithme de planification de texte

1. Envoyer un but de communication au planificateur de texte.
2. Trouver tous les opérateurs dont l'effet correspond au but à accomplir.
3. Pour chaque opérateur trouvé à l'étape 2, vérifier ses contraintes:
   si toutes ses contraintes sont satisfaites
   alors créer un opérateur candidat
   sinon si les seules contraintes non satisfaites sont des contraintes du modèle de l'utilisateur
   alors créer un opérateur candidat et conserver les hypothèses
4. Sélectionner l'un des opérateurs candidats généré à l'étape 3.
5. *Si* le but du noyau de l'opérateur choisi n'est pas un acte de discours
   alors envoyer ce but puis retourner à l'étape 2.
6. *Si* le satellite de l'opérateur choisi existe
   et que le mécanisme d'explications est en mode verbeux
   et que le but du satellite de l'opérateur choisi n'est pas un acte de discours
   et que le satellite est, soit requis, soit optionnel (mais menant à la production d'un segment de texte ne correspondant pas à un contenu déjà connu par l'utilisateur)
   alors envoyer ce but puis retourner au point 2.
7. Envoyer le plan de texte au générateur de texte.

Les étapes 5 et 6 (Tableau 6.1) font ressortir une caractéristique de l'algorithme de planification de texte, à savoir: il s'agit d'un algorithme récursif du type *profondeur d'abord.* En effet, l'arbre représentant le plan de texte est généré en développant complètement une branche (du noeud-racine au noeud-feuille) avant de développer une autre branche, contrairement à un algorithme du type *largeur d'abord* où chaque noeud de niveau n est développé avant de développer un noeud de niveau $n+1$. Que l'algorithme de planification de texte soit du type *profondeur d'abord* ou qu'il ait été du

type *largeur d'abord* n'a cependant pas de conséquence, car le plan de texte résultant est le même dans les deux cas. On pourrait être porté à croire qu'étant donné les instanciations de variables effectuées dans les contraintes des opérateurs, les valeurs de ces variables instanciées pourraient dépendre de l'ordre selon lequel les opérateurs sont développés, mais il n'en est rien. Sans en faire la preuve, cela réside dans le fait qu'aucune variable n'est présente dans deux noeuds différents sans que l'ancêtre commun à ces deux noeuds n'ait défini la valeur de cette variable.

6.8 Génération du texte explicatif

Le produit de l'algorithme de planification texte est un plan de texte représenté par une structure arborescente dans laquelle chaque noeud-feuille est un acte de discours. Dans cette arborescence, le contenu du texte explicatif y est à la fois présent et structuré. D'une part, le contenu, extrait à partir des contraintes de chaque opérateur du plan de texte, est présent dans les actes de discours et dans certains buts rhétoriques. D'autre part, la structure du contenu y est représentée par la structure même du plan de texte. Bien que la structure du texte explicatif à générer soit en corrélation directe avec la structure du plan de texte, le résultat dépend du type de parcours que l'on effectuera dans cette arborescence au moment de générer l'explication.

En effet, puisque l'organisation du contenu se trouve dans une structure arborescente, on sait qu'il existe plusieurs façons de parcourir un arbre. Il faut donc s'assurer d'utiliser le bon type de parcours au moment de générer l'explication. Un parcours du type *profondeur d'abord* s'avère être le type de parcours à utiliser puisqu'en

utilisant ce type de parcours, le texte à générer correspond à l'ordre des actes de discours rencontrés. Cela est logique puisque chaque segment de texte d'une explication est en fait une branche de l'arbre et que cette explication est une suite de segments de texte liés par des relations rhétoriques. Il faut donc générer l'explication en parcourant l'arbre branche par branche et non pas niveau par niveau. Le parcours de l'arbre d'un plan de texte est donc un processus $O(n)$ où n est le nombre de noeuds à visiter.

6.8.1 Techniques de génération de texte

Il existe principalement deux techniques pour produire le texte de chacun des segments de texte d'une explication. La première technique consiste à fournir le contenu de chaque acte de discours et but rhétorique à un composant appelé *générateur de texte*. Un générateur de texte est composé d'éléments tels une grammaire de la langue, un lexique de base et un lexique du domaine, et est capable de produire des textes cohérents et naturels. La seconde technique consiste à associer le contenu de chaque acte de discours et but rhétorique à des bouts de texte préfabriqués. Cette seconde technique est beaucoup moins complexe à implanter que la première, mais cette simplicité se paye par des explications moins naturelles et dont la fidélité est plus difficile à assurer.

Considérant que les efforts du présent travail de recherche doivent être consacrés à valider l'hypothèse selon laquelle les capacités de réaction de l'algorithme à mémoire d'intentions permettent de satisfaire, en matière d'explication, un utilisateur dont on ne possède pas de modèle complet et précis, on a adopté la seconde technique.

6.9 Réaction au feed-back de l'utilisateur

Les observations qui sont ressorties de l'analyse de dialogues effectuée par Moore et Paris, présentée à la section 4.2, démontrent que le requérant (utilisateur) est rarement totalement satisfait de l'explication fournie par l'expert (système). C'est pourquoi, une fois un texte explicatif présenté, le système se met en attente d'un feed-back de l'utilisateur, ce feed-back pouvant être de trois types:

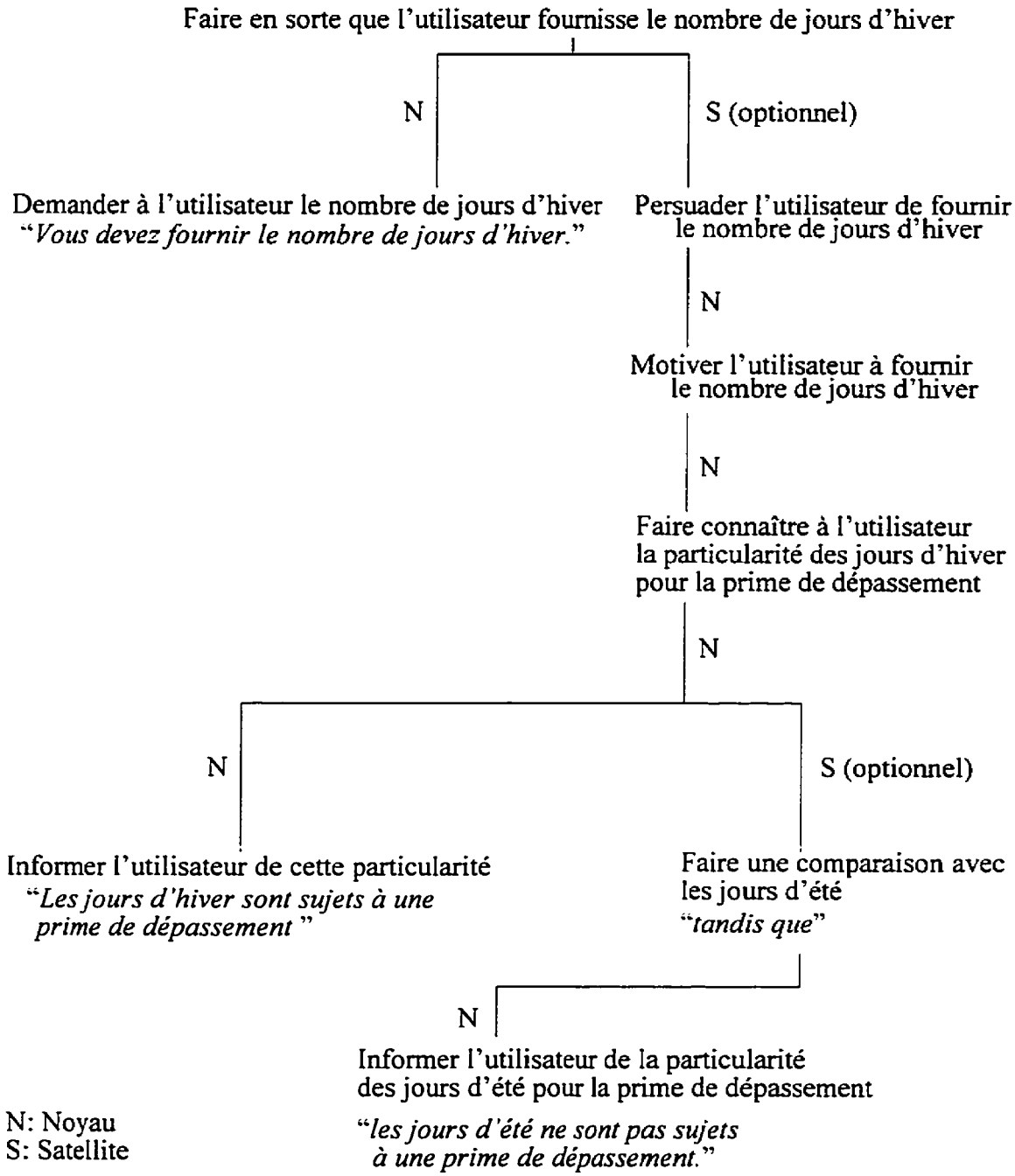
1. l'utilisateur indique qu'il a compris l'explication, auquel cas le système n'a pas d'autre texte explicatif à fournir et peut donc poursuivre à l'étape suivante de sa tâche,
2. une question est demandée en rapport avec le texte explicatif qui vient d'être présenté,
3. l'utilisateur n'a pas bien compris, auquel cas le mécanisme d'explications doit clarifier une partie du texte explicatif ou élaborer une toute autre explication.

Le premier type de feed-back, exprimé par l'utilisateur sous la forme d'un *ok* par exemple, est trivial à traiter puisqu'il n'y a rien à faire. Dans le cas des deux autres types de feed-back, la tâche du système est plus complexe puisqu'il doit interpréter et répondre à l'utilisateur en tenant compte du contexte du dialogue dans lequel s'est produit le feed-back.

6.9.1 Réaction suite à une question

Lorsqu'un texte explicatif est présenté à l'utilisateur, celui-ci peut vouloir, et cela est tout à fait légitime, poser des questions dans le but d'obtenir des éclaircissements ou d'autres détails qui vont lui permettre de mieux saisir le message contenu dans le texte explicatif. Les types de questions que l'on se propose de supporter sont soit des questions à propos des concepts, soit des questions du type *pourquoi* à propos des parties du texte.

De façon à bien illustrer les différents types de questions, voyons d'abord à la page suivante un plan de texte ainsi que le texte explicatif correspondant.




```
Vous devez fournir le nombre de jours d'hiver. Les jours d'hiver
sont sujets à une prime de dépassement tandis que les jours
d'été ne sont pas sujets à une prime de dépassement.
```

Figure 6.3 Plan de texte et l'explication produite

De ce texte explicatif, l'utilisateur peut poser des questions à propos des concepts, par exemple:

- Qu'est-ce que les jours d'hiver?
- Qu'est-ce que les jours d'été?
- Qu'est-ce qu'une prime dépassement?
- Quelles différences y a-t-il entre les jours d'hiver et les jours d'été?
- Quelles similitudes y a-t-il entre les jours d'hiver et les jours d'été?

Quant aux questions du type *pourquoi*, elles ne se rapportent pas au mot ou expression identifiant un concept, car une question du genre *Pourquoi les jours d'hivers?* n'a pas de sens. Elles se rapportent plutôt aux actes de discours contenus dans le texte explicatif. Ainsi, chaque acte de discours présent dans l'explication est sujet à une question du type *pourquoi*. Si l'on examine l'exemple de texte explicatif présenté à la Figure 6.3, et sachant que ce texte est composé de trois actes de discours (un du type *recommander* et deux du type *informer*), on a les questions potentielles suivantes:

- Pourquoi les jours d'hiver sont sujets à une prime de dépassement?
- Pourquoi les jours d'été ne sont pas sujets à une prime de dépassement?
- Pourquoi vous me demandez de fournir le nombre de jours d'hiver?

Lorsque l'utilisateur formule une question du type *Qu'est-ce que tel concept?* à propos d'un concept, le mécanisme d'explications interprète cette requête comme une demande pour faire connaître quelque chose à propos de ce concept. La stratégie consiste

alors à envoyer à l'algorithme de planification de texte, un but de communication dont l'effet à créer sur l'utilisateur est de faire en sorte qu'il connaisse quelque chose à propos du concept auquel la question se rapporte. Or, on sait qu'il peut exister plusieurs relations rhétoriques pour un même effet. Dans ce cas-ci, on peut utiliser par exemple une relation d'élaboration, une relation circonstancielle ou encore une relation comparative avec un autre concept.

Par exemple, avec la question suivante:

```
Qu'est-ce qu'une prime de dépassement?
```

alors le mécanisme d'explications peut répondre par une relation d'élaboration par définition telle que:

```
C'est un prix supplémentaire à payer pour chaque
kilowatt de puissance appelée au-delà des limites
établies selon le tarif général applicable.
```

ou par une relation circonstancielle de temps comme suit:

```
C'est un prix supplémentaire à payer en période
d'hiver seulement.
```

Lorsque plusieurs stratégies existent, comme c'est le cas avec l'exemple précédent, le mécanisme d'explications doit s'assurer d'utiliser une stratégie qui ne répète pas des choses qui ont déjà été dites. Cette vérification s'effectue à l'aide de l'historique du dialogue. Ainsi, toujours selon l'exemple précédent, si l'historique du dialogue indique que l'on a déjà informé l'utilisateur à propos du concept *prime de dépassement* à l'aide d'une relation rhétorique d'élaboration par définition, on devra opter pour une autre

stratégie qui n'a pas été utilisée. Si une telle stratégie n'existe pas, on pourrait signaler à l'utilisateur qu'il n'y a pas de texte explicatif à fournir puisqu'il connaît déjà toutes les réponses. Ou bien, on pourrait utiliser une stratégie par défaut, comme par exemple une relation d'élaboration par définition, laquelle est toujours applicable si l'on exige que chaque concept possède une définition dans la base de connaissances explicatives. Cette seconde approche est celle qui a été retenue. On pense qu'il est préférable de fournir une réponse à l'utilisateur plutôt que de le laisser sans réponse puisque c'est l'utilisateur lui-même qui a explicitement demandé de l'information à propos d'un concept. C'est pour cette même raison qu'on ne valide pas une stratégie en fonction du modèle de l'utilisateur puisque, même si ce modèle indiquait que l'utilisateur connaît le concept *prime de dépassement*, on doit produire un texte explicatif puisque l'utilisateur le demande.

Dans le cas des questions visant à comparer deux concepts (par exemple, *Quelles différences y a-t-il entre les jours d'hiver et les jours d'été?*), le mécanisme d'explications envoie un but de communication dont l'objectif est de faire en sorte que l'utilisateur connaisse les différences entre les deux concepts. Ce type d'intention ne peut être réalisé que par une relation rhétorique comparative.

Pour ce qui est des questions du type *pourquoi*, le mécanisme d'explications doit d'abord interpréter le sens de cette question avant de pouvoir établir sa stratégie pour répondre. Puisque ce type de question est formé à partir des actes de discours, l'interprétation qui en découle dépend du type d'acte de discours en cause. Par exemple, le fait de demander une question du type *pourquoi* en rapport avec un acte de discours du type *informer* doit être interprété comme une demande de persuasion de la véracité de

l'affirmation contenue dans l'acte de discours. Par exemple, soit la question suivante:

> Pourquoi les jours d'été ne sont pas sujets à une prime de dépassement?

Par l'envoi d'un but de communication de persuasion de l'affirmation, le mécanisme d'explications peut répondre par une relation rhétorique d'évidence[1] telle que:

> Parce que durant les mois d'été, la disponibilité plus grande sur le réseau permet de répondre à une demande accrue.

Si par contre l'acte de discours est une recommandation (faire tel choix par exemple) ou une demande (fournir une information par exemple), alors une question du type *pourquoi* en rapport avec ces actes de discours doit être interprétée respectivement comme une demande de persuasion d'effectuer un choix ou une demande de persuasion de faire une action. Par exemple, avec la question suivante:

> Pourquoi vous me demandez de fournir le nombre de jours d'hiver?

le mécanisme d'explications envoie un but de communication au planificateur de texte pour persuader l'utilisateur de faire l'action de fournir le nombre de jours d'hiver. Comme pour tout autre but de communication, il est possible qu'il existe différentes stratégies pour répondre. L'une d'elle consiste à utiliser une relation rhétorique de motivation par particularité du concept, mais cette stratégie a déjà été utilisée dans l'explication initiale (*Les jours d'hiver sont sujets à une prime de dépassement*). Une autre stratégie consiste à

---

1. Une relation rhétorique d'évidence est une relation dont l'intention du satellite est d'augmenter le degré de croyance de l'utilisateur face au contenu présenté par le noyau.

utiliser une relation rhétorique de motivation par le but. Ainsi, si le modèle de l'utilisateur contient une information quant à un but de l'utilisateur dont l'action de fournir le nombre de jours d'hiver est une étape ou une sous-étape, alors le mécanisme d'explications peut produire une réponse du genre:

> `Parce que je dois calculer la prime de dépassement et que le nombre de jours d'hiver est un des paramètres pour calculer la prime de dépassement.`

6.9.2 Réaction suite à une incompréhension

Dans d'autres cas, il arrive que l'utilisateur ne sache pas ce qu'il ne comprend pas et a donc des difficultés à formuler une question claire. Dans ce type de situation, le mécanisme d'explications doit permettre à l'utilisateur d'exprimer son désarroi. Cela peut se faire par exemple en permettant une requête du type *Hein?*. Ainsi, chaque fois que le feed-back de l'utilisateur est *Hein?*, le mécanisme d'explications doit interpréter ce feed-back comme une demande de clarification de l'explication produite.

La première difficulté à laquelle est confronté le mécanisme d'explications est de déterminer quel but de communication dans le texte explicatif n'a pas été atteint, puisqu'il y a bel et bien au moins un tel but qui a failli à la tâche étant donné que l'utilisateur n'a pas bien compris l'explication.

Il existe différentes raisons pour lesquelles un but de communication n'a pas réussi à créer l'effet voulu sur l'utilisateur. L'une d'entre elles est l'utilisation d'une stratégie ayant nécessité de poser une hypothèse sur le modèle de l'utilisateur. En effet, on

se rappellera que durant la planification de texte, le mécanisme d'explications peut supposer que l'utilisateur connaît une certaine chose de façon à pouvoir utiliser une stratégie dont l'une des contraintes est que l'utilisateur connaisse cette chose. Ce faisant, il est alors possible que le mécanisme d'explications ait à clarifier une partie de l'explication en rapport avec le fait que l'utilisateur n'avait pas la connaissance requise pour bien comprendre l'explication initiale. Cela revient à initier un nouveau but de communication par lequel le mécanisme d'explications va s'assurer que l'hypothèse posée soit maintenant vraie. Par exemple, soit le texte explicatif suivant:

> Vous devez fournir le nombre de jours d'hiver. Les jours d'hiver sont sujets à une prime de dépassement tandis que les jours d'été ne sont pas sujets à une prime de dépassement.

En supposant que le modèle de l'utilisateur ne contient aucune mention au fait que l'utilisateur connaît quelque chose à propos du concept *prime de dépassement*, la stratégie employée a donc nécessité de poser une hypothèse, soit celle selon laquelle l'utilisateur sait quelque chose à propos de ce concept. Suite au feed-back de l'utilisateur, un feed-back du type *Hein?*, le mécanisme d'explications lancera un nouveau but de communication pour rendre vraie l'hypothèse, donc un but de communication dont l'objectif est de faire en sorte que l'utilisateur connaisse quelque chose à propos du concept *prime de dépassement*. Le texte explicatif suivant pourrait alors être généré:

> Une prime de dépassement est un prix supplémentaire à payer pour chaque kW de puissance appelée au-delà des limites établies selon le tarif général applicable.

La seconde raison pour laquelle un but de communication n'a pas réussi à créer l'effet voulu sur l'utilisateur est l'absence de support au texte produit par le noyau d'un opérateur (stratégie). Cela peut se produire lorsque l'opérateur de planification de texte utilisé pour le but de communication contient un satellite optionnel qui n'a pas été développé. Par exemple, le plan de texte de la Figure 6.3 montre que le but de communication en tête de l'arbre est réalisé par un opérateur contenant un satellite optionnel. En supposant que le mécanisme d'explications est dans un mode laconique, ce satellite optionnel n'est pas développé et le texte explicatif suivant en résulte:

```
Vous devez fournir le nombre de jours d'hiver.
```

Si le feed-back de l'utilisateur indique qu'il ne comprend pas la demande, le mécanisme d'explications détectera, en examinant le plan de texte contenu dans l'historique du dialogue, qu'un satellite optionnel n'a pas été développé et initiera une nouvelle planification de texte pour développer ce satellite. Dans le cas de la Figure 6.3, il s'agit d'un but de communication de persuasion qui produit le texte suivant:

```
Les jours d'hiver sont sujets à une prime de dépasse-
ment tandis que les jours d'été ne sont pas sujets à
une prime de dépassement.
```

On remarque que le texte explicatif produit est le même que celui qui aurait été produit si le feed-back de l'utilisateur avait été *Pourquoi vous me demandez de fournir le nombre de jours d'hiver?*. Cela est tout à fait cohérent puisqu'en rapport avec un acte de discours de type *demander*, le mécanisme d'explications interprète une question du type *pourquoi* comme une requête de persuasion du bien fondé de cette demande. Et puisque le

satellite optionnel développé a justement pour but de persuader l'utilisateur, cela explique pourquoi on aurait obtenu la même explication.

Le mécanisme d'explications possède donc au moins deux stratégies de recouvrement pour venir en aide à l'utilisateur lorsqu'une explication n'est pas bien comprise: l'envoi d'un but de communication pour rendre vraie une hypothèse et le développement d'un satellite optionnel. Ces stratégies de recouvrement ont essentiellement pour objectif de fournir un complément d'information, par rapport au texte original, pour le noeud ayant failli à la tâche: rendre vraie une hypothèse en élaborant un concept ou justifier une affirmation par le développement de son satellite optionnel.

Plusieurs questions se posent cependant. Premièrement, la recherche du noeud ayant failli à la tâche doit-elle s'effectuer en considérant d'abord le premier segment de texte de l'explication ou le dernier segment? Deuxièmement, quelle stratégie de recouvrement doit-on privilégier en premier: rendre vraie une hypothèse ou développer un satellite optionnel?

Pour répondre à la première question, on introduit ici la notion de *centre d'attention.* Chaque personne qui lit un texte centre nécessairement son attention, consciemment ou inconsciemment, sur des concepts ou des objets de ce texte (Mckeown [16]). On distingue cependant deux types de centre d'attention: immédiat et global. Le centre d'attention immédiat est celui qui fait référence à la façon dont le centre d'attention change ou demeure constant lors du passage d'une phrase à l'autre. Le centre d'attention global décrit pour sa part le centre d'attention sur l'ensemble du texte. Il englobe donc plus

de concepts, plus d'informations que le centre d'attention immédiat. Les deux stratégies de recouvrement que l'on a exposées sont des stratégies qui viennent ajouter un complément d'information à un noeud ou à un segment du plan de texte. Elles ont donc une portée locale et affectent le centre d'attention immédiat. Pour cette raison, on considère qu'il est préférable de parcourir le plan de texte en commençant par le dernier segment de texte pour trouver le but ayant failli à la tâche: on essaye de corriger d'abord le sujet abordé dans le centre d'attention immédiat plutôt qu'un sujet abordé antérieurement.

Quant à savoir laquelle des deux stratégies doit être appliquée en premier, on pose l'hypothèse qu'il est préférable de s'assurer d'abord que l'utilisateur ait les connaissances requises (rendre vraie une hypothèse) avant de lui fournir davantage d'informations (développer un satellite optionnel).

Lorsqu'aucune des deux stratégies de recouvrement ne peut s'appliquer, c'est-à-dire, lorsqu'aucun noeud du plan de texte n'a généré d'hypothèse et ne possède de satellite optionnel non développé, le mécanisme d'explications a recours à une autre stratégie. Cette stratégie consiste à essayer autre chose pour produire le texte explicatif. Cette autre chose, c'est la liste des opérateurs candidats, conservée dans chacun des noeuds du plan de texte, qui va permettre de le générer. Parmi les opérateurs candidats d'un noeud, on retrouve bien sûr tous les opérateurs dont l'effet correspond au but à accomplir, mais également chaque intanciation d'un même opérateur.

Ainsi, reprenons l'exemple de la Figure 6.3 dont le texte explicatif est

reproduit ci-dessous:

> Vous devez fournir le nombre de jours d'hiver. Les jours d'hiver sont sujets à une prime de dépassement tandis que les jours d'été ne sont pas sujets à une prime de dépassement.

En supposant qu'aucune hypothèse n'a été émise et que tous les satellites optionnels ont été développés lors de la planification de texte, le mécanisme d'explications tente alors de trouver un noeud possédant une autre stratégie pour générer un autre texte explicatif. Il se trouve, pour fin de l'exemple, que le noeud dont le but est de motiver l'utilisateur à fournir le nombre de jours d'hiver possède une autre stratégie que celle utilisée (motiver en citant une particularité). Cette autre stratégie consiste à motiver par le but comme suit:

> Je dois calculer la prime de dépassement et le nombre de jours d'hiver est un des paramètres pour calculer la prime de dépassement.

Le recouvrement par l'utilisation d'un autre opérateur candidat est une stratégie qui tend non pas à ajouter un complément à un segment de texte, mais plutôt à vouloir dire les choses autrement. Il s'agit d'une approche plus globale qui, on suppose, affecte davantage le centre d'attention global que le centre d'attention immédiat. Pour cette raison, la recherche d'un noeud possédant des opérateurs candidats doit s'effectuer en commençant par le premier segment de texte (noeud-racine du plan de texte) plutôt que par le dernier segment de texte comme dans le cas des autres stratégies de recouvrement.

Le mécanisme de planification de texte qui vient d'être présenté a été implanté puis testé à l'aide d'un prototype. Ce prototype fait l'objet du prochain chapitre.

CHAPITRE 7
PROTOTYPE

Un prototype du mécanisme d'explications présenté au chapitre précédent a été implanté de façon à vérifier sa faisabilité, mais surtout, de façon à valider ses capacités de réaction. Le présent chapitre expose la méthodologie employée pour la réalisation de ce prototype.

7.1 Domaine d'application

Pour être en mesure de tester le mécanisme d'explications et son algorithme de planification de texte, on doit dans un premier temps choisir un domaine de connaissances avec lequel l'utilisateur et le système vont pouvoir participer à un dialogue explicatif. Le choix de ce domaine de connaissances ne s'est pas fait au hasard. Considérant que ce mécanisme d'explications est l'une des composantes d'un système d'aide au transfert de connaissances pour les représentants commerciaux d'Hydro-Québec, il convient, ne serait-ce que pour susciter encore plus l'intérêt, de choisir un domaine de connaissances avec lequel les représentants commerciaux doivent composer. L'un de ces domaines est la tarification de l'électricité à Hydro-Québec. En plus de constituer un domaine de connaissances d'intérêt pour les représentants commerciaux, ce domaine a l'avantage de ne pas être trop complexe, d'être bien circonscrit et d'avoir des connaissances facilement accessibles [9,10,11].

7.2 Exemples de dialogues

L'une des façons de faire pour valider les capacités de réaction du mécanisme d'explications consiste à comparer des dialogues explicatifs obtenus entre un utilisateur et le système à ceux obtenus entre un utilisateur et un expert humain.

Dans cette optique, on a soumis quelques participants (de niveaux débutant et intermédiaire dans le domaine de la tarification) à de courtes séances de consultation avec un expert humain. Chacun des dialogues s'amorçait par une réponse de l'expert, suite à quoi le participant devait fournir un feed-back en se restreignant toutefois aux types de feed-back que l'on a présentés précédemment, soit:

- une question du type *Qu'est-ce que tel concept?*,
- une question du type *Quelle est la différence entre telle chose et telle chose?*,
- une question du type *Pourquoi telle chose?*,
- une question du type *Hein?* pour demander une clarification si l'explication est mal comprise,
- un *OK* pour mettre fin au dialogue.

Les Tableaux 7.1 à Tableau 7.4 illustrent quatre exemples de dialogue obtenus entre un participant et l'expert humain.

Tableau 7.1: Dialogue humain-humain #1

Expert: *Il n'y qu'un seul tarif applicable, c'est le tarif M.*

Requérant: *Pourquoi?*

Expert: *Étant donné que vous avez indiqué une puissance souscrite de 600 kW, c'est le tarif M qui s'applique.*

Requérant: *Hein?*

Expert: *Bien oui, le tarif M s'applique à tout abonnement qui a une puissance souscrite d'au moins 100 kW, mais inférieure à 5000 kW.*

Requérant: *C'est quoi au juste une puissance souscrite?*

Expert: *C'est la puissance minimale à payer pour un abonnement. Elle peut être supérieure à la puissance maximale appelée.*

Requérant: *Hein?*

Expert: *Par exemple, si vous prenez une puissance souscrite de 600 kW, vous devrez payer une prime de puissance minimale pour 600 kW même si votre appel maximal de puissance n'a été que de 300 kW!*

Requérant: *OK!*

Tableau 7.2: Dialogue humain-humain #2

Expert: *Les tarifs applicables sont G et G-9.*

Requérant: *Quelles différences qu'il y a entre les tarifs G et G-9?*

Expert : *Pour le tarif G vous devez payer une redevance d'abonnement, mais pas pour le tarif G-9. Aussi, le tarif G comporte deux tranches de prix régressifs pour l'énergie. Le tarif G-9 lui n'a qu'une seule tranche de prix pour l'énergie. En plus, la prime de puissance n'est pas la même.*

Requérant: *Quelles différences entre la prime de puissance du tarif G et celle du tarif G-9?*

Expert: *Pour le tarif G la prime de puissance est de 13,17$ pour chaque kW de puissance à facturer qui excède 40 kW. Au tarif G-9 tous les kW de puissance à facturer sont facturés à 3,45$ chacun.*

Requérant: *Quelles différences entre le prix de l'énergie du tarif G et celui du tarif G-9?*

Expert: *Au tarif G les 11700 premiers kWh sont facturés à 7,18 cents, l'excédant lui à 3,62 cents. Pour le tarif G-9 tous les kWh sont facturés à 7,42 cents.*

Requérant: *OK!*

Tableau 7.3: Dialogue humain-humain #3

Expert: *Vous devriez opter pour le tarif G-9 au lieu du tarif G.*

Requérant: *Pourquoi?*

Expert: *Parce que ça va vous coûter moins cher.*

Requérant: *Pourquoi?*

Expert: *Parce que votre facteur d'utilisation est faible.*

Requérant: *Hein?*

Expert: *D'abord, au tarif G, la prime de puissance est plus élevée qu'au tarif G-9. Et puisque votre facteur d'utilisation est faible vous ne pourrez pas bénificier d'économies substancielles sur la deuxième tranche de prix de l'énergie du tarif G. Donc le tarif G-9 est plus avantageux étant donné que la facture sera moins élevée.*

Requérant: *OK!*

Tableau 7.4: Dialogue humain-humain #4

Expert: *Vous devez indiquer le nombre de jours d'hiver de la période.*

Requérant: *Pourquoi?*

Expert: *J'ai besoin de calculer la prime de dépassement pour calculer le montant de la facture, et le nombre de jours d'hiver est un des paramètres dont j'ai besoin.*

Requérant: *Hein?*

Expert: *Bien oui! Les jours d'hivers sont sujets à une prime de dépassement mais pas les jours d'été.*

Requérant: *C'est quoi une prime de dépassement?*

Expert: *C'est un prix supplémentaire à payer pour chaque kW de puissance appelée au-delà d'une certaine limite, et ça, en période d'hiver seulement.*

Requérant: *Hein?*

Expert: *Par exemple, la période du 25 mars au 25 avril comporte 7 jours d'hiver sujets à une prime de dépassement puisque la période d'hiver se termine le 31 mars.*

Requérant: *Pourquoi les jours d'hiver sont sujets à une prime de dépassement?*

Expert: *Parce que durant les mois d'hiver, le réseau est très sollicité et l'achat d'électricité peut s'avérer nécessaire.*

Requérant: *OK!*

7.3 Analyse des dialogues

L'analyse des dialogues précédents doit permettre de faire ressortir les éléments essentiels au mécanisme d'explications pour qu'il puisse participer à de tels types de dialogues explicatifs. Ces éléments essentiels sont les intentions, les relations rhétoriques et les connaissances explicatives. Il est important de signaler ici que les types d'intentions et de relations rhétoriques répertoriés auront pour but de représenter les composantes de la structure des dialogues, et que de telles composantes ne sont pas exclusives aux dialogues étudiés. C'est donc dire que des dialogues touchant d'autres parties du domaine de connaissances de la tarification de l'électricité ou touchant tout autre domaine de connaissances dont les types de dialogues peuvent s'apparenter aux dialogues analysés sont aussi concernés. Il suffit pour cela d'incorporer les connaissances explicatives nécessaires.

7.3.1 Détermination des intentions et des relations rhétoriques

Les capacités de réaction du mécanisme d'explications reposent sur le fait que son algorithme de planification de texte gère à la fois la structure des intentions et la structure des relations rhétoriques d'un texte explicatif. Comme on le sait maintenant, une même intention peut être réalisée parmi un choix de relations rhétoriques, et une même relation rhétorique peut servir à diverses intentions.

L'examen des dialogues explicatifs des Tableau 7.1 à Tableau 7.4 fait ressortir les intentions suivantes:

- Persuader de quelque chose (par exemple, *Parce que votre facteur d'utilisation est faible* (dialogue #3)).
- Persuader de faire quelque chose (par exemple, *J'ai besoin de calculer la prime de dépassement pour calculer le montant de la facture, et le nombre de jours d'hiver est un des paramètres dont j'ai besoin.* (dialogue #4)).
- Faire connaître un concept (par exemple, *C'est la puissance minimale à payer pour un abonnement.* (dialogue #2)).
- Faire connaître la différence entre deux concepts (par exemple, *Au tarif G les 11700 premiers kWh sont facturés à 7,18 cents, l'excédant lui à 3,62 cents. Pour le tarif G-9 tous les kWh sont facturés à 7,42 cents.* (dialogue #2)).
- Amener à croire que c'est le meilleur choix (par exemple, *Parce que ça va vous coûter moins cher.* (dialogue #3)).
- Amener à croire à une proposition (par exemple, *Il n'y a qu'un seul tarif applicable et c'est le tarif M.* (dialogue #1)).
- Amener à croire qu'une action est une des étapes pour atteindre un but (par exemple, *...pour calculer le montant de la facture, et le nombre de jours d'hiver est un des paramètres dont j'ai besoin.* (dialogue #4)).

Chacune de ces intentions est réalisée par une relation rhétorique, cette relation rhétorique étant parfois choisie parmi un ensemble de relations rhétoriques. Ainsi, par exemple, l'intention de persuasion de faire quelque chose (fournir un paramètre, opter pour un choix recommandé) a été réalisée par une relation rhétorique de motivation. Il peut

s'agir d'une relation rhétorique de motivation par le but (*J'ai besoin de calculer la prime de dépassement pour calculer le montant de la facture, et le nombre de jours d'hiver est un des paramètres dont j'ai besoin.)*, de motivation par particularité (*Bien oui! Les jours d'hivers sont sujets à une prime de dépassement...*) ou de motivation par avantage (*Parce que ça va vous coûter moins cher.*). Pour ce qui est de l'intention de faire connaître un concept, cela a été réalisé soit par une relation rhétorique d'élaboration par définition (*C'est le rapport de la consommation moyenne sur la consommation maximale pour une période de consommation.*), soit par une relation rhétorique circonstancielle (*...en période d'hiver seulement.*), soit par une relation d'élaboration par un exemple (*Par exemple, la période du 25 mars au 25 avril comporte 7 jours d'hiver sujets à une prime de dépassement puisque la période d'hiver se termine le 31 mars.*) ou encore par une relation comparative (*La puissance souscrite peut être supérieure à la puissance maximale appelée*). D'un autre côté, l'intention d'amener à croire une proposition a pu être réalisée soit par une relation rhétorique d'élaboration par énumération des membres d'un ensemble (*Les tarifs applicables sont G et G-9.*), soit par une relation rhétorique d'élaboration par attribut d'un concept (*Les jours d'hivers sont sujets à une prime de dépassement.*), soit par une relation rhétorique de cause à effet (*Étant donné que vous avez indiqué une puissance souscrite de 600 kW, c'est le tarif M qui s'applique.*) ou encore par une relation rhétorique comparative (*Les jours d'hivers sont sujets à une prime de dépassement, mais pas les jours d'été.*).

Le Tableau 7.5 résume les liaisons intentions/relations rhétoriques observées dans les quatre dialogues.

Tableau 7.5: Liaisons intentions/relations rhétoriques

Intentions	Relations Rhétoriques
Persuader de quelque chose	Évidence
Persuader de faire quelque chose	Motivation par le but Motivation par particularité Motivation par avantage
Faire connaître un concept	Élaboration par définition Élaboration par exemple Circonstancielle (temps)
Faire connaître la différence entre deux concepts	Comparative
Amener à croire que c'est le meilleur choix	Comparative
Amener à croire à une proposition	Élaboration Ensemble-Membre Élaboration Objet-Attribut Cause/Effet Comparative
Amener à croire qu'une action est une des étapes pour atteindre un but	Élaboration Processus-Étape Circonstancielle (manière)

Bien sûr, ce tableau ne dresse pas la liste exhaustive de toutes les liaisons possibles entre les intentions et les relations rhétoriques, mais bien celles observées dans les quatre dialogues. Cette remarque est importante puisque sinon, on pourrait être porté à croire que certaines relations rhétoriques ne sont applicables que pour une seule intention au niveau des idées, et qu'à la limite, la structure des intentions pourrait être déduite en grande partie par les relations rhétoriques employées. Il faut conserver en tête que d'autres liaisons intentions/relations rhétoriques peuvent venir s'ajouter pour enrichir l'ensemble des stratégies d'explication, et qu'il est donc nécessaire de dissocier la structure des

intentions de celle des relations rhétoriques (il suffit de se rappeler l'exemple présenté à la section 5.2, *"Problème de représentation des intentions"*, pour s'en convaincre).

7.4 Implantation

Comme premier prototype, on s'est limité à l'implantation des connaissances explicatives et des opérateurs de planification de texte requis pour réaliser les dialogues explicatifs #1 et #4 (Tableau 7.1 et Tableau 7.4).

7.4.1 Outils, langages et environnement de développement

L'une des premières décisions qui doit être prise au moment de réaliser un prototype concerne le choix des outils informatiques, des langages de programmation et de l'environnement de travail à utiliser. Ce choix est bien sûr effectué en fonction des besoins et exigences du client, en l'occurrence les représentants commerciaux.

Certains de ces besoins et exigences sont régis par des règles et pratiques établies. C'est le cas de l'environnement de travail. Chaque représentant commercial ayant à sa disposition un ordinateur portatif de type PC dont les applications s'exécutent sous le système d'exploitation Windows, le choix de la plate-forme et du système d'exploitation dans ce cas-ci est quasi immuable.

D'autres exigences sont dictées par la nature même de la tâche à informatiser. Dans le cas de l'implantation de l'algorithme de planification de texte à mémoire d'intentions, on distingue deux caractéristiques importantes. D'abord, la production des plans de texte nécessite une construction itérative d'arborescences, c'est-à-dire, une

construction pouvant nécessiter des retours-arrières si l'un des chemins en exploration mène à un cul-de-sac. Ensuite, la vérification des contraintes de même que l'identification des opérateurs potentiels nécessitent plusieurs instanciations de variables. Ces deux caractéristiques, soit la gestion des retours-arrières et l'instantiation de variables, sont les points forts d'un langage de 5^{e} génération tel Prolog [1]. En effet, Prolog, de par son moteur d'inférence intégré, offre une gestion automatique des retours-arrières. De plus, Prolog permet de profiter de capacités de traitement symbolique hautement efficace sans aucun effort, contrairement aux langages procéduraux traditionnels comme C, FORTRAN, Pascal, etc. Le choix de Prolog pour l'implantation de l'algorithme de planification de texte est donc justifié de cette façon et l'environnement de programmation Prolog choisi est LPA-PROLOG de la compagnie Logic Programming Associates Ltd [13].

Finalement, la satisfaction de certaines exigences s'effectue de façon plus arbitraire. C'est le cas de l'interface personne-ordinateur. Bien que LPA-PROLOG offre les prédicats nécessaires pour construire une interface personne-ordinateur, il n'est pas aussi souple que d'autres environnements de développement. On préfère plutôt utiliser ses capacités de s'interfacer avec d'autres applications via le mécanisme *DDE* (*Dynamic Data Exchange*) sous Windows, et profiter d'un environnement de développement plus adéquat pour la construction de l'interface personne-ordinateur. Cet environnement, c'est Delphi 2 de la compagnie Borland [29]. Delphi 2 est un puissant environnement de développement en langage Pascal. Il combine à la fois les caractéristiques de la programmation d'interface personne-ordinateur ainsi que la programmation orientée

objet. Son environnement de développement est très convivial et il fournit une architecture basée sur la réutilisation de composants. Cela permet donc un développement rapide d'applications, caractéristique d'un développement par prototypage.

Bref, le mécanisme d'explications verra l'algorithme de planification de texte développée avec LPA-PROLOG, son interface personne-ordinateur avec Delphi 2, le tout fonctionnant sur un ordinateur de type PC dans l'environnement Windows.

7.4.2 Représentation des connaissances explicatives

Les connaissances explicatives nécessaires à la production des explications sont représentées par une base de faits Prolog.

D'abord, chaque concept possède un fait d'identification:

***concept*(identificateur, '*texte*').**

par exemple,

prime_de_depassement(identificateur, 'prime de depassement').

Le texte de l'identificateur est celui qui apparaît dans une explication chaque fois que l'on doit mentionner le nom du concept.

Chaque concept contient également deux faits indiquant son genre (masculin ou féminin) et son nombre (singulier ou pluriel). Par exemple, pour le concept *prime de dépassement*:

prime_de_depassement(genre, feminin).
prime_de_depassement(nombre, singulier).

Ces informations sont prises en compte au moment d'utiliser l'identificateur d'un concept lorsqu'un texte explicatif est généré.

Finalement, chaque concept contient un ensemble de prédicats reliant une relation rhétorique à un bout de texte explicatif. Par exemple:

prime_de_depassement(elaboration_par_definition, 'un prix supplementaire a payer pour chaque kilowatt de puissance appelee au-dela des limites etablies selon le &(tarif general) applicable').

prime_de_depassement(circonstancielle_temps, 's'applique en &(periode d'hiver) seulement').

prime_de_depassement(elaboration_par_exemple, 'la periode du 25 mars au 25 avril comporte 7 jours d'hiver sujet a une prime de depassement').

prime_de_depassement(evidence, elaboration_par_exemple, 'la &(periode d'hiver) debute le 1er decembre et se termine le 31 mars de l'annee suivante').

Ce sont ces faits de la base de connaissances explicatives qui permettent au planificateur et au générateur de l'explication d'extraire les informations nécessaires à la réalisation des intentions par des relations rhétoriques.

On remarquera que certains bouts de texte sont bornés par des marqueurs spéciaux, *&( )*. Comme on le verra un peu plus loin, cette syntaxe sert à identifier les concepts pouvant faire l'objet d'une question du type *Qu'est-ce que...?*.

7.4.3 Représentation des buts de communication et des buts linguistiques

Chaque planification de texte s'amorce par l'envoi, au planificateur de texte, d'un but de communication. On sait qu'un but de communication est représenté en terme d'effet que le système entend créer sur l'état mental de l'utilisateur. Par exemple, on

représentera le fait que l'on veuille que l'utilisateur ait pour but de fournir le nombre de jours d'hivers de la période par le prédicat suivant:

a_pour_but(faire(nbjh_periode)).

De même, le fait que l'on veuille faire connaître à l'utilisateur quelque chose à propos du concept *prime de dépassement* est réalisé par un but de communication représenté comme ceci:

connait_qqc(concept(prime_de_depassement)).

Ensuite, il y a les buts linguistiques, dont la nature est soit un but rhétorique, soit un acte de discours. Chaque but rhétorique est représenté par un prédicat de même nom que la relation qu'il représente et ayant pour argument l'information nécessaire à sa réalisation. Par exemple, un but rhétorique d'élaboration d'un concept par définition est représenté par le prédicat suivant:

elaboration_par_definition(concept(Concept)).

où *Concept* est une variable à être instanciée.

Quant aux actes de discours, qui sont au nombre de trois (*informer*, *demander* et *recommander*), ils sont également représentés par un prédicat portant leur nom et dont l'argument contient l'information nécessaire à la production d'un segment de texte (les actes de discours étant les noeuds-feuilles d'un plan de texte à partir desquels chaque segment de texte d'une explication est généré). L'information faisant partie d'un segment de texte est prise à même la base de connaissances explicatives. Par exemple,

informer(elaboration_par_definition(Concept, Definition)).

permet de générer le segment de texte contenu dans la variable *Definition* découlant de la relation rhétorique d'élaboration par définition du concept contenu dans la variable *Concept*. Ainsi, si la variable *Concept* est instanciée à *prime_de_depassement*, il est alors possible d'instancier la variable *Definition*, sachant que le fait suivant existe dans la base de connaissances explicatives:

prime_de_depassement(elaboration_par_definition, 'un prix supplementaire a payer pour chaque kilowatt de puissance appelee au-dela des limites etablies selon le &(tarif general) applicable').

7.4.4 Représentation du modèle de l'utilisateur

Pour représenter le modèle de l'utilisateur, il est facile, en langage Prolog, de créer un ensemble de faits représentant les connaissances de l'utilisateur, d'en ajouter ou d'en enlever dynamiquement au fur et à mesure du dialogue, et de les utiliser pour vérifier les contraintes d'un opérateur.

Au niveau du prototype, le modèle de l'utilisateur ne comprend initialement que deux faits, soit les buts de l'utilisateur, à savoir, déterminer les tarifs applicables et calculer le montant de la facture:

but(determiner_tarif).
but(calculer_facture).

Il ne contient aucune information quant aux connaissances de l'utilisateur. Elles seront plutôt inférées durant le dialogue. Par exemple, lorsque le but de communication **connait_qqc(concept(prime_de_depassement))** est réalisé par une

relation rhétorique d'élaboration par définition, le fait suivant est ajouté au modèle de l'utilisateur:

connait_qqc(prime_de_depassement, definition).

L'ajout dynamique de connaissances est réalisé grâce au prédicat **assert**. Le retrait d'information du modèle de l'utilisateur s'effectue quant à lui par le prédicat **retract**. Cela peut être nécessaire lorsque les buts de l'utilisateur changent, par exemple.

Parmi les informations contenues et gérées dans le modèle de l'utilisateur du prototype, seuls les buts de l'utilisateur et ses connaissances à propos des concepts y sont représentés.

7.4.5 Représentation de l'historique du dialogue

Tout comme le modèle de l'utilisateur, l'historique du dialogue est représenté par un ensemble de faits. Ces faits relatent ce qui a été dit et comment cela a été dit. Ainsi, lors du parcours d'un plan de texte, au moment de générer une explication, chacun des buts de communication ou des buts linguistiques instanciés des opérateurs du plan de texte permet de créer, à l'aide du prédicat **assert**, un fait pour l'historique du dialogue. Par exemple:

assert(elaboration_par_definition(concept(prime_de_depassement))).
assert(persuade(faire(nbjh_periode))).

Ainsi, lorsque plusieurs stratégies existent, le mécanisme d'explications peut, par exemple, s'assurer d'utiliser une stratégie qui ne répète pas des choses qui ont déjà été dites.

7.4.6 Représentation des opérateurs de planification de texte

Chaque noeud d'un plan de texte est représenté par un opérateur de planification de texte. Il y a donc un opérateur pour chaque but de communication (intention), chaque but rhétorique (relation rhétorique) et chaque acte de discours (*informer*, *demander*, *recommander*). Chaque opérateur comporte les informations suivantes: un nom, un effet, une liste de contraintes, un noyau, un satellite et le statut du satellite. On représente chaque opérateur par un prédicat *op_pt* d'arité six de la forme suivante:

op_pt(Nom, Effet, [Liste_de_Contraintes], Noyau, Satellite, Statut_Satellite).

Les attributs *Effet*, *Noyau* et *Satellite* (lorsqu'il y en a un) correspondent à des buts de communication ou à des buts linguistiques dont la forme a été présentée à la section 7.4.3, *"Représentation des buts de communication et des buts linguistiques"*. Lorsqu'un opérateur n'a pas de satellite. l'attribut *Satellite* prend la valeur *nil*. L'attribut *Statut_Satellite* prend la valeur *nil* (s'il n'y a pas de satellite), *obligatoire* ou *optionnel*. Quant à la liste de contraintes, il s'agit d'une liste de faits à instancier avec des faits de la base de connaissances explicatives ou des faits du modèle de l'utilisateur.

À titre d'exemple, prenons le cas d'une intention de faire connaître quelque chose à propos d'un concept. On sait que cette intention peut être réalisée par une relation rhétorique d'élaboration par définition, qui elle-même est réalisée par un acte de discours de type *informer*. Cette décomposition comporte trois noeuds représentés par les trois opérateurs suivants:

```
op_pt(faire_connaitre_par_definition,
        connait_qqc(concept(Concept)),
        [Concept(identificateur, _)],
        elaboration_par_definition(concept(Concept)),
        nil, nil).

op_pt(elaboration_par_definition,
        elaboration_par_definition(concept(Concept)),
        [Concept(identificateur, _), Concept(definition, Definition)],
        informer(definition(Concept, Definition)),
        nil, nil).

op_pt(informer,
        informer(Proposition),
        [ ],
        afficher(Proposition),
        nil, nil).
```

On remarque que ces opérateurs sont hiérarchiquement rattachables dû au fait que le noyau ou le satellite de l'un se marie à l'effet d'un autre. On dénote aussi que les contraintes de l'opérateur *elaboration_par_definition* s'assurent que le concept faisant l'objet de l'élaboration existe et possède une définition. Finalement, le noyau *afficher(Proposition)* de l'opérateur *informer* correspond en fait à un prédicat comportant une clause pour chaque type de *Proposition* possible. C'est ce même prédicat qui est responsable de la génération de chaque segment de texte d'une explication. On verra un peu plus loin une description plus détaillée de ce prédicat.

7.4.7 Génération d'un plan de texte

La génération d'un plan de texte s'amorce par la réception d'un but de communication par le planificateur de texte. Au niveau du prototype, le planificateur de texte est représenté par un prédicat récursif dont le but est de construire l'arborescence du

plan de texte:

planification(Effet, Arbre) :-
... traitement ...
planification(Noyau, Sous_Arbre_Noyau),
planification(Satellite, Sous_Arbre_Satellite).

Cette structure algorithmique est à l'image d'une construction arborescente par *profondeur d'abord*. La partie *traitement* correspond aux étapes 2, 3 et 4 de l'algorithme de planification de texte du Figure 6.1: trouver les opérateurs potentiels (ceux qui se marient à l'attribut *Effet*), vérifier les contraintes et sélectionner un opérateur candidat.

La sélection d'un opérateur candidat s'effectue présentement de façon un peu arbitraire, c'est-à-dire, sans aucune heuristique bien établie. Quelques opérateurs sont classés selon un ordre de préséance, mais sans plus. Par exemple, on suppose que comme première information d'un concept, il est préférable de donner sa définition plutôt que de présenter un exemple. Si le fait de donner une définition ne suffit pas, alors là on peut donner un exemple. C'est pourquoi, lorsqu'un opérateur d'élaboration par définition et un opérateur d'élaboration par un exemple sont tous deux candidats, le premier a préséance sur le second.

Chaque noeud de l'arbre du plan de texte est identifié par un numéro unique comme indiqué à la Figure 7.1.

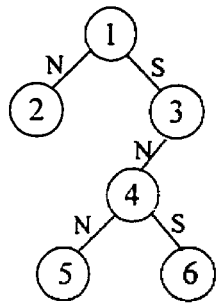


Figure 7.1 Numérotation des noeuds d'un plan de texte

Chaque arbre est représenté par un terme contenant sa structure. La structure de l'arbre à la Figure 7.1 est représentée comme ceci:

a(1,a(2,nil,nil),a(3,a(4,a(5,nil,nil),a(6,nil,nil)),nil)).

Cette forme de représentation est très efficace en Prolog, tant du point de vue de la construction de la structure que de son parcours.

Aussi, pour chaque noeud, la seule information contenue dans l'arbre est son numéro d'identification. Cela n'est cependant pas suffisant puisque que pour chacun des noeuds, on doit également conserver la liste des noeuds candidats, les hypothèses émises, de même que tout satellite optionnel non développé. À défaut de les conserver dans la structure même de l'arbre, non pas que cela soit plus efficace, mais plutôt pour simplifier la structure de l'arbre, ces informations sont conservées dans les structures suivantes:

noeud_candidats(Id, [Liste_de_Candidats]).
noeud_hypothese(Id, Hypothèse).
noeud_satellite_non_developpe(Id, Satellite).

7.4.8 Génération d'un texte explicatif

La génération d'un texte explicatif s'effectue en parcourant l'arbre d'un plan de texte et en générant un segment de texte pour chaque noeud-feuille de l'arbre. Au niveau du prototype, le générateur de texte est représenté par un prédicat récursif qui permet de générer du texte chaque fois qu'un noeud-feuille (acte de discours) est rencontré:

```
parcours_arbre(Arbre) :-
    Arbre = a(Noeud, Sous_Arbre_Noyau, Sous_Arbre_Satellite)
    ... vérifier si Noeud est un acte de discours puis générer le texte ...
    parcours_arbre(Sous_Arbre_Noyau),
    parcours_arbre(Sous_Arbre_Satellite).
```

Cette structure algorithmique permet un parcours de l'arbre par *profondeur d'abord*. Il s'agit du type de parcours nécessaire pour générer un texte explicatif dans le bon ordre (section 6.8, *"Génération du texte explicatif"*).

Lorsque l'algorithme rencontre un noeud-feuille, celui-ci correspond à l'un des trois actes de discours supportés: *informer*, *demander* et *recommander*. À chacun de ces actes de discours est associé un prédicat qui permet, dans un premier temps, de créer un préfixe au segment de texte à générer: *Vous devez* (pour l'acte de discours *demander*), *Vous devriez* (pour l'acte de discours *recommander*) et rien pour l'acte de discours *informer*.

S'en suit, dans un deuxième temps, la génération du segment de texte. Cela s'effectue par le prédicat *afficher(Proposition)*. Ce prédicat contient une clause pour chaque type de *Proposition* possible. Par exemple la proposition suivante:

elaboration_par_definition(Concept, Definition)

permet de générer le segment de texte constitué de la définition d'un concept. L'information nécessaire est contenue dans la base de connaissances explicatives. Par exemple, si la variable *Concept* a pour contenu *prime_de_depassement*, alors la variable *Definition* sera instanciée selon la valeur trouvée dans la base de connaissances explicatives:

prime_de_depassement(elaboration_par_definition, 'un prix supplementaire a payer pour chaque kilowatt de puissance appelee au-dela des limites etablies selon le &(tarif general) applicable').

En utilisant les faits suivants:

prime_de_depassement(identificateur, 'prime de depassement').
prime_de_depassement(genre, feminin).
prime_de_depassement(nombre, singulier).

il est alors possible de générer le segment de texte suivant:

'La prime de dépassement est un prix supplementaire a payer pour chaque kilowatt de puissance appelee au-dela des limites etablies selon le tarif general applicable.'

7.4.9 Communication interprocessus

Puisque l'on a opté pour l'utilisation de l'outil de développement d'interface personne-ordinateur Delphi 2 plutôt que celui fourni par LPA-PROLOG, il est nécessaire de transmettre à l'interface personne-ordinateur chaque segment de texte généré en Prolog. Également, puisque l'on a décidé d'implanter l'algorithme de planification de texte et l'interface personne-ordinateur dans deux processus ou tâches séparés, on utilise

le mécanisme d'échange de données de Windows connu sous le nom de *DDE* (*Dynamic Data Exchange*). Ainsi, chaque requête (but de communication) est transmise de l'interface Delphi vers le processus Prolog via le mécanisme *DDE*, et inversement, dans le cas de la réponse à la requête (texte explicatif).

7.4.10 Création des questions relatives au texte explicatif

De façon à permettre à l'utilisateur d'exprimer son feed-back, des questions relatives aux différents types de feed-back supportés sont générées.

D'abord, une question du type *Qu'est-ce que...?* est générée pour chaque concept rencontré et délimité par les caractères *&( )* (par exemple *&(tarif general)*). Cette tâche est réalisée par l'interface Delphi lorsqu'elle reçoit un texte explicatif.

Ensuite, des questions du type *Pourquoi?* sont générées en rapport avec les actes de discours contenus dans l'explication. Un fait est conservé pour chaque acte de discours générant une question du type *Pourquoi?*:

pourquoi(Id, Acte_de_discours).

La liste des questions du type *Pourquoi?* accompagnées de leur numéro d'identification (*Id*) est transmise, en même temps que le texte explicatif, à l'interface Delphi.

Finalement, l'interface Delphi complète la liste de questions par une question du type *Hein?*. La Figure 7.2 montre un exemple de texte explicatif accompagné de la liste des questions potentielles en rapport avec ce texte.

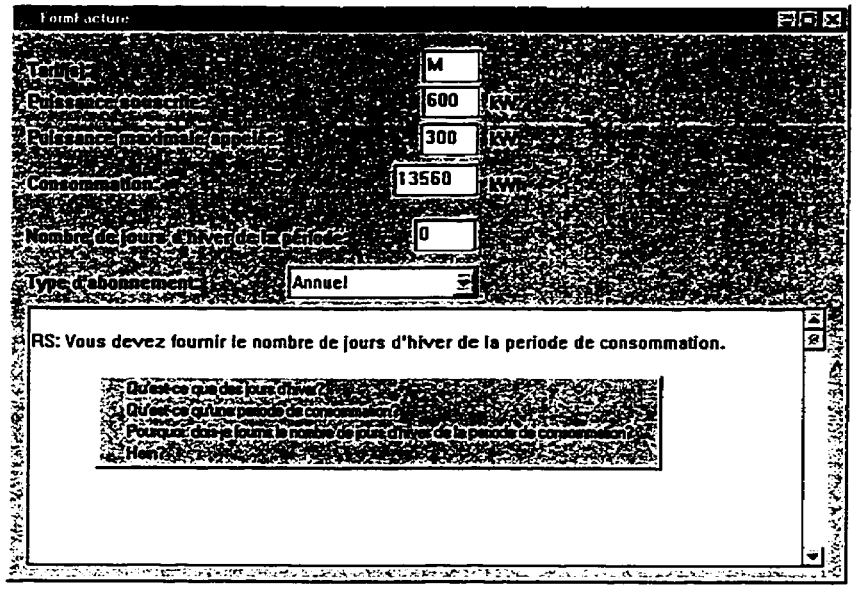


Figure 7.2 Texte explicatif et liste de questions

7.4.11 Traitement du feed-back de l'utilisateur

L'utilisateur exprime son feed-back par un choix au menu de questions (Figure 7.2). Le traitement effectué alors est fonction de la nature de ce feed-back. Si le feed-back est une question du type *Qu'est-ce que...?*, alors l'interface Delphi envoie au processus Prolog un but de communication du type:

connait_qqc(Concept).

où *Concept* est le concept exprimé dans la question. Dès lors, une nouvelle planification de texte s'amorce et un nouveau texte explicatif est généré.

Si le feed-back est une question du type *Pourquoi?*, alors l'interface Delphi envoie au processus Prolog un appel au prédicat suivant:

persuader(Id).

où *Id* est le numéro d'identification de la question du type *Pourquoi?* fourni par le processus Prolog lors de la génération des questions. Ce prédicat permet au processus Prolog de récupérer l'acte de discours correspondant à la question et d'envoyer à son planificateur de texte le but de communication suivant:

persuade(Acte_de_discours).

Finalement, si le feed-back est une question du type *Hein?*, alors l'interface Delphi envoie au processus Prolog un appel au prédicat *hein*. Ce prédicat est responsable du processus de recouvrement lorsqu'un texte explicatif n'est pas bien saisi par l'utilisateur (section 6.9.2, *"Réaction suite à une incompréhension"*). Il y a trois stratégies de recouvrement implantées:

1. rendre vraie une hypothèse,
2. développer un satellite optionnel non développé,
3. essayer un autre opérateur candidat.

Ces stratégies sont répertoriées en deux catégories:

1. celles en rapport avec le centre d'attention immédiat (stratégies 1 et 2),
2. celles en rapport avec le centre d'attention global (stratégie 3).

Comme on l'a mentionné (section 6.9.2, *"Réaction suite à une incompréhension"*), les stratégies en rapport avec le centre d'attention immédiat s'appliquent en considérant d'abord le dernier segment de texte d'un texte explicatif,

tandis que les stratégies en rapport avec le centre d'attention global s'appliquent en considérant d'abord le premier segment de texte. De plus, on privilégie la première catégorie de stratégies avant la seconde. Ainsi, le prédicat *hein* tente d'abord d'appliquer une stratégie de recouvrement de la première catégorie, et s'il n'y en a pas, il essaye une stratégie de recouvrement de la seconde catégorie. S'il n'y en a pas non plus, le mécanisme d'explications signale à l'utilisateur qu'il n'y a pas de recouvrement possible.

```
hein:-
    recouvrement_categorie1(Arbre),
    ... si nécessaire ...
    recouvrement_categorie2(Arbre),
    ... si nécessaire...
    ... mentionner qu'il n'y a pas de recouvrement possible ...
```

Le recouvrement pour la première catégorie effectue une recherche en commençant par le dernier segment de texte, c'est-à-dire, le dernier acte de discours, c'est-à-dire, le dernier noeud-feuille.

```
recouvrement_categorie1(Arbre) :-
    Arbre = a(Id, Sous_Arbre_Noyau, Sous_Arbre_Satellite),
    recouvrement_categorie1(Sous_Arbre_Satellite),
    recouvrement_categorie1(Sous_Arbre_Noyau),
    ... vérifier s'il y a hypothèse pour ce noeud ...
    traite_hypothese(Id),
    ... sinon, vérifier s'il y a un satellite à développer pour ce noeud ...
    traite_satellite_non_developpe(Id),
```

Quant au recouvrement pour la seconde catégorie, il effectue une recherche en commençant par le premier segment de texte:

recouvrement_categorie2(Arbre) :-
Arbre = a(Id, Sous_Arbre_Noyau, Sous_Arbre_Satellite),
... vérifier s'il y a un autre candidat pour le noeud ...
traite_autre_candidat(Id),
... sinon, passer au noeud suivant ...
recouvrement_categorie2(Sous_Arbre_Noyau),
recouvrement_categorie2(Sous_Arbre_Satellite).

Chacun des prédicats de recouvrement, soit:

traite_hypothese(Id).
traite_satellite_non_developpe(Id).
traite_autre_candidat(Id).

utilise les informations contenues respectivement dans les structures suivantes, que l'on a présentées précédemment, pour savoir si une stratégie donnée s'applique pour le noeud de numéro d'identification *Id*:

noeud_hypothese(Id, Hypothèse).
noeud_satellite_non_developpe(Id, Satellite).
noeud_candidats(Id, [Liste_de_Candidats]).

Ainsi, selon la stratégie de recouvrement applicable, on a que:

- une hypothèse est rendue vraie par l'envoi, au planificateur de texte, du but de communication *Hypothese*, où *Hypothese* est de la forme *connait_qqc(Concept)*,
- un satellite non développé est développé par l'envoi, au planificateur de texte, du but de communication ou du but linguistique contenu dans *Satellite*,
- un autre opérateur candidat est choisi et l'on envoie, au planificateur de texte, l'attribut *Effet* de l'opérateur candidat choisi.

L'annexe de ce mémoire montre un exemple complet de dialogue explicatif obtenu avec le mécanisme d'explications.

CHAPITRE 8
ANALYSE DES RÉSULTATS

L'implantation d'un prototype ayant permis de démontrer la faisabilité du mécanisme d'explications, le prototype devait également être testé pour permettre de vérifier si les objectifs fixés ont été atteints. Rappelons que ces objectifs ont été exprimés en termes de critères de qualité que doivent posséder le mécanisme d'explications et les explications produites. Ces critères ont été identifiés à la section 4.3, *"Type de mécanisme choisi"* et se résument à la phrase suivante:

> *Produire des explications naturelles à l'aide d'un mécanisme d'explications flexible et extensible doté de capacités de réaction et d'adaptation.*

8.1 Type de tests

Le prototype fut testé auprès de cinq utilisateurs dont les connaissances dans le domaine de la tarification de l'électricité correspondent à celles d'un utilisateur de type débutant. Pour cette expérience, le mécanisme d'explications était dans un mode de production de textes explicatifs laconiques. Ainsi, pour chaque explication, l'utilisateur recevait le minimum d'informations. Cela correspond à ne développer aucun des satellites optionnels des opérateurs de planification de texte. L'utilisateur avait accès à un menu de questions pour exprimer son feed-back et obtenir d'autres informations, agissant donc comme guide sur le type d'explication qu'il désirait recevoir. Également, la base de connaissances explicatives ne contenait que les connaissances contenues dans les

dialogues présentés aux Figure 7.1 et Figure 7.4. Ces deux dialogues correspondent respectivement à l'atteinte des deux objectifs suivants pour l'utilisateur:

1. Être convaincu que le tarif M est applicable.
2. Être convaincu de devoir fournir le nombre de jours d'hiver.

8.2 Constatations

Les constatations suivantes sont ressorties de l'expérience avec les cinq utilisateurs:

- Tous les utilisateurs ont affirmé avoir appris quelque chose en ce qui a trait au domaine de connaissances qu'est la tarification de l'électricité.
- Les connaissances acquises n'ont pas toutes été les mêmes pour chacun des utilisateurs, certains préférant obtenir des informations par rapport à un concept plutôt qu'à un autre. Par exemple, certains utilisateurs ont demandé *Qu'est-ce que le tarif M?* alors que d'autres ont demandé *Qu'est-ce qu'une puissance souscrite?*.
- Plus de la moitié des utilisateurs ont affirmé préférer un texte explicatif court, à propos duquel ils peuvent demander des informations supplémentaires au besoin, à un texte explicatif plus long. Les autres ont dit qu'ils auraient aimé avoir l'information tout de suite parce que de toute façon, il a fallu qu'ils la demandent. Ces derniers étaient des utilisateurs n'ayant préalablement aucune connaissance du domaine.

- Certains utilisateurs n'ont pu obtenir totale satisfaction puisque des concepts contenus dans les textes explicatifs n'avaient pas de question de type *Qu'est-ce que...?* au menu de questions (par exemple, *puissance maximale appelée*).

- Certains utilisateurs auraient aimé pouvoir poser d'autres questions. Par exemple, comme feed-back au texte explicatif *Les tarifs applicables sont: tarif M*, certains auraient voulu demander *Pourquoi pas d'autres tarifs?* ou *Pourquoi pas le tarif G?*.

- Plusieurs utilisateurs ont utilisé la question *Hein?* comme type de feed-back, non pas parce qu'ils n'avaient pas bien compris l'explication, mais plutôt parce qu'ils désiraient obtenir plus d'informations.

- À d'autres occasions, plusieurs utilisateurs ont posé la question *Hein?* afin d'obtenir un type de réponse qu'ils préféraient. Par exemple, la relation d'élaboration par définition était la première stratégie utilisée chaque fois qu'une question du type *Qu'est-ce que...?* était demandée. Le feed-back *Hein?* était alors exprimé pour obtenir une élaboration par un exemple, qui était la seconde stratégie.

- Quelques utilisateurs auraient aimé pouvoir spécifier de quelle manière ils désiraient être persuadés lorsqu'une question du type *Pourquoi?* était demandée.

- Tous les utilisateurs ont fait remarquer qu'il n'était pas suffisant de ne pouvoir poser une question qu'en rapport avec le dernier texte explicatif

présenté. Très souvent, une question en rapport avec l'avant dernière explication était désirée. Cela se produit, par exemple, lorsqu'un texte explicatif est présenté et que l'utilisateur désire à la fois poser une question du type *Qu'est-ce que?* (pour mieux connaître un concept) suivie d'une question du type *Pourquoi?*. Pour arriver à leurs fins, les utilisateurs devaient redémarrer le dialogue.

- Quelques utilisateurs ont manifesté le désir d'obtenir non seulement l'information propre à un concept, mais également l'information qui lie deux concepts. Par exemple, les utilisateurs pouvaient obtenir de l'information quant aux concepts *jours d'hiver* et *période de consommation*, mais auraient aimé connaître l'interaction qu'il y a entre ces deux concepts (se faire dire par exemple que la période de consommation peut être exclue, incluse ou en chevauchement avec les jours d'hiver).

- Tous les utilisateurs ont été satisfaits de la vitesse de réaction du mécanisme d'explications, cette vitesse de réaction étant inférieure à une seconde.

- Tous ont su reconnaître que les questions au menu étaient établies selon le contexte, c'est-à-dire, en rapport avec le texte explicatif venant d'être présenté.

- Aucune remarque quant à l'aspect naturel des textes explicatifs ou au niveau de langage employé. On peut donc présumer qu'ils étaient adéquats.

8.3 Analyse

Que peut-on déduire de ces constatations ?

D'abord, le fait que les utilisateurs affirment avoir acquis de nouvelles connaissances permet de conclure que le mécanisme d'explications peut être utilisé à des fins de transfert de connaissances. De plus, en exprimant leur feed-back par une question contextuelle (en rapport avec le texte explicatif affiché) et en pouvant compter sur différentes stratégies de réponse, les utilisateurs avaient donc en leurs mains un mécanisme d'explications doté de capacités de réaction et d'adaptation.

Toutefois, les constatations font également ressortir que la satisfaction des utilisateurs n'est pas totale. L'un des facteurs en cause est dû au manque de connaissances explicatives qui s'est traduit, par exemple, par des concepts non présents dans les questions de type *Qu'est-ce que...?* et par l'absence de liens entre les concepts. Cependant, cette lacune n'est pas imputable à la méthode choisie pour générer les textes explicatifs, mais plutôt aux ressources, telle la base de connaissances explicatives, dont cette méthode a besoin. Il appert donc nécessaire d'élargir la base de connaissances explicatives de façon à incorporer davantage de concepts et à augmenter la richesse de cette base de connaissances explicatives en représentant les liens, le maillage qui existe entre les divers concepts du domaine de connaissances.

Un autre point d'insatisfaction concerne le fait que l'utilisateur soit limité à poser une question uniquement en rapport avec l'explication qui vient de lui être présentée. Cela semble insuffisant puisque les utilisateurs ont à maintes reprises manifesté le désir de

poser une question en rapport avec un texte explicatif antérieur, dans la majorité des cas avec l'avant dernier texte explicatif. Cette observation est corroborée par une étude [26], menée à l'Université d'Ottawa, qui démontre que les réponses fournies par un système d'information, en particulier la dernière et l'avant dernière réponse, ont un effet important sur les questions posées par les utilisateurs. En effet, l'étude, menée auprès de 48 personnes, montre que 46% des questions des utilisateurs sont en rapport avec la dernière réponse et que 62% des questions sont en rapport avec la dernière ou l'avant dernière réponse. Au niveau du mécanisme d'explications que l'on a implanté, cela ne pose pas de problème puisque chaque plan de texte créé peut être conservé. Ainsi, on pourrait permettre à l'utilisateur de poser une question en rapport avec l'avant dernier texte explicatif, et même des textes explicatifs antérieurs.

L'étude fait également ressortir que seulement 23% des questions, donc un peu plus d'une question sur cinq, est une question sans rapport avec les réponses présentées par le système d'information. Cela consolide donc l'approche que l'on a choisie et qui consiste à fournir à l'utilisateur, via un menu par exemple, un ensemble de questions potentielles en rapport avec l'explication qu'on lui présente. Notre propre expérience a cependant démontré que, bien que l'utilisateur désire poser une question en rapport avec le texte explicatif qu'on lui présente, cette question n'apparaît pas toujours au menu de questions. Pour pallier à ce type de problème, on a besoin d'un mécanisme d'explications qui soit extensible. C'est le cas du présent mécanisme d'explications. Effectivement, l'ajout de nouveaux opérateurs de planification de texte ayant des buts de communication liés à ces nouveaux types de questions est une solution. Encore une fois, comme ce fut le

cas avec l'un des problèmes précédents, soit le manque de connaissances explicatives, le problème des questions manquantes est un problème de ressources (opérateurs de planification de texte) et non un problème de méthode.

Également, étant donné que les utilisateurs n'ont pas tous les mêmes requêtes en ce qui a trait aux questions de types *Qu'est-ce que...?*, cela tend à démontrer qu'il est préférable de ne pas surcharger les textes explicatifs et de laisser les utilisateurs demander l'information supplémentaire dont ils ont besoin. Par contre, le fait que certains utilisateurs aient mentionné préférer avoir l'information tout de suite, parce que de toute façon ils vont devoir la demander, penche en faveur d'un mécanisme d'explications plus verbeux. L'une des solutions pourrait résider dans un meilleur modèle de l'utilisateur, qui permettrait de connaître les préférences de cet utilisateur. L'autre solution, qui croît-on est préférable à la première puisqu'elle colle à l'approche selon laquelle l'utilisateur a un rôle de décideur dans le dialogue explicatif, consiste à laisser l'utilisateur indiquer le type de texte explicatif qu'il désire recevoir, à savoir: verbeux ou laconique.

Finalement, une autre constatation est celle concernant l'utilisation de la question *Hein?*. Bien qu'elle ait été utilisée à bon escient dans bien des cas, cette question a également été utilisée pour obtenir une autre explication, non pas parce que l'explication initiale n'avait pas été comprise, mais parce que l'utilisateur n'aimait pas la stratégie employée ou en préférait une autre. Ce phénomène a également été observé avec la question de type *Pourquoi?* dont la réponse tend à persuader l'utilisateur de quelque chose. Dans ce cas, une fois l'explication fournie, l'utilisateur faisait usage de la question *Hein?* pour être persuadé d'une autre façon. Une solution à ce problème, et qui tendrait à

diminuer le nombre d'échanges dans le dialogue, serait de fournir à l'utilisateur l'ensemble des stratégies applicables pour une question donnée. Il faut se demander toutefois si cette approche ne surchargerait pas trop l'utilisateur qui verrait un menu de stratégies s'ajouter à chaque question du menu de questions. Le problème peut être résolu encore une fois en laissant le choix à l'utilisateur d'avoir accès ou non à ces menus de stratégies. Mais au-delà de ces considérations, il y a un désavantage à fournir l'ensemble des stratégies pour chacune des questions, désavantage qui se paye en terme de temps de réaction de la part du système. En effet, déterminer les stratégies applicables à une question revient en fait à trouver tous les opérateurs candidats pour le but de communication associé à la question, et à effectuer une planification de texte (créer un plan de texte) pour chacun d'eux puisqu'il faut s'assurer qu'ils puissent être accomplis. Étant donné que la complexité de l'algorithme de planification de texte est de $O(p^n)$, où p est le nombre d'opérateurs candidats par noeud et n le nombre de noeuds d'un plan de texte, le coût d'un tel traitement est non négligeable si le nombre de questions au menu est grand.

L'expérience a donc montré d'une part que le mécanisme d'explications, basé sur un algorithme de planification de texte à mémoire d'intentions, offre des capacités de réaction et d'adaptation qui permettent à un utilisateur de participer à un dialogue explicatif sans que l'on ait recours à un modèle de l'utilisateur très sophistiqué. Également, la flexibilité et l'extensibilité du mécanisme d'explications sont assurées par l'enrichissement possible de la librairie d'opérateurs de planification de texte et de la base de connaissances explicatives, tandis que l'aspect naturel des explications repose sur

l'utilisation de segments de textes préfabriqués. D'autre part, l'expérience a aussi démontré certaines limites et contraintes du mécanisme d'explications. Le prochain chapitre examine donc les principaux correctifs à apporter pour pallier à ces limites et contraintes.

CHAPITRE 9
CONCLUSION ET VOIES DE RECHERCHE FUTURES

L'analyse des résultats de l'expérience que l'on a effectuée avec un groupe d'utilisateurs a permis de faire ressortir quelques problèmes avec le prototype. Il existe également d'autres problèmes qui ne sont pas ressortis lors de l'expérience, mais dont on est conscient de l'existence et dont on fera mention. Parmi tous ces problèmes, qui sont en fait des limites, des contraintes ou des compromis, certains permettent de mettre en évidence des voies de recherche possibles que l'on identifiera dans les paragraphes qui suivent.

On peut regrouper les problèmes en deux catégories: ceux en rapport avec la méthode et ceux en rapport avec les ressources nécessaires à la méthode. Lorsque l'on parle de la méthode, on fait référence ici à l'algorithme de planification de texte, à la gestion et l'utilisation des structures des intentions et des relations rhétoriques de même qu'aux heuristiques employées. Quant aux ressources, on parle ici de la base de connaissances explicatives, des opérateurs de planification de texte, du modèle de l'utilisateur et de l'historique du dialogue. Certes, quelques-uns des problèmes peuvent être imputables à la méthode, mais comme on le verra, les principaux problèmes sont liés aux ressources.

Ainsi, le principal problème observé est celui de la complétude de la base de connaissances explicatives. On s'aperçoit rapidement, lors d'un dialogue explicatif que

chaque utilisateur peut emprunter une tangente différente qui mène à l'exploration de différents sous-domaines du domaine de connaissances. La représentation de tous les concepts pour couvrir l'ensemble du domaine, de même que la représentation des connaissances de maillage entre les concepts est un travail fastidieux et est en fait le goulot d'étranglement du mécanisme d'explications. Il y a donc des efforts à mettre en oeuvre pour trouver des représentations de connaissances qui facilitent l'expression des connaissances explicatives. On peut également entrevoir la possibilité de développer des méthodes ou des mécanismes pour générer automatiquement les connaissances explicatives, capables de les extraire et les représenter à partir de textes du domaine par exemple.

Malgré que l'on ait mentionné à maintes reprises que l'obtention d'un modèle de l'utilisateur n'était pas une nécessité pour permettre d'entretenir un dialogue explicatif entre un système et un utilisateur de ce système, il n'en demeure pas moins qu'un tel modèle se veut un complément à ne pas négliger. En effet, sans modèle de l'utilisateur, l'algorithme de planification de texte doit émettre beaucoup d'hypothèses quant aux connaissances de l'utilisateur, de sorte que le processus de recouvrement réagit souvent à tort, lorsqu'une explication est mal comprise, en rendant vraie une hypothèse tout simplement parce l'on ne connaît pas ce que l'utilisateur sait et ne sait pas. Il semble donc à première vue que l'une des solutions possibles serait de fournir un modèle le plus complet possible des connaissances de l'utilisateur de façon à limiter les recouvrements où l'on explique des concepts.

En contrepartie, un modèle plus complet tend à augmenter le nombre

d'imprécisions et cela peut mener à des situations problématiques. Par exemple, qu'advient-il si initialement le modèle de l'utilisateur indique que l'utilisateur connaît le concept *prime de dépassement*, alors que dans les faits il n'en est rien? Dans ce cas, il n'y aura aucune hypothèse d'émise et le mécanisme d'explications n'a aucun moyen de savoir qu'en réalité l'utilisateur ne connaît pas le concept *prime de dépassement*. Si l'utilisateur sait pourquoi il n'a pas bien compris l'explication, il lui sera possible de formuler une question du genre *Qu'est-ce qu'une prime de dépassement?*. Si par contre l'utilisateur ne sait pas pourquoi il n'a pas bien compris l'explication et répond par un *Hein?*, cela pose effectivement un problème.

Alors, quel type de modèle de l'utilisateur convient le mieux à la méthode utilisée? Un modèle moins complet mais plus précis, un modèle plus complet mais moins précis, un modèle par stéréotype fixé au départ, un modèle statique, un modèle dynamique, etc. Il y a donc ici place à d'autres travaux de recherche.

L'historique du dialogue peut également devenir une source de problèmes lorsqu'un dialogue explicatif est long. En effet, cet historique du dialogue peut devenir immense, difficile à gérer et mener à une dégradation des performances du mécanisme d'explications. On doit alors se poser des questions, à savoir s'il est possible d'enlever des choses de l'historique du dialogue, et si oui, quand peut-on effectuer ces retraits. Sinon, on doit examiner la possibilité de combiner des informations contenues dans l'historique du dialogue, de manière à diminuer sa taille tout en ne perdant pas d'informations. Également, on doit voir s'il est possible et adéquat de faire passer des informations de l'historique du dialogue vers le modèle de l'utilisateur.

Bien que cela n'ait pas transpiré de l'expérience effectuée, l'utilisation de segments de texte préfabriqués n'est pas aussi naturelle que cela puisse paraître. Si à la sortie les textes ont une allure naturelle, c'est au niveau de la tâche de représentation de chaque segment de texte dans la base de connaissances que ce l'est moins. En effet, la création de chaque segment de texte a nécessité une attention particulière à leur structure pour qu'ils puissent bien s'agencer dans un texte explicatif. Il est donc nécessaire de trouver des moyens pour minimiser cette surcharge de travail en considérant, par exemple, une approche avec générateur de texte en langage naturel. Également, il est important de bien choisir les bonnes structures syntaxique et lexicale, sans quoi l'effet pourrait être inefficace. D'où la nécessité d'avoir un générateur de texte capable de contrôler le niveau de langage requis pour une explication donnée, ce qui n'est pas le cas avec les segments de texte préfabriqués.

Du point de vue de la méthode, il y a également quelques problèmes liés aux diverses stratégies employées ou aux comportements adoptés:

- Quoi faire lorsqu'à une question *Hein?* ne correspond plus aucune stratégie? Vaut-il mieux refournir une réponse ou bien laisser l'utilisateur sans réponse pour ne pas tomber dans la répétition?

- Si lors d'un même dialogue l'utilisateur pose la même question *Qu'est-ce que...?* à propos d'un concept, doit-on conclure que c'est parce qu'il a oublié, auquel cas on pourrait lui répéter la réponse que l'on lui a déjà fournie, ou bien doit-on répondre en utilisant une autre stratégie parce que l'on suppose

qu'il n'a pas bien compris antérieurement (bien qu'il aurait dû à ce moment exprimer un feed-back de type *Hein?*)?

- Comment établir les heuristiques de sélection d'opérateurs candidats et de stratégies de recouvrement? Doivent-elles être établies selon des critères tels que ce qui a été dit, ce que l'utilisateur connaît, la quantité d'informations qu'elles révèlent, etc. ou bien doit-on laisser chaque utilisateur indiquer ses préférences?

- Au menu de questions, doit-on présenter ou non les questions qui ont rapport avec ce qui a déjà été dit ou avec ce que l'utilisateur connaît déjà? En somme, est-ce qu'il est préférable d'alléger la tâche cognitive de l'utilisateur en ne lui présentant pas de questions redondantes ou bien laisser à l'utilisateur le plein contrôle du déroulement du dialogue?

- Est-il préférable de pouvoir poser des questions en rapport avec la dernière et l'avant dernière explication ou de permettre de revenir sur n'importe quelle explication antérieure?

Certains éléments de réponse se trouvent probablement dans l'étude des dialogues entre humains, et des recherches en ce sens pourraient s'avérer bénéfiques pour le type de mécanisme d'explications que l'on a implanté. Cependant, il n'est peut être pas suffisant de copier le dialogue humain. Il faut exploiter ce que peut offrir l'ordinateur. En effet, l'utilisation d'autres médias peut possiblement s'avérer adéquat comme support à une explication (par exemple, l'emploi d'un graphique ou d'un extrait de bande

magnétoscopique).

Il serait également intéressant d'enrichir la variété de textes explicatifs en implantant des opérateurs de planification de texte ayant une structure ou un comportement différent de ceux que l'on a implantés. Par exemple, certaines relations rhétoriques peuvent nécessiter des opérateurs à multinoyaux ou à multisatellites (par exemple, une relation rhétorique de séquence). Également, il existe des relations rhétoriques pour lesquelles il est plus naturel que le texte du satellite apparaisse avant celui du noyau. Cette constatation est ressortie des travaux sur la théorie de la structure rhétorique de Mann et Thompson. Le Tableau 9.1 ci-dessous présente la tendance d'ordonnancement de quelques relations rhétoriques.

Tableau 9.1: Ordonnancement Satellite/Noyau de relations rhétoriques

Satellite **avant** le Noyau	Préalable
	Condition
Satellite **après** le Noyau	Élaboration
	Habilitation
	Évidence

Par exemple, pour une relation rhétorique de condition:

```
Si votre puissance souscrite est inférieure à
100 kW (satellite), vous êtes admissibles au
tarif G (noyau).
```

ou pour une relation d'habilitation, c'est-à-dire, rendre une personne apte à faire quelque chose:

```
Indiquez le nombre de jours d'hiver (noyau). La
période d'hiver se termine le 31 mars (satellite).
```

Il serait intéressant également de tester le mécanisme d'explications dans un contexte de formation où des textes explicatifs composés de plusieurs paragraphes seraient générés. On pourrait dans ce genre d'interaction permettre à l'utilisateur d'interrompre à tout moment l'explication en cours pour demander par exemple des précisions sur un concept avant de poursuivre, cette approche ayant été utilisée par Cawsey [3]. Cela demande donc de générer un texte explicatif au fur et à mesure qu'il est planifié. L'algorithme de planification de texte se prête bien à ce type de comportement puisque sa planification en *profondeur d'abord* correspond à l'ordre des phrases dans le texte explicatif. Il subsiste toutefois un problème relié aux possibles retours-arrières lors d'une planification de texte. En effet, puisque les segments de texte seraient générés au fur et à mesure de leur planification, qu'advient-il dans les cas où l'on doit effectuer des retours-arrières à cause d'impasses rencontrées lors de la planification de texte?

Malgré les limitations et contraintes énumérées, on peut affirmer que la participation d'un utilisateur à un dialogue explicatif basé sur un algorithme de planification de texte à mémoire d'intentions est possible, et ce, sans avoir recours à un modèle complet et précis de cet utilisateur. L'utilisation d'un langage comme Prolog a permis d'obtenir un prototype rapidement. Ses capacités de traitement symbolique et son moteur d'inférence ont facilité grandement l'implantation de l'algorithme de planification

de texte.

En terminant, on constate qu'il y a un lien étroit entre l'architecture d'un mécanisme d'explications et sa capacité à fournir de bonnes explications. Il n'est pas faux de dire que l'évolution des architectures des mécanismes d'explications est en très grande partie redevable aux besoins exprimés en matière d'explication. De meilleures architectures permettent la production de meilleures explications, mais en contrepartie, cela augmente la complexité des mécanismes d'explications. Cette augmentation de la complexité rend à son tour la construction d'explications plus difficile parce que les mécanismes d'explications sont confrontés à des structures complexes et à plusieurs types de connaissances et de ressources pour extraire le contenu explicatif à présenter à l'utilisateur. Mais cela en vaut la peine puisque le type de dialogue explicatif qui en résulte constitue un support à la tâche de réflexion de l'utilisateur. Il rend possible l'apprentissage d'un domaine de connaissances et contribue aussi au transfert de connaissances.

BIBLIOGRAPHIE

1. BRATKO, Y. (1990). Prolog Programming for Artificial Intelligence (second edition, Addison -Wesley. 597p.

2. CARCAGNO, D., THOMAS, I. (1992). Building explanations for an expert system. Twelfth International Confeence on Artificial Intelligence, experts Systems, Natural Language, 1, 93-104.

3. CAWSEY, A. (1989). Explanatory dialogues. Interacting with Computers, 1(1), 69-92.

4. CLANCEY, W.J. (1983). The Epistemology of a Rule-Based Expert System: A Framework for Explanation. Artificial Intelligence, 215-251.

5. CLANCEY, W.J., LETSINGER, R. (1981). NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. Proceeding of the Seventh International Joint Conference on Artificial Intelligence, 829-836.

6. GILBERT, N. (1989). Explanation and dialogue. Knowledge Engineering review, 4(3), 235-247.

7. HOVY, E.H. (1990). Approaches to the Planning of Coherent Text. Natural Language in Artificial Intelligence and Computational Linguistics, Paris, C.L., Swartout, W.R. and Mann, W.C. (eds), 83-102.

8. HOVY, E.H. (1993). Automated Discourse Generation Using Discourse Structure Relations. Artificial Intelligence, 63, Specail Issue on Natural Language Processing.

9. HYDRO-QUÉBEC (1993). Règlement tarifaire-3,T-331, Manuel du participant.

10. HYDRO-QUÉBEC (1995). Questions et réponses sur la tarification. 32p.

11. HYDRO-QUÉBEC (1996). Tarif d'électricité 96. 172p.

12. KARSENTY, L., BRÉZILLON, P. (1995). Coopération Homme-Machine et Explication . Le Travail humain, 58(4), 289-310.

13. LPA-PROLOG Technical Reference (1995). Reference Manual by Logic Programming Associate Ltd. 346p.

14. MANN, W.C., THOMSON, S.A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. TEXT, 8(3), 243-281.

15. MAYBURY, M.T. (1992). Communicative acts for explanation generation. International Journal of Man-Machine Studies, 37, 135-172.

16. MCKEOWN, K.R. (1985). Text Generation: Using discourse strategies and focus constraints to generate natural language text. Cambridge University Press.

17. MCKEOWN, K.R. (1988). Generating goal-oriented explanations. International Journal of Expert Systems Research and Applications, 4, 377-395.

18. MILLET, C., GILLOUX, M. (1989). A study of the knowledge required for explanation in expert systems. Intelligence Applications, 83-90.

19. MOORE, J.D. (1989). A reactive approach to explanation in expert and advice-giving systems. Ph.D. dissertation, University of California in Los Angeles.

20. MOORE, J.D. (1989). Respondings to 'huh?': answering vaguely articulated folow-up questions. SIGCHI Bulletin, 91-96.

21. MOORE, J.D., PARIS, C.L. (1989). Planning text for advisory dialogues. Proceedings of the Twenty-Seventh Annual Meeting of the Association for Computational Linguistics.

22. MOORE, J.D., PARIS, C.L. (1991). Requirements for an expert system explanation facility. Computational Intelligence, 7(4), 367-370.

23. MOORE, J.D., PARIS, C.L. (1992). Exploiting user feedback to compensate for the unreliability of user models. User Modeling and User-Adapted Interaction, 2, 287-330.

24. MOORE, J.D., SWARTOUT, W.R. (1989). A reactive approach to explanation. IJCAI-89 Proceeding of the Eleventh International Joint Conference on Artificial Intelligence, 2, 1504-1510.

25. PARIS, C.L. (1988). Tailoring object descriptions to a user's level of expertise. Computational Linguistics, 14(3), 64-78.

26. PATRICK, A., LOCMELIS, W.J., WHALEN, T. (1993). The Role of Previous Questions and Answers in Natural Language Dialogues with Computers. International Journal of Human-Computer Interaction, 5(2), 129-145.

27. SHORTLIFFE, E.H. (1974). Mycin: A rule based computer program for advising physician regarding anti-microbial therapy selection. Technical Report AI Lab Memo AIM-250. Standford University.

28. SPARCK, J.K. (1989). Realism about user modelling. User models in Dialog Systems, Springer-Verlag, 341-363.

29. Special Edition Using Delphi 2 (1996). Reference Manual by Que Corporation. 891p.

30. SUTHERS, D.D. (1991). A task-appropriate hybrid architecture for explanation. Computational Intelligence, 7(4), 315-333.

31. SWARTOUT, W.R. (1983). Xplain: A System for Creating and Explaining Expert Consulting Programs. Artificial Intelligence, 283-325.

32. SWARTOUT, W.R. (1990). Evaluation criteria for expert system explanation. Proceedings of the AAAI-90 Workshop on Evaluatin of Natural Language Generation Systems.

33. SWARTOUT, W.R., MOORE, J.D. (1993). Explanation in Second Generation Expert Systems. Second Generation Expert Systems, Springer Verlag, 543-585.

34. WALLIS, J.W., SHORTLIFFE, E.H. (1984). Customized explanations using causal knowledge, Rule-Based Expert System: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, 371-388.

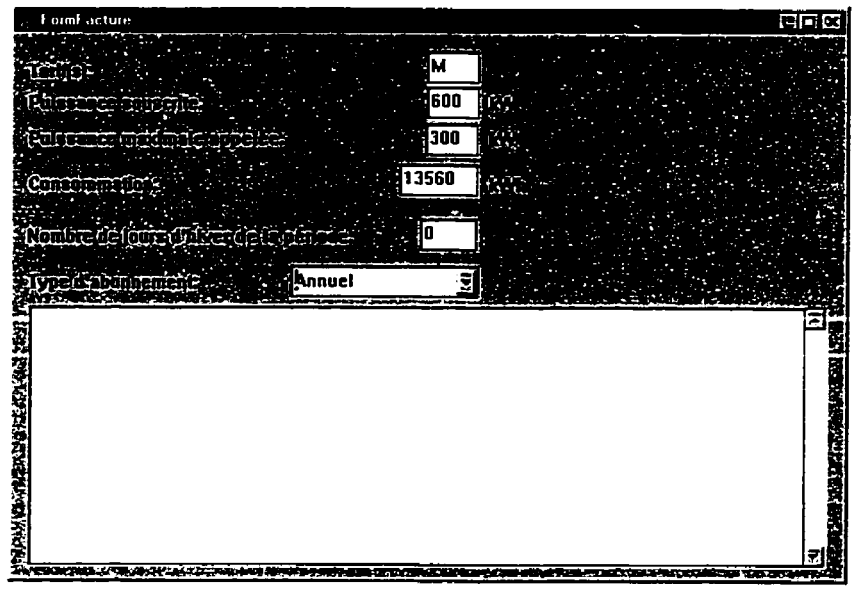
35. WHITE, F. (1994). The user interface of expert systems: what recent research tells us. Library Software Review, 13(2), 91-98.

36. WICK, M.R., THOMPSON, W.B. (1992). Reconstructive expert system explanation. Artificial Intelligence, 54(1-2), 33-70.

ANNEXE

Cette annexe présente un exemple complet de dialogue explicatif réalisé entre un utilisateur et le prototype afin d'illustrer le comportement du mécanisme d'explications, et par le fait même, ses capacités de réaction. On ne décrira pas en détails, à chaque fois, tout ce que fait l'algorithme de planification de texte à chacune des étapes du dialogue. On fera plutôt ressortir les comportements, les types de prise de décision et les situations qui n'auront pas encore été rencontrés jusqu'alors dans le dialogue.

L'état initial de l'interface personne-ordinateur est illustré ci-dessous:



L'interface personne-ordinateur est composée d'une série de champs d'entrée et d'une zone de dialogue. C'est dans cette zone de dialogue qu'est affiché chaque réponse ou texte explicatif du système (préfixé par les caractères *RS* pour *réponse-système*) et chaque feed-back ou question de l'utilisateur (préfixé par les caractères *QU* pour *question-*

utilisateur). À noter qu'aucun test d'utilisabilité de l'interface n'a été réalisé auprès des utilisateurs.

Le dialogue explicatif s'amorce lorsque l'utilisateur clique avec le bouton gauche de la souris sur l'un des champs d'entrée. Pour le présent exemple, le dialogue est amorcé par le champ *Tarif(s)*. Il s'agit d'un dialogue explicatif visant à convaincre l'utilisateur que, dans son cas, seul le tarif M est applicable.

Ainsi donc, lorsque l'utilisateur clique sur le champ *Tarif(s)*, le but de communication suivant, exprimé sous forme de prédicat Prolog, est envoyé de l'interface personne-ordinateur à l'algorithme de planification de texte:

connait_qqc(concept(`tarifs applicables`))

Il s'agit ici d'un but de communication dont l'intention est de faire connaître quelque chose à l'utilisateur à propos du concept *tarifs applicables*. La première étape consiste à trouver tous les opérateurs dont l'effet correspond au but à accomplir. La librairie d'opérateurs de planification de texte en comporte quatre:

```
op_pt(faire_connaitre_par_exemple,
      connait_qqc(concept(Ident)),
      [ident2concept(Concept, Ident)],
      elaboration_par_exemple(concept(Concept)),
      nil, nil).

op_pt(faire_connaitre_par_circonstance,
      connait_qqc(concept(Ident)),
      [ident2concept(Concept, Ident)],
      elaboration_par_circonstancielle_de_temps(concept(Concept)),
      nil, nil).

op_pt(faire_connaitre_par_definition,
      connait_qqc(concept(Ident)),
```

```
        [ident2concept(Concept, Ident)],
        elaboration_par_definition(concept(Concept)),
        nil, nil).

op_pt(faire_connaitre_par_ensemble_membre,
        connait_qqc(concept(Ident)),
        [ident2concept(Concept, Ident)],
        elaboration_par_ensemble_membre(concept(Concept)),
        nil, nil).
```

Chacun de ces opérateurs a pour seule contrainte de s'assurer que le concept, que l'on veut faire connaître, existe dans la base de connaissances explicatives. Le noyau de chacun de ces opérateurs est un but rhétorique représentant une relation rhétorique d'élaboration:

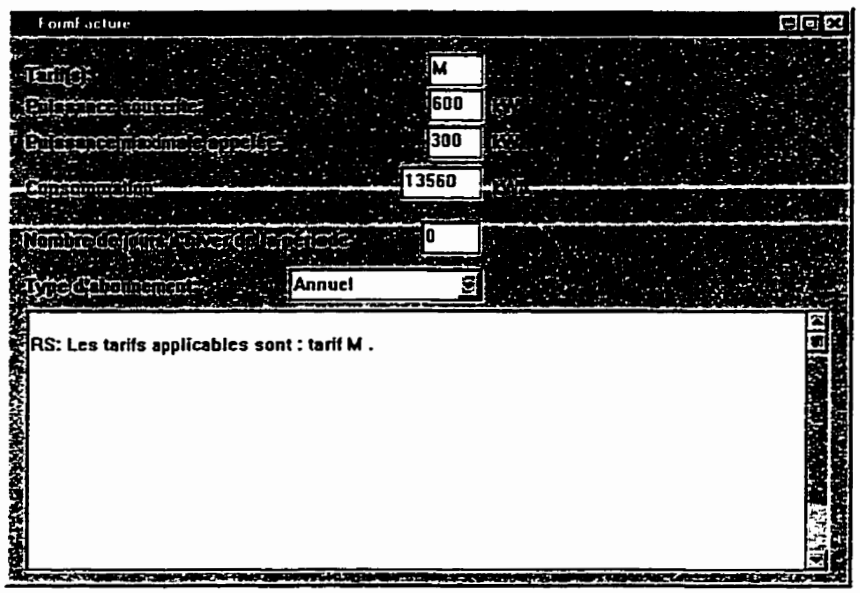
```
op_pt(elaboration_par_exemple,
        elaboration_par_exemple(concept(Concept)),
        [Concept(exemple, Exemple)],
        informer(exemple(Concept, Exemple)),
        nil, nil).

op_pt(elaboration_par_circonstancielle_de_temps,
        elaboration_par_circonstancielle_de_temps(concept(Concept)),
        [Concept(circ_temps, Circ_temps)],
        informer(circ_temps(Concept, Circ_temps)),
        nil, nil).

op_pt(elaboration_par_definition,
        elaboration_par_definition(concept(Concept)),
        [Concept(definition, Definition)],
        informer(definition(Concept, Definition)),
        nil, nil).

op_pt(elaboration_par_ensemble_membre,
        elaboration_par_ensemble_membre(concept(Concept)),
        [ensemble_membre(Concept, Membres)],
        informer(ensemble_membre(Concept, Membres)),
        nil, nil).
```

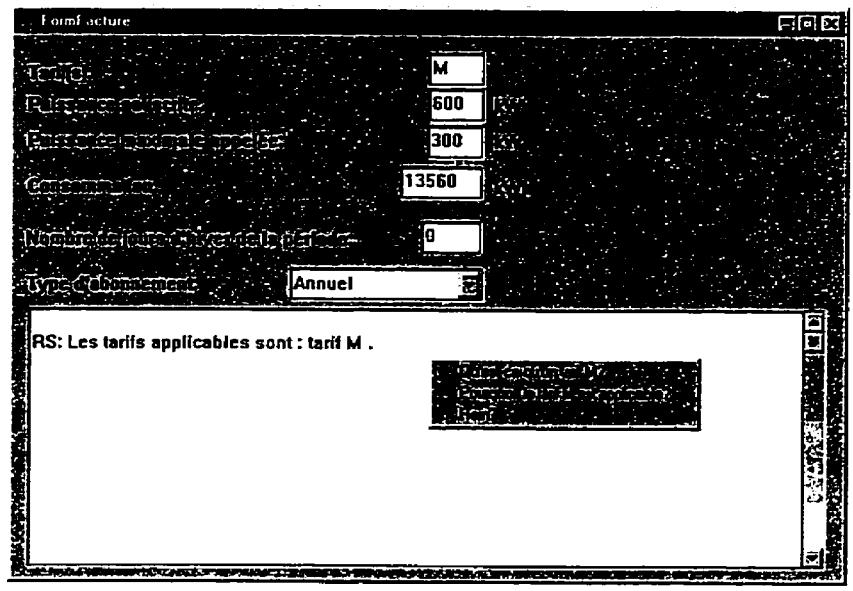
Toutefois, la base de connaissances explicatives ne contenant ni exemple, ni information circonstancielle pour le concept *tarifs applicables*, cela étant détecté lors de la vérification des contraintes, les deux premiers opérateurs (*elaboration_par_exemple* et *elaboration_par_circonstancielle_de_temps*) sont alors rejetés. Les deux autres opérateurs sont quant à eux des opérateurs candidats puisque leurs contraintes sont satisfaites. Étant donné l'ordre de préséance que l'on a choisi d'accorder aux stratégies, l'algorithme choisit d'abord la stratégie d'élaboration par ensemble-membre. Cette stratégie permet d'obtenir la liste des tarifs applicables grâce à l'appel au prédicat **ensemble_membre(Concept, Membres)** via l'opérateur *informer* (noyau de l'opérateur *elaboration_par_ensemble_membre*), puis de fournir la réponse ci-dessous:



L'autre opérateur candidat est conservé dans la liste des opérateurs candidats pour le noeud correspondant du plan de texte:

noeud_candidats(Id, [elaboration_par_definition]).

Suite à cette réponse, un menu de questions de feed-back est construit, menu que l'utilisateur peut afficher en plaçant le curseur de la souris dans la zone de dialogue puis en cliquant sur le bouton droit de la souris:



Supposons que l'utilisateur ait choisi la question *Pourquoi le tarif M est applicable?*. Cela représente une demande de persuasion de la proposition *le tarif M est applicable*, demande que l'on réalise au moyen d'un opérateur de persuasion par évidence. Dans le présent contexte, le but de communication envoyé à l'algorithme de planification de texte est un but de persuasion par applicabilité d'un concept:

persuade(applicabilite(tarif_M)).

Dans la librairie d'opérateurs, il n'existe qu'un seul opérateur dont l'effet

correspond à ce but de communication:

op_pt(persuader_par_evidence,
persuade(applicabilite(Concept)),
[Concept(applicabilite, Applicabilite),
Concept(attribut_applicabilite, Attribut_Applicabilite),
connait_qqc(concept(Concept))],
informer(valeur_applicabilite(Concept, attribut_applicabilite)),
evidence(applicabilite(Concept)), optionnel).

Cet opérateur possède trois contraintes: les deux premières en rapport avec la base de connaissances explicatives, l'autre en rapport avec le modèle de l'utilisateur. Les deux premières contraintes sont satisfaites puisque la base de connaissances explicatives contient les faits suivants pour le concept *tarif_M*:

tarif_M(applicabilite, `un tarif qui s'applique a tout abonnemment qui a une &(puissance souscrite) d'au moins 100 kW mais inferieure a 5000 kW`).

tarif_M(attribut_applicabilite, puissance_souscrite).

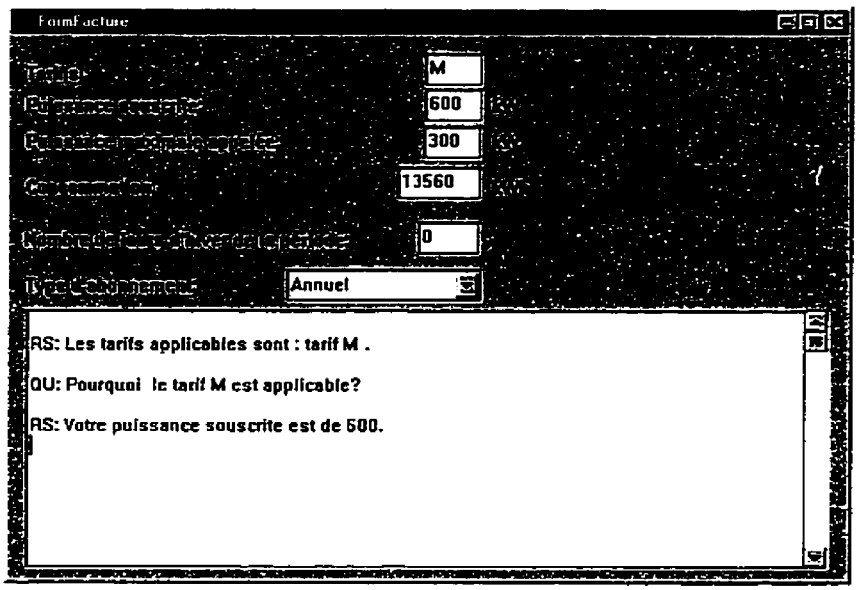
La contrainte en rapport avec le modèle de l'utilisateur n'est quant à elle pas satisfaite puisqu'initialement, ce modèle ne contient aucune des connaissances de l'utilisateur. Une hypothèse est donc émise pour le noeud courant:

noeud_hypothese(Id, connait_qqc(concept(tarif_M))).

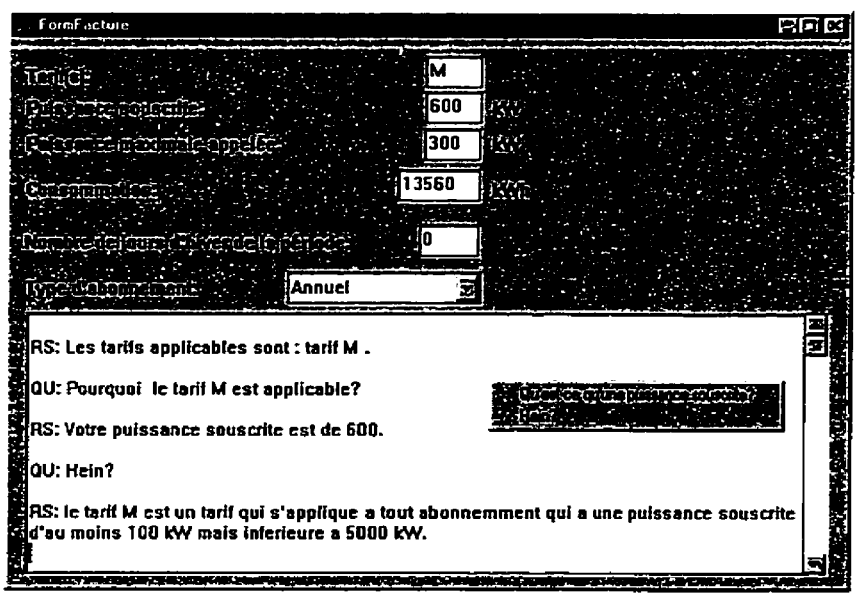
Finalement, puisque l'opérateur contient un satellite optionnel, alors que le mécanisme d'explications est en mode de génération de textes explicatifs laconiques, ce satellite n'est pas développé et le fait suivant est ajouté pour le noeud courant:

noeud_satellite_non_developpe(Id, evidence(applicabilite(tarif_M))).

Le noyau de l'opérateur *persuader_par_evidence*, qui est un acte de discours de type *informer*, est alors réalisé et permet de fournir la réponse suivante:



Puis le menu de questions suivant est alors généré:



Supposons maintenant que l'utilisateur exprime un feed-back de type *Hein?*. Selon son processus de recouvrement, le mécanisme d'explications vérifie d'abord s'il y a eu une hypothèse d'émise ou s'il existe un satellite optionnel qui n'a pas été développé. Dans le cas présent, ces deux possibilités existent. Cependant, étant donné que l'on ne possédait aucun modèle de l'utilisateur en ce qui a trait à ses connaissances du domaine, on a décidé de privilégier la stratégie de recouvrement qui consiste à développer un satellite optionnel non développé, avant celle qui consiste à rendre vraie une hypothèse. Cela permet d'éviter d'avoir recours inutilement à cette dernière stratégie puisque l'utilisateur peut posséder plusieurs connaissances du domaine que l'on ignore. Ainsi donc, un plan de texte est alors construit pour réaliser le satellite non développé:

evidence(applicabilite(tarif_M)).

Un seul opérateur de planification de texte a un effet correspondant:

```
op_pt(evidence,
      evidence(applicabilite(Concept)),
      [Concept(applicabilite, Applicabilite) ],
      croit(applicabilite(Concept)),
      nil, nil).
```

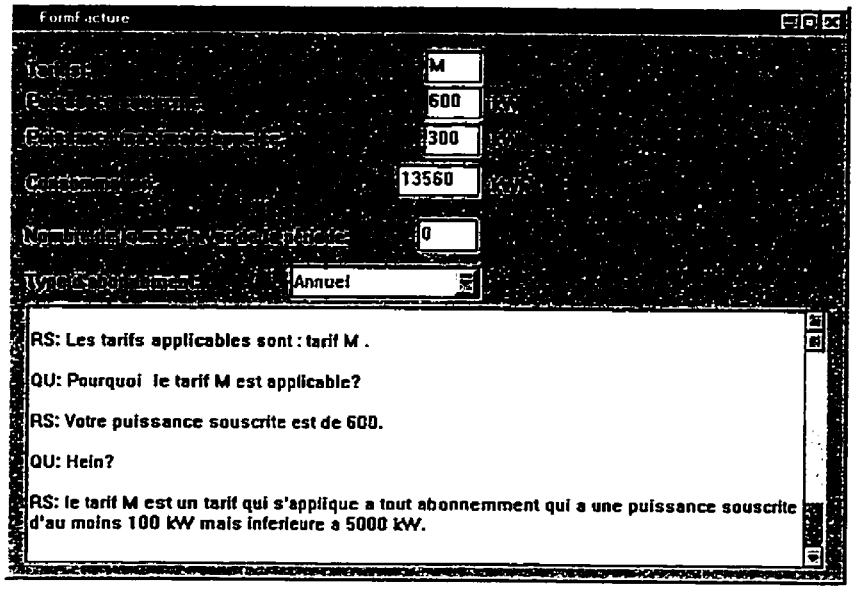
Le noyau de cet opérateur est une intention visant à faire croire l'évidence à l'utilisateur:

```
op_pt(croit,
      croit(applicabilite(Concept)),
      [ ],
      informer(applicabilite(Concept)),
      nil,nil).
```

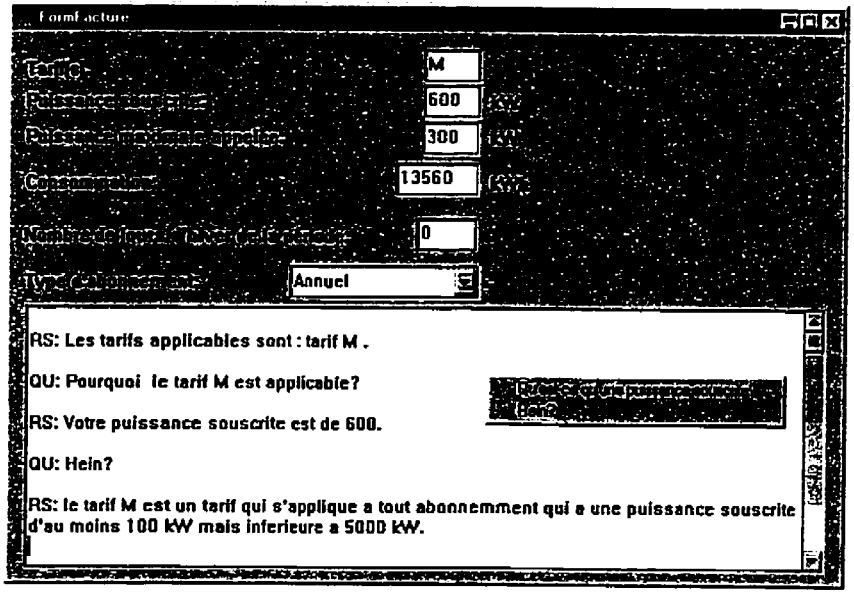
Puisque le base de connaissances explicatives contient ceci:

tarif_M(applicabilite, `un tarif qui s'applique a tout abonnemment qui a une &(puissance souscrite) d'au moins 100 kW mais inferieure a 5000 kW`).

la réponse suivante est fournie:



Un nouveau menu de questions est alors généré:



Cette fois l'utilisateur décide de demander *Qu'est ce qu'une puissance souscrite?*. On se rend alors compte que l'on aurait peut-être dû inverser l'ordre de préséance des stratégies de recouvrement précédemment envisagées, c'est-à-dire, rendre vraie l'hypothèse au lieu de développer le satellite non optionnel puisque l'utilisateur demande maintenant ce qu'est une puissance souscrite. Mais comme on peut le constater, les capacités de réaction du mécanisme d'explications et la participation de l'utilisateur au dialogue explicatif font que cela n'a guère de conséquences puisque l'utilisateur parvient tout de même à ses fins. Le but de communication suivant est donc envoyé au planificateur de texte:

connait_qqc(concept(`puissance souscrite`)).

Comme on l'a vu au début du dialogue, il existe quatre opérateurs dont l'effet correspond à ce but, lesquels mènent à la réalisation de buts rhétoriques d'élaboration. En examinant la base de connaissances explicatives, on remarque que seules les relations rhétoriques d'élaboration par définition et d'élaboration par un exemple peuvent être utilisées:

puissance_souscrite(definition, `la puissance minimale a payer pour un abonnement`).

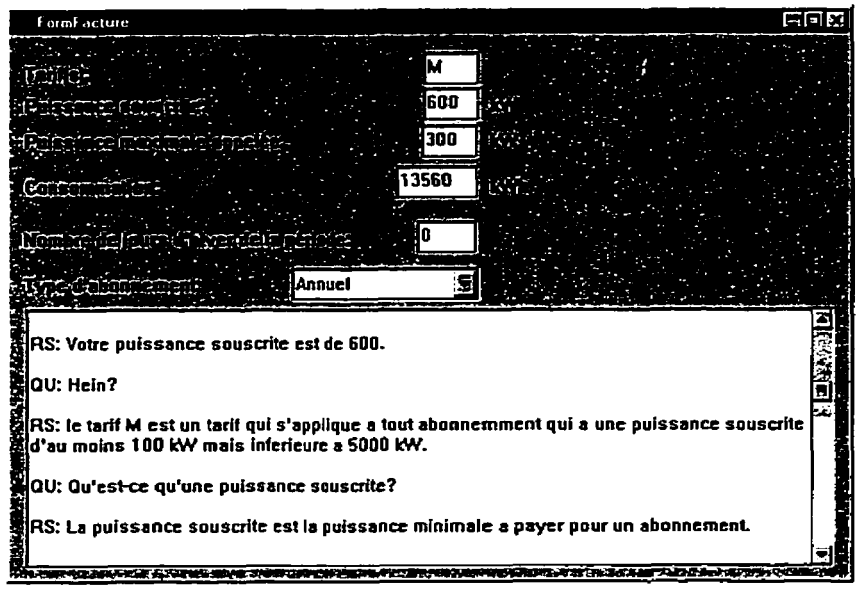
puissance_souscrite(exemple, `si vous prenez une puissance souscrite de 600 kW, vous devrez payer une prime de puissance minimale pour 600 kW meme si votre appel maximale de puissance nà ete que de 300 kW`).

Selon l'ordre de préséance que l'on a établi, l'algorithme de planification de texte choisit d'abord, parmi les opérateurs candidats, l'opérateur d'élaboration par définition avant l'opérateur d'élaboration par un exemple. L'autre opérateur candidat est

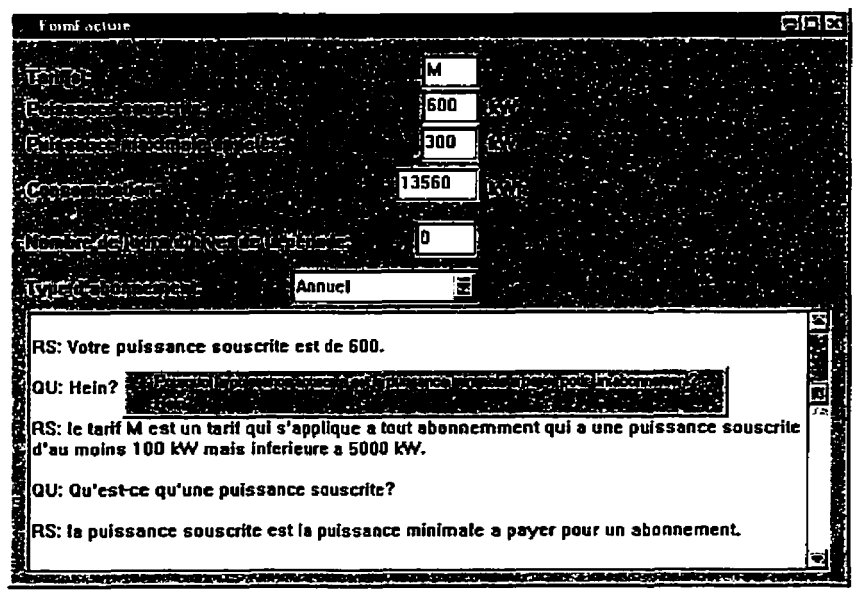
conservé dans la liste des opérateurs candidats pour le noeud correspondant du plan de texte:

noeud_candidats(Id, [elaboration_par_un_exemple]).

L'explication suivante est alors fournie à l'utilisateur:



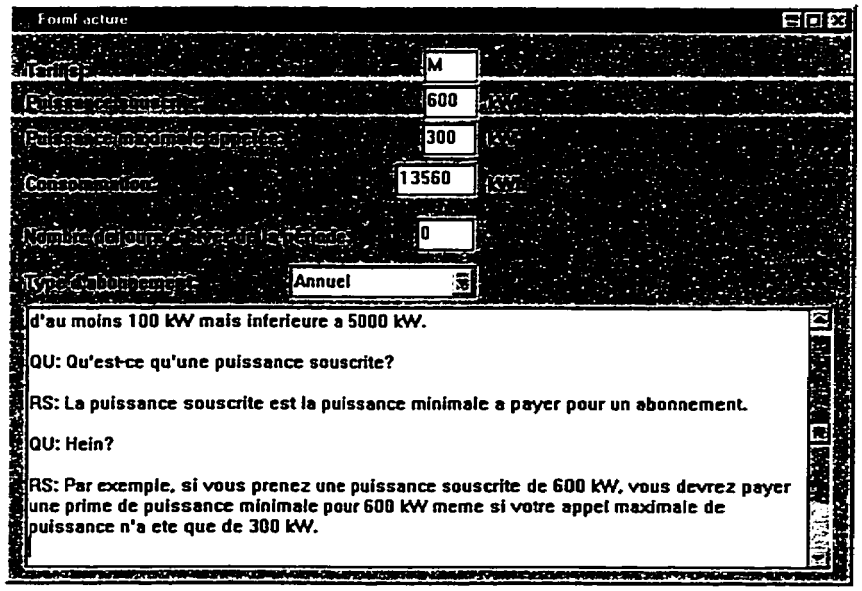
Un nouveau menu de questions est ensuite généré:



Ensuite, l'utilisateur choisit le feed-back *Hein?*. Cette fois, aucune hypothèse n'a été émise et il n'existe aucun satellite optionnel non développé, de sorte que le processus de recouvrement cherche alors à trouver un noeud du plan de texte pour lequel il existe un ou plusieurs opérateurs candidats n'ayant pas été utilisés. On s'aperçoit alors que l'on peut utiliser une relation rhétorique d'élaboration par un exemple pour recouvrer la stratégie d'élaboration par définition. Sachant que ceci est contenu dans la base de connaissances explicatives:

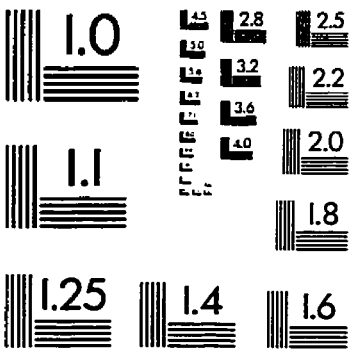
> **puissance_souscrite(exemple, `si vous prenez une puissance souscrite de 600 kW, vous devrez payer une prime de puissance minimale pour 600 kW meme si votre appel maximale de puissance nà ete que de 300 kW').**

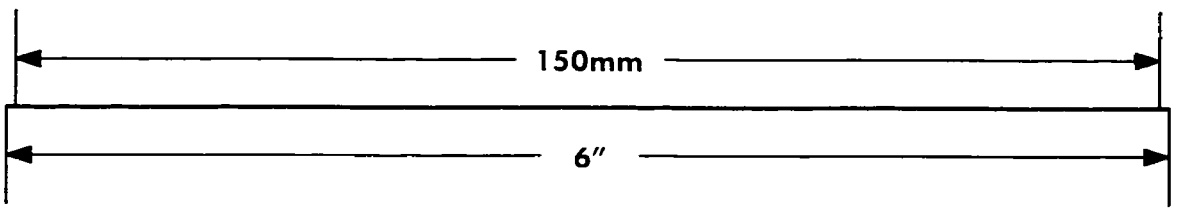
la réponse suivante est fournie et représente, dans ce cas-ci, la dernier texte explicatif du dialogue:



Ce qui termine la présentation de l'exemple.

TEST TARGET (QA-3)





APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved