Application de méthodes d'analyse linguistique à la gestion d'une base de nomenclatures : le logiciel PIAFBCN

Annie Culet

THESE

présentée à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR INGENIEUR

par

Annie CULET

APPLICATION DE METHODES D'ANALYSE LINGUISTIQUE A LA GESTION D'UNE BASE DE NOMENCLATURES: LE LOGICIEL PIAFBCN

Thèse soutenue le 30 octobre 1981 devant la commission d'examen.

C. DELOBEL	Président
Y. CHIARAMELLA V. JOLOBOFF G. VEILLON	Examinateurs
M. YVAIN	Invité

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1979-1980

Président : M. Philippe TRAYNARD

Vice-Présidents : M. Georges LESPINARD
M. René PAUTHENET

PROFESSEURS DES UNIVERSITES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	BENOIT Jean	Radioélectricité
	BESSON Jean	Chimie Minérale
	BLIMAN Samuel	Electronique
	BLOCH Daniel	Physique du Solide - Cristallographie
	BOIS Philippe	Mécanique
	BONNETAIN Lucien	Génie Chimique
	BONNIER Etienne	Métallurgie
	BOUVARD Maurice	Génie Mécanique
	BRISSONNEAU Pierre	Physique des Matériaux
	BUYLE-BODIN Maurice	Electronique
	CHARTIER Germain	Electronique
	CHERADAME Hervé	Chimie Physique Macromoléculaires
Mme	CHERUY Arlette	Automatique
MM.	CHIAVERINA Jean	Biologie, Biochimie, Agronomie
	COHEN Joseph	Electronique
	COUMES André	Electronique
	DURAND Francis	Métallurgie
	DURAND Jean-Louis	Physique Nucléaire et Corpusculaire
	FELICI Noël	Electrotechnique
	FOULARD Claude	Automatique
	GUYOT Pierre	Métallurgie Physique
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du Solide - Cristallographie
	LACOUME Jean-Louis	Géographie - Traitement du Signal
	LANCIA Roland	Electronique - Automatique
	LESIEUR Marcel	Mécanique
	LESPINARD Georges	Mécanique
	LONGEQUEUE Jean-Pierre	Physique Nucléaire Corpusculaire
	MOREAU René	Mécanique
	MORET Roger	Physique Nucléaire Corpusculaire
	PARIAUD Jean-Charles	Chimie - Physique
	PAUTHENET René	Physique du Solide - Cristallographie
	PERRET René	Automatique

.../...

MM.	PERRET Robert	Electrotechnique
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	POLOUJADOFF Michel	Electrotechnique
	POUPOT Christian	Electronique - Automatique
	RAMEAU Jean-Jacques	Chimie
	ROBERT André	Chimie Appliquée et des matériaux
	ROBERT François	Analyse numérique
	SABONNADIERE Jean-Claude	Electrotechnique
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie - Physique
Mme	SCHLENKER Claire	Physique du Solide - Cristallographie
MM.	TRAYNARD Philippe	Chimie - Physique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNY François	Electronique

CHERCHEURS DU C.N.R.S. (Directeur et Maître de Recherche)

M.	FRUCHART Robert	Directeur de Recherche
MM.	ANSARA Ibrahïm	Maître de Recherche
	BRONOEL Guy	Maître de Recherche
	CARRE René	Maître de Recherche
	DAVID René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	KAMARINOS Georges	Maître de Recherche
	KLEITZ Michel	Maître de Recherche
	LANDAU Ioan-Doré	Maître de Recherche
	MERMET Jean	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

E.N.S.E.E.G.

MM.	ALLIBERT Michel	
	BERNARD Claude	
	CAILLET Marcel	
Mme	CHATILLON Catherine	
MM.	COULON Michel	
	HAMMOU Abdelkader	
	JOUD Jean-Charles	
	RAVAINE Denis	
	SAINFORT	C.E.N.G.

.../...

MM. SARRAZIN Pierre
SOUQUET Jean-Louis
TOUZAIN Philippe
URBAIN Georges — Laboratoire des Ultra-Réfractaires ODEILLO

E.N.S.M.E.E.

MM. BISCONDI Michel
BOOS Jean-Yves
GUILHOT Bernard
KOBILANSKI André
LALAUZE René
LANCELOT Françis
LE COZE Jean
LESBATS Pierre
SOUSTELLE Michel
THEVENOT François
THOMAS Gérard
TRAN MINH Canh
DRIVER Julian
RIEU Jean

E.N.S.E.R.G.

MM. BOREL Joseph
CHEHIKIAN Alain
VIKTOROVITCH Pierre

E.N.S.I.E.G.

MM. BORNARD Guy
DESCHIZEAUX Pierre
GLANGEAUD François
JAUSSAUD Pierre
Mme JOURDAIN Geneviève
MM. LEJEUNE Gérard
PERARD Jacques

E.N.S.H.G.

M. DELHAYE Jean-Marc

E.N.S.I.M.A.G.

MM. COURTIN Jacques
LATOMBE Jean-Claude
LUCAS Michel
VERDILLON André

Je tiens à remercier :

Monsieur C. DELOBEL, Professeur à l'Université Scientifique et Médicale de Grenoble, Directeur du Laboratoire IMAG qui m'a fait l'honneur de présider le Jury de cette Thèse.

Monsieur G. VEILLON, Professeur à l'Institut National Polytechnique de Grenoble, Directeur de l'ENSIMAG qui a assuré la direction de cette Thèse et dont les critiques et les conseils m'ont été précieux.

Monsieur V. CHIARAMELLA, Maître-Assistant à l'Institut Universitaire de Technologie B, pour l'intérêt qu'il a toujours porté à ces travaux et la part importante qu'il y a prise. Je tiens à le remercier tout particulièrement pour les efforts et le temps qu'il m'a consacrés, ses encouragements constants et le soutien amical et confiant qu'il m'a accordé.

Monsieur V. JOLOBOFF, Ingénieur de Recherche à la Compagnie CII-HB qui a accepté de juger cette Thèse et de participer au Jury. Je tiens à lui exprimer toute ma gratitude pour ses nombreux conseils et la disponibilité dont il a fait preuve pour m'aider.

Monsieur J. LEBEAU, Ingénieur à la Société THOMSON-CSF, qui m'a fourni les informations pratiques et dont la collaboration a permis l'expérimentation des logiciels. Je suis heureuse de le remercier ici pour l'amitié qu'il m'a toujours témoignée.

Je voudrais remercier aussi tous ceux qui m'ont aidé dans mes recherches, notamment Monsieur M. LEVY, Madame M.F. BRUANDET et Monsieur M. DYMETMAN ainsi que Monsieur M. YVAIN qui est à l'origine de cette étude.

Je remercie enfin Mademoiselle V. CATRIS qui a assuré la dactylographie de cette Thèse, ainsi que le service de reproduction qui en a réalisé le tirage.

راه البرق عظيم بين التفاح والرمانة.
واش ما فرق بين انت، وانت، وانت، وانا.

ناس الغيوان

(Nass el Ghiwane)

Table des matières

CHAPITRE IV. CONCLUSION

CHAPITRE 0

SITUATION ET OBJECTIFS

0.1 INTRODUCTION

0.1.1 Présentation générale du problème

L'objet de cette étude est la constitution d'une base de données contenant des références à des objets du monde réel et la définition de méthodes de recherche à partir d'informations partielles ou floues relatives à ces références.

La référence attachée à un objet est un ensemble de valeurs de caractéristiques, exprimées au moyen d'un certain langage défini par le créateur de l'objet. Cette référence, en tant que désignation, doit donc être utilisée pour répertorier l'objet mais elle peut en constituer aussi une description (description "standard"). Les objets étant de type et de provenance divers, les langages de désignations sont nombreux et variés.

On constate malheureusement, qu'il est souvent difficile d'exprimer correctement ces désignations que ce soit au niveau de la constitution de la base ou au niveau de sa consultation.

- La phase de saisie Celle-ci est déterminante pour les recherches futures. En effet, les nombreuses erreurs d'origines diverses peuvent survenir lors de cette opération ; elles introduisent des incohérences entre les informations (plusieurs descriptions pour un même objet, éventualité de ne pas pouvoir retrouver l'objet cherché ou d'en retrouver plusieurs).

  Il est donc important de pouvoir définir des critères de fiabilité des informations lors de la saisie.

  Cette opération devra être réalisée conformément aux descriptions "standard" en tenant compte du fait que celles-ci sont en général mal connues des personnes qui devront effectuer cette tâche (les difficultés proviennent de la diversité et de la complexité des langages utilisées).

  L'analyse préalable des descriptions, en vue d'un contrôle, est donc une étape essentielle de la saisie.

- La recherche Lors de la recherche, la description proposée pour la sélection peut être aussi sujette à des erreurs ou bien, sans être erronnée, être

incomplète ou différente de celle adoptée initialement à la saisie. Pour ces raisons, il peut être difficile, voire impossible de faire un rapprochement direct entre une question et la référence à un objet (résultat ambigü).

Cette situation est analogue à celle rencontrée dans les systèmes documentaires, aussi, bien que l'on se trouve dans un contexte de données différentes, nous utilisons des techniques classiques dans ce domaine (recherche par mots-clés) en tentant d'obtenir une couverture du résultat réel, une méthode linguistique étant utilisée pour l'analyse des références et des questions.

Le point de départ de cette étude a été une application sur une base de données de nomenclatures et nous précisons son cadre dans la suite de ce chapitre.

0.1.2 Cadre de l'expérience

Ce travail a été entrepris à la suite d'une demande d'étude provenant du Bureau Central de Numération (BCN) de la Société Thomson-CSF. Ce service possède un fichier des composants nécessaires aux fabrications réalisées par les différentes unités de Thomson. L'ensemble des informations associées à un composant constitue un article et l'information d'accès à un article est sa désignation ou référence-article, celle-ci étant élaborée par le fabricant du composant.

Une référence-article se présente sous la forme d'un texte très court permettant de connaître les caractéristiques d'un composant, soit parce qu'elles apparaissent plus ou moins explicitement dans le texte lui-même soit par référence à un catalogue.

Ces références-articles, étant créées par des organismes différents ne présentent aucune cohérence entre elles dans leur expression (formalismes différents). Les raisons qui conduisent un fabricant à choisir un type de représentation sont diverses : elles peuvent être historiques, liées au processus de fabrication, contraintes par l'existence d'une norme pour le composant...

Le service BCN affecte à chaque référence-article un numéro : celui-ci est la référence interne (propre à l'entreprise) du composant. Cette opération se fait à la suite d'une demande provenant d'une unité de fabrication lorsque

la nomenclature d'un objet a été réalisée.

Exemples simplifiés d'articles

référence interne	nom du fabricant (réel ou norme)	dénomination du composant	référence-article
99034622	AFNOR	VIS	VIS CHC M5x35/17 U ACIER QUALITE 10.9 CD8B
99086280	JEDEC	DIODE	1N5473A
91326756	RAYCHEM	CABLE	44/1132-20-0/2/9-9

Ces exemples illustrent diverses formes de références-articles. Dans la référence-article de la VIS, les caractéristiques principales (forme, dimension, matière...) sont données explicitement c'est-à-dire qu'un lecteur ayant une connaissance générale des composants métallurgiques (normes) pourra les interpréter directement.

Par contre, dans l'exemple du CABLE, celles-ci sont données sous une forme codée. Pour interpréter cette référence-article, il faut connaître les règles du codage propres au fabricant et pour ce faire se référer au catalogue "FILS et CABLES de type 44" publié par celui-ci. Le lecteur pourra alors découvrir, par exemple, que le chiffre "2" en 7ième position signifie "âme en cuivre argenté".

Enfin, l'exemple de la diode est celui d'une référence-article n'ayant aucune interprétation : les caractéristiques de ce composant sont décrites dans le catalogue du fabricant et ne peuvent être déduites de l'analyse de la référence.

D'autres exemples d'articles sont donnés en Annexe VI

0.1.3 Expériences actuelles dans ce domaine

Toutes les grosses sociétés qui achètent des produits à l'extérieur ou des organismes centralisateurs (tel l'OTAN) sont confrontés à ces problèmes.

A notre connaissance, dans le domaine des bases de nomenclatures, les méthodes

de classification sont restées manuelles et les méthodes de recherche demeurent peu sophistiquées. Elles deviennent rapidement de plus en plus impraticables et les résultats de moins en moins satisfaisants lorsque le nombre des produits s'accroît au cours des années.

Citons par exemple le cas de Thomson : le fichier-articles contient actuellement environ 400000 produits. La société dispose d'un service d'une quinzaine de personnes pour effectuer la classification des produits nouveaux, chacun étant spécialiste d'un domaine technologique particulier. Le travail est très fastidieux (consultation incessante de catalogues) et la qualité des informations médiocre (beaucoup de produits déclassés, d'erreurs dans les désignations des produits, d'incohérences...).

Il s'est donc créé une certaine "situation d'urgence" -urgence de satisfaction de besoins- qui a conduit à rechercher d'autres méthodes (transposer au niveau d'un logiciel l'essentiel de l'"assurance de qualité" qui aujourd'hui repose sur les seuls opérateurs).

0.1.4 Domaines proches

Cette étude est orientée, dans sa conception, vers la terminologie. Alors que la linguistique est la théorie de la langue en général, la science de la terminologie est la théorie des langages scientifiques ou techniques. Son principe est d'unifier des notions et des termes.

Les besoins terminologiques se manifestent dans le cadre général du besoin de nommer ; ils concernent essentiellement la nécessité de construire (élaboration de terminologies), de contrôler (normalisation) et de communiquer un ensemble de connaissances.

Normaliser signifie faire admettre et respecter un système de valeurs ; certains éléments de la pratique (termes) sont sélectionnés ou proposés pour constituer une référence et une règle (norme). La normalisation a un rôle de régularisation ; c'est un élément correcteur et fixateur du langage.

Le travail terminologique a un caractère international (INFOTERM). La normalisation de la terminologie revet cependant une telle importance que d'une façon tout à fait générale, elle est aujourd'hui assurée également par des organisations techniques nationales (ex. AFNOR), internationales (ex. ISO),

des entreprises (ex. SIEMENS) et des administrations dans leur domaine propre.

Les banques de données terminologiques rendent disponibles la terminologie normalisée et peuvent aussi être utilisées comme outil d'aide à la traduction (NORMATERM [LAU], TEAM [SCH]).

Le domaine de la terminologie est donc proche de notre étude :

- à la saisie, les descriptions sont normalisées selon une certaine terminologie ;
- à l'interrogation, on recherche par le biais d'une description plus ou moins conforme, la forme normalisée utilisée pour désigner un objet.

0.2 LA NOTION DE FIABILITE DANS LES SYSTEMES D'INFORMATION

0.2.1 Introduction

Nous avons évoqué au I.1 la notion d'information floue. Le terme a été employé pour exprimer les faits suivants :

- Le monde réel est un ensemble d'objets que l'on identifie à leurs descriptions standard.
- Les informations saisies ont une forme plus ou moins "correcte" vis à vis de cette réalité.
- Une question peut être une image incomplète ou différente (peut-être faussée) de cette réalité.

Pour préciser les problèmes posés, nous allons définir ce qu'on entend par "fiabilité des informations" dans un système en distinguant le cas des bases factuelles (qui contiennent des informations factuelles) et celui des bases de références (qui contiennent des données textuelles).

0.2.2 Fiabilité dans une base factuelle

0.2.2.1 Définition d'une base factuelle

Nous appelons base factuelle, une base de données dans laquelle les faits sont utilisés presque exclusivement pour des besoins de recherche d'information,

par opposition à une base de connaissances qui contient des faits pouvant être utilisés pour d'autres usages [MYL] (ex. formules logiques, réseaux sémantiques...).

En pratique, ces bases sont en général limitées à des données structurées et formattées [SALT1] [McL] ; elles contiennent souvent relativement peu de "types" de données différents mais beaucoup de valeurs pour chacun (contrairement à une base de connaissances).

Dans tous les cas, le désir de décrire la réalité de manière indépendante de tout traitement informatique conduit à la définition d'un modèle de données. Celui-ci représente la connaissance que l'on a d'une certaine réalité. Il est composé d'un outil qui définit la façon de décrire cette réalité (structure du modèle et ses possibilités) et d'un ensemble de données représentant les valeurs exprimant les faits élémentaires.

On peut distinguer une hiérarchie des faits [DEL] :

- Des faits associés à des données (relations entre des objets) ;
- Des règles sémantiques ou règles de cohérence exprimant des contraintes sur les données ou certaines liaisons entre celles-ci ;
- Des règles de déduction ou de calcul.

Parmi les modèles classiques existants, nous choisissons le modèle relationnel comme support de notre étude dans la suite de ce paragraphe.

0.2.2.2 Modélisation des faits

Un fait peut être représenté par une relation n-aire. Nous rappelons très brièvement les concepts de base du modèle relationnel, celui-ci proposant probablement la structure la plus simple pour décrire le contenu d'une base factuelle. On trouvera dans [Cod1] [DAT1] une présentation complète de ces concepts.

Soit A = {A1,...,An} un ensemble d'attributs. A chaque attribut Ai, est associé un ensemble de valeurs Vi. Une relation sur A est un ensemble de n-uplets dont chaque élément est pris dans Vi.

Une relation R sur A définit un schéma de relation, noté R(A1,...,An) où R est une variable dont les valeurs constituent la relation.

Ces valeurs des attributs de la relation représentent les propriétés des objets de la base.

Exemple : EMP (NOM,SAL)

Ce shéma de relation décrit pour chaque employé son nom et son salaire ; le couple (dupont,4500) appartient à la relation EMP.

0.2.2.3 Les questions

A partir des faits enregistrés dans la base, il est possible d'extraire des données ou de déduire de nouveaux faits.

Dans le modèle relationnel, toute question exprime en elle-même une relation, c'est-à-dire un fait. Cette relation peut soit correspondre au modèle défini pour la base, auquel cas son évaluation est directe, soit elle doit être reformulée en fonction des relations existantes.

0.2.2.4 La fiabilité des informations

La notion de fiabilité peut être considérée sous plusieurs aspects. Nous donnerons la définition suivante : les données de la base sont fiables si elles reflètent fidèlement la réalité. Cela implique à l'interrogation, que tout fait doit être retrouvé s'il a été correctement enregistré.

Si l'on suppose que le modèle d'organisation des données est une représentation correcte de la réalité, c'est-à-dire qu'il permet effectivement d'accéder aux faits soit directement, soit par déduction, ceci revient à poser le problème de la cohérence et de l'intégrité des données de la base.

a. Cohérence des données

Les données de la base sont cohérentes si un même fait du monde réel est représenté une seule fois dans la base (élimination de toute redondance) ou si le système est "au courant" de la redondance et en assume la responsabilité (redondance controlée). Si ce n'est pas le cas, il risque en effet d'y avoir à un moment donné, contradiction entre plusieurs représentations d'un même fait sans que rien ne permette de déterminer laquelle correspond effectivement à la réalité : les informations fournies par une base incohérente peuvent être alors incorrectesou conflictuelles.

La normalisation du système de relation n-aires est un bon moyen pour assurer cette cohérence. L'objectif général du processus de normalisation est de réduire la redondance. Nous ne détaillons pas ces principes qui sont décrits en particulier dans [DAT1].

Le contrôle de la cohérence est limité par le fait qu'on ne peut pas toujours représenter les multiples interdépendances qui existent entre les faits ; seule une partie de la sémantique du monde réel est décrite dans le modèle, ceci délimitant le niveau de représentation de la réalité [BAR]. Cela a pour conséquence que l'état des données peut devenir incohérent par rapport à la réalité stricte, à la suite d'opérations de mise à jour.

b. Intégrité de la base

L'intégrité est une notion plus générale que la cohérence [DAT1]. Le problème de l'intégrité est d'assurer que les données de la base sont valides à tout moment : même si la redondance est éliminée, la base peut cependant contenir des données incorrectes. Il y a bien sûr une limite à cet objectif car il n'y a aucun moyen de vérifier que chaque valeur individuelle entrée dans la base est effectivement correcte. Seule sa plausibilité est vérifiée.

Nous ne parlerons pas ici des problèmes de partage de l'information, sécurité de fonctionnement, confidentialité ... car ce n'est pas l'objet de notre travail, mais particulièrement de l'intégrité relative à la nature et au type des informations (intégrité interne) [HAL] [ABR2] [STO].

A chaque relation est associé un ensemble de contraintes d'intégrité : contraintes sur les attributs, contraintes référentielles [DAT2] ou autres contraintes.

Exemples : - Le diamètre d'une vis est toujours inférieur à 80mm
- si le diamètre d'une vis à bois à tête carrée est égale à 5mm alors la longueur est comprise entre 25mm et 50mm.

A chaque fois qu'une opération d'insertion, de modification ou de suppression est effectuée, on vérifie que les contraintes imposées sont toujours satisfaites. Cette vérification peut être statique ou dynamique ; en effet lors de certaines transactions, des contraintes peuvent être temporairement violées.

En conclusion, la définition de la fiabilité des informations dans une base factuelle repose sur les notions de cohérence, d'intégrité (contraintes d'intégrité) et de plausibilité.

La réalité étant correctement décrite par le modèle, un fait doit être retrouvé s'il appartient à cette réalité.

0.2.3 Fiabilité dans une base de références

0.2.3.1 Définition d'une base de références

Dans une base de données de références, on n'enregistre pas des faits, mais des références à des textes décrivant des faits. L'utilisateur n'est pas en définitive intéressé par les textes (documents) eux-mêmes, mais par les informations qu'ils sont succeptibles de contenir (le texte est un véhicule de la connaissance).

On accède donc aux faits de façon indirecte par l'intermédiaire d'un ou plusieurs documents. Inversement, un document est un rapport sur un ensemble de faits particuliers.

0.2.3.2 Représentation d'un document

En pratique, chaque document est identifié par un ensemble de termes, appelés mots clés ou descripteurs. L'information originelle est transformée afin d'en disposer sous une forme moins coûteuse et plus efficace pour la recherche. Un ensemble de mots clés est sélectionné comme le meilleur reflet du contenu d'un document (indexation), il ne s'agit donc pas de conserver la totalité de l'information liée au document original mais seulement cette partie de l'information qui permettra de le retrouver le plus aisément possible. Cet ensemble de mots clés, appelé descriptif, n'est pas la réalité mais simplement une représentation partielle donc imparfaite de celle-ci.

Deux types d'organisation sont le plus souvent utilisés en pratique [SALT1] :

- Une organisation directe. La recherche se déroule séquentiellement en comparant la question avec chaque document. Son coût étant substantiel, ce type d'organisation ne peut être envisagé que dans des circonstances particulières où la taille des fichiers est réduite.
- Une organisation inverse. C'est la plus courante. Chaque mot clé pointe sur un ensemble de documents contenant ce mot-clé.

0.2.3.3 Les questions

La représentation d'une question est identique à celle d'un document : c'est un ensemble de termes constituant un descriptif. Une réponse correspond à l'ensemble des documents dont le descriptif inclut celui de la question.

Cette notion d'inclusion peut être définie de plusieurs façons :

- Si la formulation d'une question s'obtient en reliant entre eux des termes au moyen d'opérateurs booléens, les documents qui répondent à la question sont ceux dont les descriptifs satisfont l'expression booléenne ainsi formée. Le principal avantage de cette méthode est la rapidité de la recherche ; son principal inconvénient est qu'il est alors impossible d'apprécier les différences de pertinence entre les documents répondant à la question.

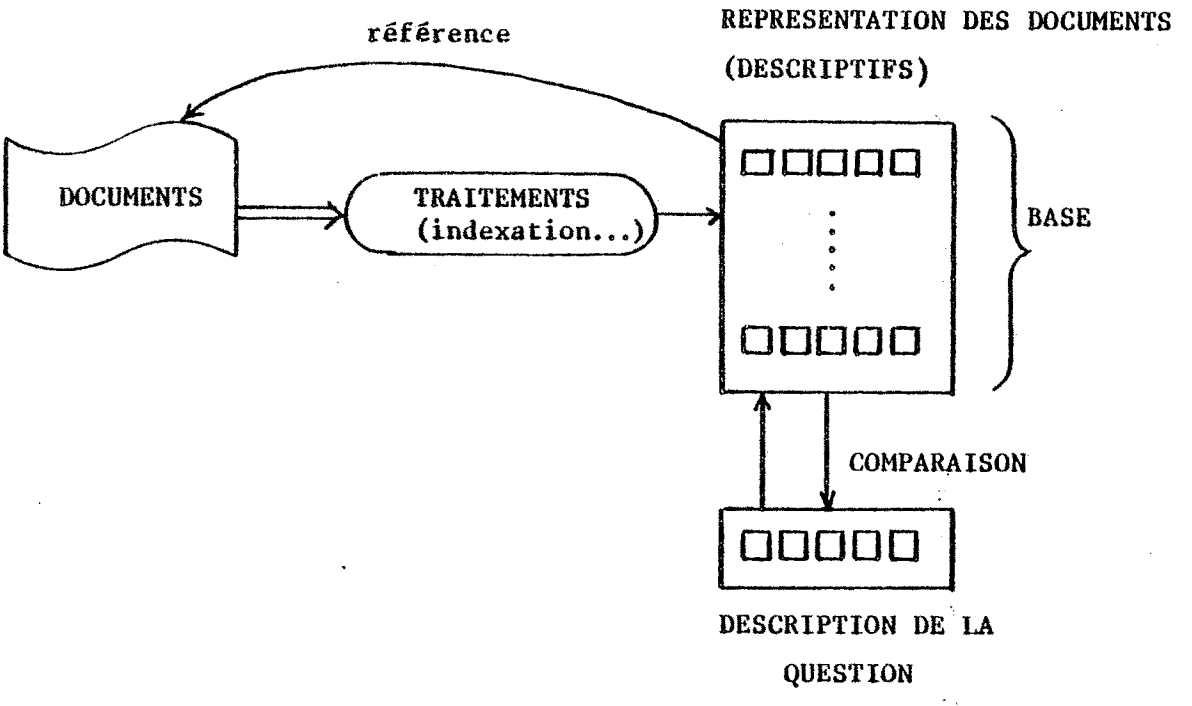
- D'autres méthodes ont eu pour objectif d'éviter les principaux inconvénients de la recherche booléenne ; à titre d'exemple, le calcul d'un degré de similarité entre le descriptif représentant la question et le descriptif représentant chaque document obtenu, en prenant éventuellement en compte le fait que tous les termes n'ont pas la même importance, permet d'affiner les réponses. Les documents sélectionnés sont ceux dont la similarité est jugée satisfaisante ; ils peuvent donc être présentés à l'utilisateur par ordre décroissant de pertinence (projet SMART [SALT2]).

0.2.3.4 La fiabilité des informations

Ici, il n'est pas question de rechercher des faits précis et on parle de la pertinence des documents retrouvés. Un document est pertinent s'il répond au besoin d'information de l'utilisateur.

Il est évident qu'un "bon" système serait celui qui pour toute question, retrouverait le maximum de documents pertinents et fournirait en même temps le minimum de documents non pertinents.

La cause fondamentale de cette imprécision vient de ce qu'une description (d'un document ou d'une question) n'est qu'une image (forcément imparfaite) de la réalité. Il y a un niveau de représentation supplémentaire par rapport aux bases factuelles.



La représentation obtenue est par définition incomplète puisqu'il y a perte d'information à la fois lors de l'élaboration du descriptif à la saisie du texte, et lors de la formulation d'une question car l'utilisateur n'a que des moyens limités pour le faire.

L'ambiguïté dans les langages de description est également une cause de cette imprécision : à un même document peuvent correspondre plusieurs descriptions ;

- C'est un des problèmes essentiels de l'indexation manuelle, car pour un même document, des indexations réalisées par des opérateurs différents peuvent produire des résultats très dissemblables. Cette incohérence affecte nécessairement les performances de la recherche.

- Toute description incluse dans le descriptif d'un document est elle-même une description du document. Cette propriété est fréquemment utilisée dans la recherche.

La notion de fiabilité peut être définie en évaluant le bruit (documents retrouvés non pertinents) et le silence (documents pertinents non retrouvés) dans les réponses. Les mesures les plus courantes permettant d'apprécier les réponses sont le rappel et la précision [SALT1] :

- Le taux de rappel représente la proportion de documents pertinents retrouvés par rapport au nombre total de documents pertinents.

- Le taux de précision représente la proportion de documents pertinents retrouvés par rapport au nombre total de documents retrouvés.

Il est difficile d'avoir simultanément des taux de rappel et de précision satisfaisants. Selon la situation, l'un des deux sera favorisé. Un taux de rappel élevé sera choisi si les utilisateurs désirent retrouver le plus grand nombre de documents probablement intéressants. D'autres utilisateurs peuvent préférer un taux de précision élevé, c'est-à-dire que tous les documents probablement sans intérêt seront rejetés.

Un des moyens pour augmenter la qualité des réponses est l'utilisation d'un thésaurus. Celui-ci peut être défini de la façon suivante : il est composé d'un ensemble de termes qui tiennent lieu de concepts (descripteurs) et d'un ensemble de relations entre ces termes. Il est possible de réduire le silence, par exemple, en reformulant automatiquement une question en tenant compte des informations contenues dans le thésaurus [SALT1].

En conclusion, la fiabilité dans une base de références peut être évaluée en fonction de la pertinence des documents retrouvés à l'issue d'une recherche. La qualité de l'indexation est alors essentielle puisqu'à la base des performances futures du processus de recherche.

0.2.4 Cas d'une base de nomenclatures

0.2.4.1 Caractéristiques de la base

Une base de nomenclatures est une base factuelle. Elle décrit une collection finie de composants. Chaque composant est défini par un ensemble de caractéristiques telles que le numéro (ou référence interne), le nom du fabricant, le prix s'il est approvisionnable, la référence-article...

Les références-articles des composants sont repertoriées dans les catalogues des fabricants, soit explicitement (par énumération) soit implicitement (en donnant les règles de construction).

La référence-article du fabricant joue à la fois un rôle d'identificateur d'un objet et de description. En tant qu'identificateur, c'est une caractéristique discriminante relativement à l'ensemble des objets (plusieurs objets ne peuvent avoir des valeurs identiques pour cette caractéristique). En tant que description, c'est une chaîne d'un certain langage. Ce double aspect caractérise ce type de base.

D'autre part, les descriptions sont courtes et concises, ceci permettant d'envisager un modèle simple de représentation de la connaissance.

0.2.4.2 Problèmes liés aux données

La saisie peut donner lieu à un grand nombre de variations ou d'erreurs dans les textes des références-articles.Les causes proviennent à la fois :

- De la grande hétérogénéîté des expressions des références-articles et de leur complexité. Il est impossible de demander à l'utilisateur une maîtrise totale du langage qui doit être employé dans chaque cas et la consultation des catalogues est une opération fastidieuse qui de toute façon, ne supprime pas toute erreur.
- D'un manque de formalisation de ces langages.

La référence-article du fabricant constitue la description originelle d'un objet. Lors de la saisie, cette description peut être plus ou moins altérée. Nous appellerons variation une forme prévisible et acceptable d'altération. Toutes les descriptions obtenues par variation à partir de la description originelle, donnent une image permettant effectivement la reconnaissance sans ambiguîté d'un objet : ce sont des descriptions synonymes.

Exemple :
référence-article : VIS CHC M3×10 U ACIER QUALITE 10.9 CD8B
description synonyme : VIS CHC M3×10 U ACIER QUAL.10.9 CADMIE
("CD8B" et "CADMIE" sont des constituants synonymes).

Cette description est effectivement acceptable car elle n'engendre pas d'ambiguîté.

Le terme d'erreur regroupe toutes les autres formes d'altération. Ces erreurs sont décelables puisqu'il est toujours possible de se référer aux descriptions originelles. Elles sont rectifiables si l'objet induit de la description originelle est unique, c'est-à-dire si elles n'engendrent pas d'ambiguîté.

0.2.4.3 Problèmes liés à la recherche

Le problème auquel nous nous intéressons est la recherche d'un objet à partir d'une description. Une question, relativement à la base, est donc une description que l'on veut identifier ou rapprocher d'un objet de la base.

Il existe deux types de questions possibles :

- La personne qui effectue une recherche, a une idée sur la façon dont un objet est identifié dans la base. Dans le cas où il n'y a pas strictement identité entre la question et un élement de l'ensemble des descriptions synonymes d'un objet, la question est une image approximative d'un objet. Cette imprécision est due à des erreurs (un cas particulier d'erreur étant l'omission), celles-ci n'étant pas toujours décelables a priori.

- L'utilisateur n'a aucune idée sur la forme de la description originelle de l'objet qu'il recherche mais il possède des informations sur cet objet. Il connait, par exemple, certaines de ses caractéristiques, mais il ne connait pas les règles de formulation de la description (cas d'une référence codée).
  Dans ce cas, sa question sera une description "libre" de cet objet, le terme "libre" signifiant libéré de toutes contraintes formelles.

La seule base de recherche possible est alors une référence au contenu des descriptions.

0.2.2.4 Fiabilité des informations

- Lors de la saisie, l'introduction dans la base de descriptions erronées ou de plusieurs descriptions synonymes a pour conséquence des incohérences et des ambiguïtés (représentations multiples d'un même objet, représentations identiques d'objets différents) qui se traduisent par l'impossibilité ultérieure de retrouver ces objets dans la base.

- L'utilisateur qui interroge la base, recherche un objet précis mais il en possède une description dont il ne connait pas la qualité (premier type de question). Cet objet doit être retrouvé s'il existe dans la base.

Compte-tenu des propriétés des informations traitées, une analyse préalable des descriptions est une phrase importante pour :

- Déceler les erreurs pouvant être commises lors de la saisie (contrôle)

- Normaliser les descriptions (reconnaissance des variations)

- Lors de la recherche, pouvoir établir un rapprochement aussi peu ambigü que possible entre une description quelconque et un objet (ou plusieurs dans le cas où celle-ci est insuffisante).

0.3 CONCLUSION

Nous avons vu que dans un système de type factuel, l'information fournie doit être soit explicitement contenue dans la base, soit déduite des informations enregistrées dans la base. Dans toute sa généralité, un tel système doit être capable de générer des réponses à une très grande variété de questions.

Une solution de type factuel ne serait envisageable que dans la mesure où il serait possible d'analyser sémantiquement toute description et de la rapprocher d'un objet. La complexité des opérations nécessaires pour décoder les questions et pour manipuler les informations enregistrées, nous a conduit à conclure que le modèle factuel n'est pas directement adapté aux données traitées.

La méthode que nous proposons, consiste à considérer le texte d'une description, non plus globalement comme une entité d'information mais comme un texte véhiculant un ensemble d'informations. Une analyse partielle du contenu, s'appuyant sur une théorie linguistique (caractérisation des unités "lexicales", règles grammaticales et sémantiques) conduit à identifier le texte associé à un objet (référence) à un ensemble de termes (ou mots-clés) reflétant son contenu. Toute question est considérée comme une référence potentielle à un objet et est identifiée d'une manière analogue.

Cette approche de type documentaire permet de rendre compte des relations existant entre un objet et une description, celles-ci étant à la base du processus de rapprochement.

Ce niveau de représentation de la connaissance est suffisant car les textes des descriptions sont courts et leur interprétation est permise par une association entre forme et signifiant (passage entre niveau sémantique et descriptif).
On s'oriente donc vers une base qui présente les deux caractéristiques : factuelle dans sa définition et documentaire dans les méthodes employées.

Dans le chapitre I de cette thèse, sont développés les aspects linguistiques de notre étude. La grammaire définie est dans son principe à rapprocher des grammaires sémantiques utilisées dans les systèmes spécialisés possédant un

interface en langue naturelle (PLANES [WAG], BAOBAB [BON], LIFER [HEN], [BUR]).

Le chapitre II est consacré à la définition des méthodes et des objectifs fixés pour la réalisation et à la présentation de cette dernière. Dans le chapitre III sont proposées des mesures dans le système en vue d'améliorer les performances globales de celui-ci. Enfin, nous concluons au chapitre IV par une évaluation de la réalisation et les perspectives offertes par cette étude.

CHAPITRE I

ASPECTS LINGUISTIQUES

I.1 CARACTERISATION DES DONNEES ET EXEMPLES

I.1.1. Les objets

Une nomenclature décrit l'ensemble des objets intervenant dans la fabrication d'un produit. Ces objets sont les composants élémentaires du produit. Chacun d'entre eux appartient à une certaine famille technologique (électrique, mécanique...) et possède une dénomination (vis, condensateur...), une désignation (ou référence-article) et une origine (fabricant identifié par sa raison sociale). La référence-article est suffisante pour identifier un objet.

I.1.2. Les références-articles

Une référence-article se présente sous la forme d'un texte de longueur quelconque mais limitée (en général inférieure à 50 caractères). Ce texte est utilisé à la fois pour identifier l'objet et pour le caractériser, c'est-à-dire que certaines de ses propriétés peuvent y être décrites. Dans ce sens, nous dirons qu'une référence-article constitue une description d'un objet bien que dans certains cas, elle n'ait qu'une référence à une description.

La référence-article d'un objet est obtenue en appliquant des règles plus ou moins complexes. Celles-ci sont élaborées par le fabricant de l'objet, quelquefois en relation avec une norme.

I.1.3. Exemples

Traditionnellement, les références-article sont classées en deux familles : descriptive et codée. Sans approfondir dès maintenant cette distinction, nous allons illustrer en reprenant les exemples cités dans le ch. 0 la façon dont un objet peut être décrit :

Exemple 1. Origine : AFNOR (famille descriptive), dénomination : VIS [AFN]

VIS	CHC	M10×80/30	U	ACIER	QUALITE 8·8	CD8B
1	2	3	4	5	6	7

Cette référence-article est constituée de 7 champs.

1 dénomination du composant
2 forme de la tête (tête cylindrique, à six pans creux)
3 dimensions en millimètre (diamètre 10, longueur de la tige 80, longueur filetée 30)
4 état de finition (fine)
5 matière
6 classe de qualité (8×8 est sensiblement la limite minimale d'élasticité en hectobars)
7 traitement

Exemple 2. Origine : RAYCHEM (famille codée), dénomination : CABLE

44/2124-22-2/6-9

1 23456 7 8 9 10

1 numéro de spécification
2 température de fonctionnement (135°C)
3 type du fil (blindé, tresse plate et isolée)
4 classe du fil (600V, usage général)
5 nombre de conducteurs (2)
6 nature de l'âme (alliage de cuivre à haute résistance, argenté)
7 dimension des conducteurs (AWG 22 → section du conducteur 0,38 MM^2)
8,9 couleurs des gaines isolantes des conducteurs (1er fil rouge, 2ème fil bleu)
10 couleur de l'isolant extérieur (blanc)

- Détail de la codification du champ 6 (selon le catalogue du fabricant)

nature de l'âme	1	cuivre étamé
	2	cuivre argenté
	3	cuivre nickelé
	4	alliage de cuivre à haute résistance, argenté
	5	aluminium
	6	alliage de cuivre à haute résistance, nickelé.

- Détail de la codification des champs 8, 9, 10.

couleur des isolants	0 → noir	5 → vert
	1 → marron	6 → bleu
	2 → rouge	7 → violet
	3 → orange	8 → gris
	4 → jaune	9 → blanc

Le code est qualifié de décomposable car les valeurs de chaque champ (à part le premier) sont interprétables.

Exemple 3. Origine : JEDEC, dénomination : DIODE

IN5665A

Ce code est qualifié d'indécomposable car il joue uniquement un rôle d'identificateur.

Exemple 4. Origine : MCB, dénomination : RESISTANCE

BOBI 12 47 OHMS ±10%

1 2 3

1 type de la résistance (identificateur d'une sous-famille de résistances)
2 résistance (valeur ohmique)
3 tolérance

"BOBI 12" a le même rôle que "44" dans l'exemple 2 : celui d'identificateur.

I.1.4. Le langage des descriptions

Le langage des descriptions est défini par l'ensemble de tous les textes de description ou d'identification des objets "acceptables" à la saisie. Il contient le langage des références-article fabricant (ensemble des descriptions standards).

Ce langage n'est pas figé (fabrication de nouveaux objets, abandon de fabrication d'un objet, choix d'une nouvelle description pour un objet...).

Définition d'une description "acceptable"

Une description est "acceptable" si elle appartient à l'ensemble des descriptions synonymes d'un objet.

Deux descriptions sont dites synonymes si elles contiennent exactement les mêmes informations relativement aux caractéristiques de l'objet. L'accès à une information peut être soit :

- direct : l'information est contenue dans le texte de la description lui même. C'est une valeur de caractéristique.
- indirect : le texte contient une référence à une information stockée ailleurs. Nous avons désigné celle-ci par le terme "identificateur".

Les possibilités de synonymie sont les suivantes :

1er cas. Objet ayant plusieurs origines possibles.

Dans ce cas, il existe au moins autant de descriptions synonymes que d'origines

Exemple : EPDF 6 BL.P (origine : FILOTEX)
et KY44A-01 (origine : UTE/CCT)

sont deux descriptions synonymes car elles correspondent au même objet physique.

2ème cas. Descriptions synonymes pour un objet ayant une origine donnée.

Deux descriptions sont synonymes si :

- les champs "identificateur" sont identiques. En effet un identificateur est par définition unique.
- les champs "valeur de caractéristique" sont synonymes ou identiques. Deux valeurs sont synonymes s'il est possible de les interchanger sans modifier l'information apportée. Cette synonymie peut être indépéndante ou non du contexte.

Exemples : . synonymie indépendante du contexte

VIS H SI 8×30/25 E24-2 NON PROTEGE

VIS H SI 8×30/25 NON PRO. ADX

Ces deux descriptions sont synonymes car

ADX et E24-2 (nuance d'acier) sont des valeurs synonymes et une abrévia-tion non ambigüe a été utilisée.

(l'ordre des champs n'est pas significatif)

BOBI 12 350 OHMS ± 10%

BOBI 12 0,35 KOHMS ± 10PCT

"BOBI 12" qui est un identificateur n'a pas de synonyme.

. synonymie dépendante du contexte

44/2113-30-0

44/2113-30-NOIR

44/2113-30 COULEUR DE L'ISOLANT NOIR

En toute logique, ces deux descriptions sont équivalentes. Mais en général, il n'existe pas de réel synonyme pour une valeur de caractéristique codée. Par exemple, la valeur "3" du champ 6 représente une information plus précise que simplement "cuivre nickelé" qui est assez vague.

Avant d'aborder la caractérisation des différents aspects du langage (I.3), cette étude conduisant au choix d'un modèle linguistique (I.4), nous définissons la notion de mot en examinant les méthodes pour y parvenir.

I.2 LA NOTION DE MOT

Le texte d'une description se présente au niveau le plus élémentaire comme une séquence de caractères comprenant des lettres, des chiffres et des symboles spéciaux.

Il convient de définir dans cette séquence les chaînes de caractères qui forment les unités linguistiques de plus haut niveau, c'est-à-dire les mots.

Pour ce faire, plusieurs approches sont possibles. La méthode d'analyse distributionnelle, qui est une méthode purement formelle, permet d'aborder la description d'une langue qu'on ne connaît pas (ce que Chomsky appelle une "procédure de découverte"). L'alternative consiste à étudier une langue connue, en se fiant à son intuition. La première démarche d'apparence modeste et réaliste est en réalité ambitieuse et complexe. La méthode est décrite en détail dans de nombreux ouvrages (notamment [HAR]).

I.2.1. L'approche distributionnelle

La première chose à faire est de réunir un corpus, c'est-à-dire un ensemble d'énoncés qui sera envisagé comme un échantillon de la langue.

On étudie l'équivalence distributionnelle, en regardant l'effet d'une substitution d'une unité par une autre dans un certain contexte, les unités substituées

pouvant être soit de simples caractères, soit des groupes de caractères.

I.2.1.1. La notion de distribution (définitions)

Chaque unité linguistique (à l'exception de la phrase) a une distribution caractéristique. Si deux unités apparaissent dans le même ensemble de contextes, on dit qu'elles ont la même distribution (ou qu'elles sont équivalentes). Si elles n'ont aucun contexte en commun, leurs distributions sont complémentaires. Entre ces deux extrêmes, il existe deux sortes d'équivalence partielle : distribution incluse (x apparaît dans tous les contextes où y apparaît mais le contraire est faux), distributions se chevauchant (équivalence dans certains contextes).

L'ensemble des contextes dans lesquels une unité linguistique peut apparaître est sa distribution.

I.2.1.2. La méthode

Le corpus une fois recueilli, on le segmente. Pour ce faire, on cherche à rapprocher des 'blocs' comparables de façon à déterminer de proche en proche les unités linguistiques.

Dans une phrase, une unité à laquelle on peut substituer d'autres'blocs' (d'un ou plusieurs caractères) pour produire des phrases acceptables est reconnue comme étant un mot du langage.

Cette méthode d'analyse distributionnelle constitue une approché formelle à l'analyse grammaticale.

I.2.1.3. Classes grammaticales

L'ensemble des unités ayant une distribution partiellement équivalente (l'ensemble des contextes étant à définir) détermine une classe d'équivalence pour la relation de substitution : cette classe est alors la classe grammaticale.

D'un point de vue purement formel, le nom choisi pour la classe est sans importance car il n'y a aucune référence à l'interprétation des mots.

I.2.1.4. Limites du distributionalisme

Quelque soit la complexité de certaines de ses procédures, la linguistique distributionnelle repose fondamentalement sur quelques idées simples et il est difficile de ne pas se comporter en distributionaliste face à une langue inconnue. Mais ses limites apparaissent vite ; il suffit d'essayer de découper une phrase en mots pour s'apercevoir que ce n'est pas chose facile [TAR].

D'autre part, l'ensemble des contextes d'un mot étant en général très restreint, les classes de substitution définies par cette méthode sont très nombreuses.

L'analyse distributionnelle est une source d'information parmi d'autres pour la description d'un langage, mais son indépendance vis à vis de notre propre connaissance du langage est limitée.

Une autre approche consiste à prendre en compte des considérations sémantiques. Un mot sera alors défini comme étant la plus petite unité grammaticale et porteuse de sens [LYO].

I.2.2 L'approche sémantique

I.2.2.1. Catégories sémantiques

La notion de catégorie sémantique est liée à l'interprétation des mots. L'idée est de regrouper les mots qui ont un trait commun du point de vue sémantique qui les distingue des autres (c'est-à-dire que les autres ne partagent pas). Ce trait commun est désigné dans la littérature par les termes de marqueur sémantique [KAF], sémème ou catégorie sémantique.

Cette classification ne peut être faite qu'à partir d'une connaissance encyclopédique du langage puisqu'elle est fondée sur une appréciation intuitive du sens des mots.

Exemples : 1 "ACIER", "CUIVRE", "ALUMINIUM"... peuvent être rattachés à la catégorie MATIERE

2 "20" dans 44/1132-20-0/2/9-9 et "DIA.30" peuvent être rattachés à la catégorie DIMENSION

3 "QUALITE 10.9" et "23CND18-12" sont des nuances d'alliage.

Le signifiant d'un mot peut être l'union de plusieurs sémèmes et ceux-ci peuvent être apportés par des affixes ou des suffixes. Dans les exemples 2 et 3 les sous-chaînes "DIA." et "QUALITE " peuvent être considérées comme des racines, les parties numériques étant des suffixes amenant une précision sur le sens.

I.2.2.2. Conclusion

Les termes d'indexation d'une description sont les mots ou groupes de mots (normalisés) déterminés par la segmentation. Nous avons vu que le langage que nous avions à décrire contient des chaînes interprétables et des chaînes non interprétables (identificateurs et certains codes dits indécomposables). Dans le premier cas, nous nous sommes servis du support sémantique pour définir et classifier les mots. Ceci nous a conduit à choisir des catégories telles que : DENOMINATION, MATIERE, TRAITEMENT, FORME, TOLERANCE, INTENSITE, COULEUR, NOMBRE DE CONDUCTEURS ...

Dans le second cas, puisqu'il n'est plus possible de prendre en compte des considérations de sens, l'approche ne peut être que formelle (cf I.2.1.2). Dans la mesure où ce type de chaînes est relativement fréquent dans le langage, la méthode distributionaliste pourrait être utilisée pour les segmenter. Cependant, dans un premier temps pour des raisons de simplicité, nous avons été conduit à choisir une méthode différente : la segmentation s'opère en tenant compte uniquement du type des caractères composant une chaîne et de la présence de séparateurs (cf. II.2.3.2). Au vue des données traitées, il nous semble que ces deux approches, bien qu'a priori très différentes, doivent fournir en fait des résultats assez voisins car le changement de type de caractères et les séparateurs sont effectivement souvent utilisés pour délimiter des sous-chaînes qui forment un tout, mais celui-ci n'est pas interprétable. Il existe une structure dans ces codes qui n'est probablement pas descriptive et donc des règles, reflétant un contenu sémantique mais qu'on ignore.

I.2.3. Interprétation d'un mot

Lorsqu'un mot possède une interprétation, celle-ci est exprimée par une référence à une catégorie sémantique et une liste de termes empruntés au langage naturel

ou à un langage technique. Ces termes sont des valeurs de caractéristiques de l'objet. La combinaison de ces deux types d'information représente le concept associé au mot.

Si le mot n'est pas une valeur de code, la liste est constituée d'un seul élément qui est le mot lui-même. Sinon, la liste représente une "traduction" de la valeur dans le contexte considéré.

Exemples : "acier" → [MATIERE] acier

dans 44/2124-22-2/6-9

(1) (2)

(1) "2" → [TYPE DU FIL] (blindé tresse plate isolée)

(2) "2" → [COULEUR] rouge

Pour les mots non interprétables (identificateurs ou éléments d'identificateurs), seule une étiquette de nature formelle peut être assignée.

Synonymes

Deux mots sont synonymes s'ils représentent la même valeur de caractéristique (voir exemples I.1.4).

Deux cas particuliers et importants de synonymie sont :

- les abréviations non ambigües et courantes d'un mot (LONGUEUR 30 MM, LONG. 30 MM, LG. 30 MM...)
- les produits de différents choix d'unités (3000 OHMS, 3 KOHMS)

I.3 ASPECTS DU LANGAGE

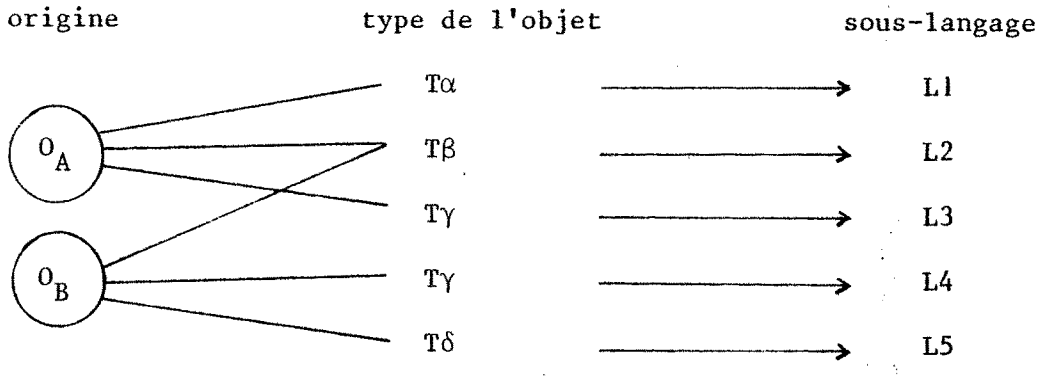
I.3.1. Introduction

Nous avons jusqu'à présent employé le terme de "langage des descriptions" comme étant globalement défini par l'ensemble de toutes les descriptions acceptables des divers objets.

En fait chaque objet est décrit en respectant des règles relatives à son origine et à son type. Ces deux données caractérisent un ensemble cohérent de descriptions ; chacun de ces ensembles définit un sous-langage.

L'origine et le type de l'objet étant connus à la saisie, ils peuvent être considérés comme les préfixes des descriptions. Ces préfixes sont discriminants pour caractériser un sous-langage.

Ceci peut être schématiquement représenté sur un exemple :



Le langage des descriptions est la réunion de tous ces sous-langages.

I.3.2. Propriétés des sous-langages

Soit Li l'ensemble des phrases préfixées par une origine et un type d'objet donnés et Wi l'ensemble des mots de Li ($Li \subseteq Wi^*$)

- En général, pour deux langages Li et Lj, $Wi \cap Wj \neq \emptyset$
- Le préfixe permet d'affirmer que tous les langages sont distincts

  $\forall \alpha \in \bigcup_{i=1}^{N} Li \quad \exists! k$ tel que $\alpha \in Lk$

  $\forall i, \forall j \quad Li \cap Lj = \emptyset$

- Dans certains langages, l'ordre de certains mots dans les phrases n'est pas significatif. Ils peuvent être permutés sans que soit modifiée l'interprétation de la phrase. Ce sont des paraphrases d'une phrase originale.
- Toute description a une longueur finie. L'ensemble des descriptions synonymes d'un objet (obtenues par paraphrases et (ou) substitution de mots synonymes) est fini ainsi que l'ensemble des objets existants.

Donc $\forall i$ Li est un langage fini

et $\bigcup_{i=1}^{N} Li$ aussi

I.3.3. Syntaxe et sémantique

Une description est une phrase du langage. Une phrase est une combinaison d'unités fondamentales qui sont les mots. Ces mots possèdent ou non une interprétation dans un contexte donné.

La syntaxe traite de la manière dont sont combinés ces mots pour former une phrase. Elle définit :

- un ordre sur les mots ;
- la définition des différents mots ou champs-caractéristiques ;
- les types des valeurs prises par ces champs (vocabulaire).

Le but final d'une description étant de nommer un objet, l'interprétation d'une phrase est l'objet physique qu'elle réfère.

Une combinaison de mots a donc un sens si elle désigne effectivement un objet qui existe physiquement, c'est-à-dire qui est fabriqué (cette existance est bien-sûr variable dans le temps).

Certaines descriptions peuvent être syntaxiquement bien formées mais dépourvues de signification car sans correspondance avec aucun objet réel parce que celui-ci n'a jamais été fabriqué (il n'y a jamais eu de demandes pour cet objet, celui-ci ne peut pas exister ou être utilisé avec de telles caractéristiques...).

On se démarque donc ici nettement de la linguistique où l'objet de l'étude est une infinité de réalisations possibles, y compris celles qui n'ont jamais été prononcées ou étendues (l'objet n'est plus empiriquement observable). Le terme de syntaxe dans notre domaine correspond à la notion de pragmatique en linguistique universelle car le niveau syntaxique décrit l'ensemble des objets susceptibles d'être représentés.

D'autre part, le niveau syntaxique est beaucoup plus riche que dans une langue naturelle car il contient potentiellement une grande partie de la sémantique du langage.

En dernier ressort, vérifier si une description est sémantiquement correcte revient à compulser une liste dans un catalogue d'articles.

Dans certains cas, des règles sémantiques sont "prévues" et énoncées par le

fabricant de façon explicite. Celles-ci sont toujours très spécifiques comme nous allons le voir sur un exemple :

1 44/2412-30-9
(1)(2)

2 44A2412-30-9
(1)(2)

(1) température de fonctionnement ("A" signifie 150°C et "/", 135°C).
(2) classe du fil (600V, fil spatial).

Les deux descriptions 1 et 2 sont syntaxiquement correctes mais 2 désigne un objet qui n'existe pas car la classe 4 du fil n'est disponible qu'en construction 135°C ("/").

En pratique donc, on ne peut que définir, pour des raisons de simplicité, des syntaxes (règles de construction des descriptions) qui correspondent à des sur-ensembles des différentes classes d'objets. La grammaire utilisée est à rapprocher des grammaires sémantiques (les symboles non-terminaux dénotent des catégories sémantiques) [HEN] [WAG] [BUR].

I.3.4. Bilan sur les codes

Le langage des descriptions est un langage essentiellement codé dans le but de représenter les informations concernant un objet de façon condensée. Il existe plusieurs types de codes :

- Les références ou codes indécomposables ;
- Les codes décomposables. La codification est soit de nature purement syntaxique, soit syntaxico-sémantique.

Exemples :

Les dimensions en visservie selon la norme AFNOR se représentent de la façon suivante : M d×l [/n]

M est le type du pas de vis ;
d est la valeur du diamètre en mm ;
l est la valeur de la longueur de la tige en mm ;
n est la valeur de la longueur filetée (si x ≠ 1).

C'est une codification syntaxique. Les valeurs sont exprimées "en clair".

La syntaxe étant M DIA × LONG [/ LONG FIL]

Dans l'exemple déjà cité 44/1214-22-9, certaines valeurs sont codées. L'interprétation de ces valeurs est déterminée à la fois par leur position (syntaxe) et par la connaissance des règles de traduction.

"9" → [COULEUR] blanc

I.4 CHOIX DU MODELE GRAMMATICAL

I.4.1 Les critères de choix

Le choix d'une grammaire, c'est-à-dire de l'ensemble des règles décrivant la façon dont sont combinés les mots pour former les phrases du langage, repose sur les critères suivants :

- simplicité. Les règles doivent être simples, c'est-à-dire commodes d'écriture. Elles doivent être l'expression des régularités du langage. Ce point est particulièrement important ici, car ces règles seront définies par des non-informaticiens.
- puissance. Les règles doivent être puissantes. Ce terme est quelque peu difficile à définir, du moins de façon technique. Nous dirons simplement qu'une règle est plus puissante qu'une autre si elle prend en compte plus de "faits" ou les reflète plus "correctement" [LYO]. Outre le critère d'économie, cette définition fait appel en particulier à des considérations sur l'"adéquation" d'une grammaire. Les règles d'une grammaire reflètent une partie de la sémantique du langage. Le rôle d'une grammaire n'est pas seulement de définir l'ensemble des combinaisons de mots grammaticales mais elle fournit aussi pour chaque combinaison correcte une description structurelle qui soustend une interprétation sémantique partielle. La notion d'"adéquation" porte sur la qualité de cette description (on suppose, que dans certains cas du moins, il est possible de dire si une description est plus "correcte" qu'une autre).

- précision. Le langage étudié ici est un corpus attesté, c'est-à-dire fini (contrairement au langage naturel) mais il n'est pas figé : un objet qui n'existe pas à un instant donné, sera peut-être fabriqué plus tard.

Nous rappelons, d'autre part, le premier objectif qui est la simplicité du modèle choisi. Les contraintes de nature sémantique sont reportées au niveau syntaxique mais celles-ci ne sont pas toutes exprimées (cf. I.3.3).

Les critères de simplicité et d'économie nous conduisent à choisir une grammaire qui génère toutes les phrases du langage mais aussi des phrases qui n'ont pas d'interprétation. Elle décrit donc un sur-ensemble du langage. Le terme de précision recouvre le fait que ceci ne doit pas conduire à créer des ambiguïtés (toute phrase du langage possède une interprétation unique).

I.4.2 Structure du langage

Le langage des descriptions (qui, nous le rappelons, est l'union d'un grand nombre de sous-langages) possède une structure linéaire : chaque phrase du langage peut être décrite d'une façon satisfaisante du point de vue grammatical par une séquence de constituants (mots). On trouve cependant la notion de "modificateur" (par exemple dans "ACIER Z3CND18-12" le mot "Z3CN18-12" qui représente une composition, modifie le mot "ACIER"). Mais il n'existe pas de phénomène d'enchâssement (ou plus simplement d'autoimbrication) ceci se traduisant par une non-récursivité de la grammaire.
Il existe dans certains sous-langages des phénomènes particuliers :

- certains mots sont permutables dans une phrase, chaque permutation produisant une paraphrase de la phrase originale.
  Exemple : RS36Y 6,19 KOHMS ± 1PCT
  RS36Y ± 1PCT 6,19 KOHMS
  6,19 KOHMS RS36Y ± 1PCT

- le nombre d'occurrences d'un champ dépend de la valeur d'une autre caractéristique ; ce phénomène est appelé "champ à occurrences multiples".
  Exemple : 44/1142-22-0/2/6/9-9
  1 2 3 4 5

Le cable est composé de 4 conducteurs (1). Chaque conducteur est isolé par une gaine d'une couleur particulière. Les 4 couleurs sont précisées en (2) (3) (4)

(5) : "0" (noir), "2" (rouge), "6" (bleu) et "9" (blanc).

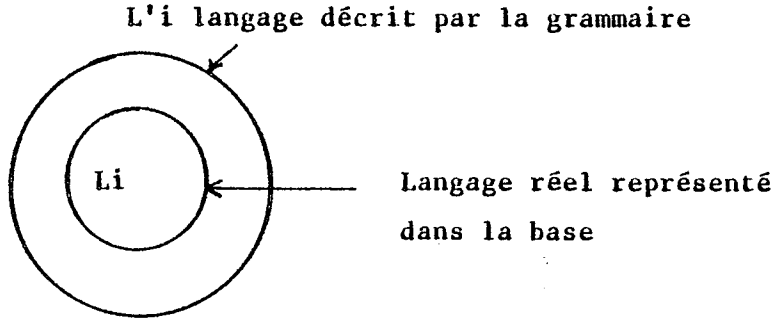
44/1121-20-0/2-9

Le cable est composé de 2 conducteurs ; la gaine isolante du 1er fil est noire, celle du 2ème fil est rouge.

I.4.3 Conclusion

Ces différentes remarques nous ont conduits à nous placer dans le cas des modèles d'états finis. Ce type de modèle répond au critère de simplicité. D'autre part, la corrélation syntaxe-sémantique étant immédiate dans le langage, la vue des relations sémantiques qu'il permet d'exprimer est suffisante.

Lors de la saisie, une validation sémantique totale ne peut être effectuée (sur-langage) ; celle-ci est ramenée lors de la recherche à une comparaison directe avec l'ensemble des descriptions enregistrées dans la base.



I.5 MODELE D'ETATS FINIS

Le modèle que nous avons choisi est donc le modèle d'états finis. Nous rappelons que l'un des objectifs de l'analyse linguistique des descriptions (cf. 0.1.1) est de fournir un moyen d'indexation de celles-ci. L'ensemble des termes d'indexation représentant une description est produit par transduction. Pour réaliser cette transduction, il faut tenir compte du fait que le langage considéré est l'union de plusieurs sous-langages. Nous sommes donc amenés à étudier les caractéristiques du transducteur d'états finis utilisé, sachant que les langages sont disjoints. Nous faisons auparavant quelques rappels sur les automates et transducteurs d'états finis.

1.5.1 Rappels

Les définitions de ce paragraphe, ainsi que les théorèmes et leur démonstration se trouvent dans [AHU].

1.5.1.1 Automate d'états finis

a. Un automate d'états finis déterministe est un quintuplet

(Q,Σ,δ,qo,F)

où Q est un ensemble fini d'états

Σ est un vocabulaire fini

δ est une application de $Q\times\Sigma$ dans Q (fonction de transition)

$qo \in Q$ (état initial)

$F \subset Q$ (ensemble des états finaux).

L'automate déterministe est tel que toute configuration a au plus une configuration successeur.

b. Un automate d'états finis non déterministe est un quintuplet :

(Q,Σ,δ,qo,F)

où Q,Σ,qo,F ont les mêmes définitions que précédemment et

δ est une application de $Q\times\Sigma$ dans les sous-ensembles de Q, c'est-à-dire que certaines configurations ont plusieurs successeurs.

c. Soit M un automate non déterministe acceptant un langage L. Alors il existe un automate déterministe M' qui accepte L, c'est-à-dire tel que L(M')=L(M).

d. Automate acceptant L1 $\cup$ L2

Soient M1 et M2, deux automates d'états finis déterministes acceptant respectivement L1 et L2.

Il existe un automate M déterministe tel que :

$L(M)=L(M1) \cup L(M2)$

I.5.1.2 Transducteur d'états finis

Il existe de nombreuses définitions d'un transducteur dans la littérature. Nous choisirons la plus simple :

$T = (Q, \Sigma e, \Sigma s, \delta, qo, F)$

Q est un ensemble fini d'états

Σe est un vocabulaire fini d'entrée

Σs est un vocabulaire fini de sortie

δ est une application de $Q \times \Sigma e$ dans les sous-ensembles de $Q \times \Sigma_s^*$

$qo \in Q$ (état initial)

$F \subset Q$ (ensemble des états finaux)

Dans cette définition, nous considérons que la base d'un transducteur d'états finis est un automate d'états finis non déterministe, car nous nous intéressons aux transducteurs accepteurs.

Un transducteur d'états finis T est dit déterministe si :

$\forall a \in \Sigma e$, $\delta(q,a)$ contient au plus un élément.

Une configuration de T est un triplet (q,w,W) où $q \in Q$, $w \in \Sigma e_e^*$, $W \in \Sigma_s^*$.
On définit une relation binaire sur les configurations, notée $\vdash_T$:

$\forall q \in Q$, $a \in \Sigma$, $w \in \Sigma_e^*$ et $W \in \Sigma_s^*$ tels que $\delta(q,a)$ contient (r,Z), alors
$(q,aw,W) \vdash_T (r,w,WZ)$
On note $\vdash_T^*$ la fermeture réflexive de $\vdash_T$

La traduction $\tau(T)$ définie par T est :

$\{(w,W) \;/\; w \in \Sigma_e^*,\ W \in \Sigma_s^*,\ (qo,w,\epsilon) \vdash_T^* (q,\epsilon,W),\ q \in F\}$

où ϵ dénote la chaîne vide .

I.5.2 Transducteur acceptant l'union de plusieurs langages

Soient T1 et T2 deux transducteurs d'états finis déterministes acceptant respectivement les langages L1 et L2.

Nous supposons $L1 \cap L2 = \emptyset$

$T1 = (Q1, \Sigma e1, \Sigma s1, \delta 1, qo1, F1)$

$T2 = (Q2, \Sigma e2, \Sigma s2, \delta 2, qo2, F2)$

Nous envisageons le cas général où $\Sigma e1 \cap \Sigma e2 \neq \emptyset$

On définit le transducteur T tel que :

- T accepte $L1 \cup L2$
- $\tau(T) = \tau(T1) \cup \tau(T2)$

$\tau(T)$ peut-elle être définie par un transducteur déterministe ?

a. Proposition 1

Si pour $w \in (\Sigma e1 \cap \Sigma e2)^*$, il existe $w1 \in \Sigma e_1^*$ et $w2 \in \Sigma e_2^*$ tels que $ww1 \in L1$ et $ww2 \in L2$, alors il n'existe pas en général de transducteur d'états finis déterministe T définissant une traduction contenant $(ww1, W1)$, $W1 \in \Sigma s_1^*$ et $(ww2, W2)$, $W2 \in \Sigma s_2^*$

Contre-exemple

$L1 = \{a^n b,\ n \in \mathbb{N}\}$ $\quad \Sigma e1 = \{a,b\}$ $\quad \Sigma s1 = \{X,Y\}$

$L2 = \{a^n c,\ n \in \mathbb{N}\}$ $\quad \Sigma e2 = \{a,c\}$ $\quad \Sigma s2$ $\quad \{W,Z\}$

Il existe deux transducteurs T1 et T2 acceptant respectivement L1 et L2 et définissant les traductions :

$\tau(T1) = \{(a^n b,\ X^n Y)\}$

$\tau(T2) = \{(a^n c,\ W^n Z)\}$

Nous allons montrer qu'il n'existe pas de transducteur déterministe T définissant la traduction $\tau(T) = \tau(T1) \cup \tau(T2)$

Supposons qu'un tel transducteur existe : la seule possibilité est alors de définir δ de la façon suivante :

a se réécrit ϵ

b se réécrit $X^n Y$

c se réécrit $W^n Z$

T ayant un nombre fini d'états, et n ne pouvant être pris aussi grand que l'on veut, on ne peut pas connaître la valeur de n lorsqu'on a reconnu les symboles b ou c.

b. Proposition 2

Soit l'ensemble des préfixes P1 (P2) des mots de L1 (L2). Une condition suffisante d'existence d'un transducteur déterministe T acceptant L1 ∪ L2 et tel que $\tau(T) = \tau(T1) \cup \tau(T2)$ est que $P1 \cap P2 = \emptyset$

La démonstration est semblable à celle relative aux automates (I.5.1.1).
Ces résultats peuvent être généralisés à l'union de n langages états finis.

I.5.3 Conclusion

Le langage des descriptions étant l'union de plusieurs sous-langages disjoints, chaque description appartient donc a priori à un seul sous-langage. L'information discriminante permettant de déterminer ce sous-langage est le préfixe constitué par l'origine et le type de l'objet associé à la description (cf. I.3.1). Puisque ces préfixes sont tous distincts, il existe donc un transducteur d'états finis déterministe qui accepte l'union de ces sous-langages et traduit correctement les chaînes d'entrée ; un modèle déterministe est suffisant pour effectuer la transduction des données que l'on a à traiter. Ceci ne serait plus vrai pour des données de nature différente où l'on ne pourrait pas se baser sur cette notion de préfixe (par exemple, si l'information discriminante n'était plus en tête).

I.6 LANGAGES REGULIERS ET FERMETURE

I.6.1 Introduction

Nous rappelons qu'un langage L est régulier ssi il existe un automate d'états finis M tel que L = L(M). Nous allons maintenant choisir l'approche régulière car celle-ci constitue un outil plus commode pour étudier certains opérateurs définis pour les langages et leurs propriétés. Ceux-ci nous permettront de caractériser le langage d'interrogation de la base, c'est-à-dire le langage défini par l'ensemble des questions.

Le processus réel de formulationd'une question étant inconnu, la définition d'un

modèle nécessite une analyse a posteriori des différents types de formulations possibles et de leurs relations avec les objets de la base. Les opérations que nous définissons permettent de rendre compte effectivement des phénomènes observés tels que la substitution synonymique, l'abréviation et la permutation de mots. Elles expriment le comportement d'un utilisateur devant formuler la description d'un objet. Ceci nous conduit à proposer un modèle pour l'indexation des descriptions et la recherche des objets dans la base.

I.6.2. Définitions (rappel) [SAL]

a) Expression régulière

Soient Σ un alphabet fini

et $I = \{+,*,\emptyset,(,)\}$ tels que $\Sigma \cap I = \{\emptyset\}$

Une expression régulière sur Σ est un mot de $\Sigma \cup I$ satisfaisant la condition suivante :

(1) x, $\forall x \in \Sigma$ et $\emptyset$ sont des expressions régulières sur Σ

(2) si α et β sont des expressions régulières sur Σ, alors $(\alpha+\beta)$, $(\alpha\beta)$, α^* le sont aussi

(3) Seules les expressions construites à partir d'un nombre fini d'applications de (1) et (2) sont des expressions régulières.

b) Langage régulier

Chaque expression régulière α sur un alphabet Σ définit un langage $L\alpha$ sur Σ selon les conventions suivantes :

(1) $L\emptyset$ est le langage vide

(2) $\forall x \in \Sigma$ $Lx = \{x\}$

(3) $\forall \alpha,\beta$ expressions régulières sur Σ

$L(\alpha+\beta) = L\alpha \cup L\beta$

$L(\alpha\beta) = L\alpha\ L\beta$

$L\alpha^* = (L\alpha)^*$

Un langage L est régulier ssi il existe une expression régulière α telle que $L = L\alpha$.

I.6.3. Propriétés de fermeture

I.6.3.1 Substitution

Définition 1

Soient Σ et Δ deux ensembles finis

Une substitution est une fonction φ qui associe à tout élément de Σ un langage.
$\forall x \in \Sigma \ \varphi(x) \subseteq \Delta^*$
On étend φ à Σ^* de la façon suivante :

$$\varphi(\varepsilon) = \{\varepsilon\}$$
$$\forall \alpha \in \Sigma^* \quad \varphi(\alpha x) = \varphi(\alpha)\,\varphi(x)$$

On étend φ à un langage en définissant

$$\varphi(L) = \bigcup_{\alpha \in L} \varphi(\alpha)$$

Théorème 1.
Si L est un langage régulier et φ une substitution telle que $\forall x \in \Sigma \ \varphi(x)$ est régulier, alors $\varphi(L)$ est régulier.
(on peut trouver la démonstration dans [HRS]).

I.6.3.2 Définition des opérateurs a, c, t

Les définitions de ces trois opérateurs sont empruntées à [SAL].

Définition 2 : opérateur d'abréviation (a)

Un mot P' obtenu à partir d'un mot P en omettant certaines lettres est une abréviation de P. Le mot vide et P sont des abréviations de P. Un mot de longueur n possède au plus 2^n abréviations.

On définit récursivement la fonction suivante a d'un ensemble d'expressions régulières dans lui-même :

1. $a(\emptyset) = \emptyset$; $a(x) = x + \varepsilon \ \forall x \in \Sigma$
2. si α et β sont des expressions régulières
   $a(\alpha+\beta) = a(\alpha) + a(\beta)$
   $a(\alpha\beta) = a(\alpha)\,a(\beta)$
3. $a(\alpha^*) = (a(\alpha))^*$

Pour toute expression régulière α, La(α) est l'ensemble de toutes les abréviations des mots de Lα.

On note a(L) l'ensemble de toutes les abréviations des mots de L.

Définition 3 : opérateur de commutativité (c)

c est l'opérateur sur les langages, tel que c(L) est le langage défini par l'ensemble de tous les mots obtenus en permutant les lettres dans les mots de L.

Ainsi, pour $L = \{x^n y^n, n \in \mathbb{N}\}$, c(L) est l'ensemble de tous les mots où le nombre d'occurrences de x est égal au nombre d'occurrences de y.

Définition 4 : opérateur trace (t)

La trace d'un mot P est n'importe quel mot (mot vide inclus) obtenu par concaténation de certaines lettres de P.
Exemple : ε,x,y,xy,yx,yy,yyx,xyy, yxy
sont les traces du mot yxy.

On note t(L) le langage défini par l'ensemble de toutes les traces des mots L.
On démontre facilement que t(L) = a(c(L))

I.6.3.3. Incidence des opérateurs sur la régularité des langages

Théorème 2.
Si L est un langage régulier, alors a(L) est régulier.

démonstration :

si L est régulier, il existe une expression régulière telle que L = Lα.
a(α) étant une expression régulière, La(α) = a(L) est un langage régulier

Proposition 3.
L'opérateur c ne préserve pas la régularité des langages.

contre-exemple :
Soit $L = (01)^*$. L est un langage régulier c(L) est l'ensemble de tous les mots sur {0,1} ayant le même nombre de 0 et de 1. c(L) est un langage hors contexte.

Théorème 4.
Si L est un langage régulier, alors t(L) est régulier.

démonstration

1. Nous allons tout d'abord démontrer que :

$\forall L1, \forall L2\ t(L1 \cup L2) = t(L1) \cup t(L2)$

. $t(L1 \cup L2) \subseteq t(L1) \cup t(L2)$

$\forall w, w \in t(L1 \cup L2)\ \exists w'\ w' \in L1 \cup L2$ tq $w = t(w')$

donc $w \in t(L1)$ ou $w \in t(L2)$

$\Rightarrow w \in t(L1) \cup t(L2)$

. $t(L1) \cup t(L2 \subseteq t(L1 \cup L2)$

$\forall w, w \in t(L1) \cup t(L2)$

$\exists w'\ w' \in L1$ ou $w' \in L2$ tq $w = t(w')$

$\Rightarrow w \in t(L1 \cup L2)$

2. Toute expression régulière α peut se mettre sous la forme $\sum_{i=1}^{n} \alpha i$ où $\forall i\ \alpha i$ est une expression régulière où le "+" n'apparait pas.

On transforme α en utilisant les égalités suivantes :

Pour toutes expressions régulières β, γ, δ

$\beta(\gamma+\delta) = \beta\gamma + \beta\delta$

$(\beta+\gamma)\delta = \beta\delta + \gamma\delta$

$(\beta+\gamma)^* = (\beta^*\gamma^*)^*$

3. Soit α une expression régulière sur un alphabet Σ où seuls les opérateurs concaténation et "*" apparaissent. On peut ordonner Σ de sorte que :

$\Sigma = \{a1, a2, \ldots, ak, ak+1, \ldots, an\}$

où ai $1 \leq i \leq k$ est une lettre dont le nombre pi d'occurrences dans un mot est fini.

et ai, $k < i \leq n$ une lettre dont le nombre d'occurrences peut être infini.

Pour toute expression régulière α sur Σ :

$t(\alpha) = t(a1^{P1}\ a2^{P2} \ldots ak^{Pk}\ ak+1^*\ ak+2^* \ldots an^*)$

en utilisant la commutativité

Pour simplifier la démonstration, prenons $\Sigma = \{a,b\}$

(dans le cas où Σ est réduit à une lettre, elle est triviale).

Ceci revient à étudier 4 cas d'expressions :

(1) $a^p b^q$

(2) $a^* b^q$

(3) $b^* a^p$

(4) $a^* b^*$

(1) $a^p b^q$ est un langage fini

$\Longrightarrow t(a^p b^q)$ est fini donc régulier

(4) $t(a^* b^*) = (a+b)^*$

(2) (3) $t(b^* a^p) = b^* + (b^* a)b^* + \ldots + (b^* a)^i b^* + \ldots + (b^* a)^p b^*$

$$= \sum_{i=0}^{p} (b^* a)^i b^*$$

La démonstration est analogue lorsque Σ contient plus de deux lettres, mais il est plus simple dans ce cas de construire l'automate correspondant.

L'automate est défini par :

- l'ensemble des états <i1, i2,..., ik>

où $1 \leq i1 \leq p1, \ldots, 1 \leq ik \leq pk$

plus un état ERREUR

Tous ces états (excepté ERREUR) sont des états finaux

- la fonction de transition suivante :

$\delta(<i1,i2,\ldots,ik>,aj)$ = si $ij<pj$ alors $<i1,i2,\ldots,ij+1,\ldots,ik>$

sinon ERREUR pour $1 \leq j \leq k$

$\delta(<i1,i2,\ldots,ik>,aj) = <i1,i2,\ldots,ik>$ pour $k<j$

$\delta(\text{ERREUR},aj) = \text{ERREUR}$

Le rôle de cet automate est simplement de vérifier que le nombre d'occurrences d'un aj, $1 \leq j \leq k$ dans un mot est compris entre 0 et pj.

4. $\forall \alpha \ \exists \alpha 1, \alpha 2 \ldots \alpha n$ ne contenant pas le "+" tq

$\alpha = \sum_{i=1}^{n} \alpha i$ d'après 2.

$t(\alpha i) = \beta i$ où βi est une expression régulière

$t(\alpha i) = t(\sum_{i=1}^{n} \alpha i) = \sum_{i=1}^{n} t(\alpha i) = \sum_{i=1}^{n} \beta i$ d'après 1.

donc $t(\alpha)$ est une expression régulière

Nous avons donc démontré que t(L) est régulier et de plus qu'il est possible de construire la trace d'un langage régulier.

Théorème général

La trace de tout langage est un langage régulier.

La démonstration est similaire à la précédente. On distingue les lettres pouvant apparaître une infinité de fois dans un mot et celles dont le nombre d'occurrences est fini.

Dans le premier cas, les lettres sont sous l'opérateur "*" dans la trace. Dans le second cas, le nombre d'occurrences est borné.

exemple : $t(a^n b^n) = t(a^{2^n b^n}) = (a+b)^*$

Mais dans le cas où le langage n'est pas régulier, on ne peut pas forcément construire la trace. C'est pourquoi nous avons séparé dans la démonstration le cas des langages réguliers.

I.7 MODELISATION DE LA FORMULATION D'UNE DESCRIPTION

I.7.1 Synonymie et normalisation

La substitution de certains mots dans une description par des éléments synonymes produit une paraphrase de la description originale. La paraphrase obtenue peut être strictement équivalente (substitution d'un mot par un mot strictement synonyme) ou équivalente au sens large (substitution d'un mot par une liste de mots qui ne constitue pas à proprement parler un synonyme mais plutôt une définition ou une explication du mot original).

exemples : "UE12P" et "CUSN12" (alliage cuivre) sont strictement synonymes .
dans UN27-801-53ZJ (connecteur, origine ATI)
"Z" a comme synonyme au sens large "contact à souder sur fil"

Nous rappelons d'autre part, que des mots peuvent être interchangeables quelque soit le contexte ou dans un contexte déterminé (cf. I.1.4 et exemples ci-dessus).

La reconnaissance de termes synonymes permet de normaliser toute description, cette opération devant être effectuée aussi bien à la saisie qu'à l'interrogation. A la saisie, une description est dite normalisée si elle est standard.

a. Synonymie

Ces relations de synonymie peuvent être définies par une fonction de substitution (cf. I.6.3.1) de la façon suivante :

soient L un sous-langage régulier sur $\Sigma = \{ai, 1 \leq i \leq n\}$,
Δ un ensemble fini et φ une substitution telle que :
$\forall ai \in \Sigma \ \varphi(ai) \subseteq \Delta^*$ et $\varphi(ai)$ est un langage régulier

On fait de plus les hypothèses suivantes :

- $\forall i, \varphi(ai)$ est un langage fini
- $\forall i, \forall j \ \varphi(ai) \cap \varphi(aj) = \emptyset$ pour $i \neq j$

- $\forall i$, $ai \in \varphi(ai)$

On définit aussi des classes d'équivalences sur $\bigcup_{i=1}^{n} \varphi(ai)$ données par $[ai] = \varphi(ai)$, ai étant choisi comme représentant.

Les données des classes d'équivalence [ai] permet de définir des relations de synonymie (au sens strict ou large) dans un langage. Le représentant d'une classe sera appelé synonyme préférentiel.

Il faut noter que cette définition n'est valable dans la mesure où $\varphi(ai)$ est unique pour tout i. Or cette propriété n'est plus vérifiée s'il existe des cas de polysémie ou d'homographie dans le sous-langage considéré ; ceci est fréquent dans les expressions codées où un même mot peut avoir plusieurs interprétations différentes selon sa position dans le code et donc plusieurs ensembles de synonymes distincts.

exemple 44A1111-30-9

Les quatre "1" ont des interprétations différentes (cf. I.1.3)

La relation n'est alors plus une relation d'équivalence car elle n'est pas transitive.

b. Descriptions équivalentes

Soient $\alpha, \beta \in \varphi(L)$. On dit que α est directement équivalent à β (noté $\alpha <==> \beta$) si β peut être obtenu à partir de α en remplaçant une occurrence d'un élément par un élément appartenant à la même classe.

On définit la fermeture transitive <==> de <—>. On dit que α est équivalent à β ($\alpha <==> \beta$) si

$\exists \alpha 0, \alpha 1 \ldots \alpha n \in \varphi(L)$ tels que

$\alpha = \alpha 0 <—> \alpha 1 <—> \ldots <—> \alpha m = \beta$

ou si m = 0 $\alpha = \beta$

La relation <==> est une relation d'équivalence sur $\varphi(L)$

Les classes sont données par :

$[\alpha] = \{\beta / \alpha <==> \beta\}$

Cette relation permet de définir la notion de descriptions synonymes dans les langages concernés par cette étude.

Il est naturel de se demander si étant données deux phrases α et β de $\varphi(L)$, on peut effectivement dire si elles sont équivalentes pour la relation <==>.

Soit S = {[α], α∈φ(L)} l'ensemble des classes d'équivalence sur φ(L) ; on définit [α].[β] = [αβ] et on prend [ε] comme élément neutre. Avec ces définitions, S forme un monoïde.

Théorème général

Il n'existe pas d'algorithme qui, un ensemble fini de relations étant donné entre des mots d'un monoïde, puisse décider si deux mots appartenant à ce monoïde sont équivalents pour ces relations.

On peut trouver la démonstration de ce théorème dans [MAY].

Dans la pratique cependant, le problème ne se pose jamais à un tel niveau de généralité ; la présence de séparateurs par exemple, ou de toute autre information pour délimiter les mots, permet de résoudre ce problème. C'est dans ce cadre que nous nous plaçons.

c. Forme normalisée d'une description

Soient α,β∈φ(L)

On définit α comme étant la forme normalisée de β si α<==>β et α ne contient que des synonymes préférentiels. La transformation de β en α, étant donnée la substitution φ est appelée opération de normalisation.

I.7.2 Définition du langage d'interrogation

Le corpus de base est un ensemble fini de descriptions normalisées d'objets.
Cet ensemble, noté Lo, est régulier (fini).
Il est possible de définir le langage d'interrogation relatif à ce corpus à partir des opérateurs précédemment définis et de Lo (cf. I.6.3.2).

Soient c,a,t les opérateurs de commutativité, abréviation et trace et φ une fonction de substitution permettant de définir des classes de phrases équivalentes (synonymes) (cf. I.7.1).

On pose $L = \varphi(Lo) = \bigcup_{x \in Lo} \varphi(x)$

Lo étant un langage fini, les ensembles L, c(L), a(L), t(L) sont finis (dans ce cas particulier, c(L) est donc régulier).

On définit le langage d'interrogation comme un sous-ensemble de t(φ(Lo)).

Nous noterons $\emptyset_x$, C_x, T_x les ensembles relatifs à un élément x de Lo (x étant la description normalisée, c'est-à-dire celle qui est utilisée -et obligatoire- pour identifier l'objet) où :

- $\emptyset_x = \varphi(x)$ est l'ensemble de descriptions synonymes de x, cet ensemble pouvant être éventuellement réduit à l'élément x lui-même. $\emptyset_x$ est donc l'ensemble des paraphrases obtenues par substitution synonymique.

- $C_x \subseteq c(\emptyset_x)$ est l'ensemble des dexcriptions obtenues à partir des éléments de $\emptyset_x$ en permutant certains mots et tel que toute description de C est interprétable et son interprétation est identique à celle de x. C_x est l'ensemble des paraphrases obtenues par permutation à partir des éléments de $\emptyset_x$.

Soit Li le sous-langage de Lo auquel appartient x, nous dirons que Li est commutatif si $\forall x \in Li$, $C_x = c(\emptyset_x)$, partiellement commutatif si $\emptyset_x \subset C_x \subset c(\emptyset_x)$, non commutatif si $C_x = \emptyset_x$.

exemples : ROND BRONZE UA9 DIA.50 appartient à un langage commutatif.

VIS F/90 M4×10/8 LAITON NI 5 appartient à un langage partiellement commutatif (car M4×10/8 est un code).

UN27-29WL-SUP+3T appartient à un langage non commutatif (code).

- $T_x \subseteq (a(C_x) - \varepsilon)$ est l'ensemble des descriptions obtenues à partir des éléments de C_x en omettant certains mots mais de façon à ce que chacun des mots restant soit interprétable.

L'ensemble $T_x - C_x$ représente l'ensemble des descriptions incomplètes dont l'interprétation ne permet pas en général d'identifier un objet de façon unique.

exemples : ROND DIA.50

UN27-29WL-SUP

En résumé, une requête R relative à un objet déterminé x est :

- suffisante (car discriminante) si $R \in C_x$
- incomplète (et en général non discriminante puisqu'elle présente des omissions) si $R \in T_x - C_x$
- nous dirons qu'elle est "surabondante" si $\exists R1$ tel que $R1 \in a(R)$ et $R1 \in C_x$.

Un terme supplémentaire peut être soit une redondance dans la description

(ex. UN27-29WL-SUP+3T 29 CONTACTS), soit une contradiction (ex. UN27-29WL-SUP+3T 17 CONTACTS), soit une incohérence (ex. UN27-29WL-SUP+3T CD8B, car le terme "CD8B" (cadmié) ne s'applique pas à un connecteur)

- R peut présenter à la fois des omissions et contenir des termes en trop, c'est-à-dire $\exists R1$ tel que $R1 \epsilon a(R)$ et $R1 \epsilon T_x - C_x$.

Si R n'entre dans aucun des cas précédemment cités, R est dite totalement incompatible avec x.

I.7.3 Conclusion

L'analyse des différents types de requêtes d'interrogation nous a permis de définir le langage d'interrogation de la base.

Toute requête de ce langage peut être ramené à un élément du langage des descriptions (Lo) par une suite d'opérations (substitution, abréviation, permutation). Ceci nous amène à proposer un modèle pour l'indexation automatique des descriptions. Nous rappelons que notre objectif est de minimiser le silence lors de la recherche d'un objet dans la base. Toute description est normalisée à la saisie ; l'indexation consiste à extraire les mots de la forme normalisée et à leur associer une étiquette (catégorie sémantique ou étiquette formelle). La représentation d'une requête d'interrogation est semblable à celle d'une description. On effectue la recherche en sélectionnant les descriptions dont la représentation présente un maximum de termes communs avec la représentation de la requête.

CHAPITRE II

PRÉSENTATION DE LA RÉALISATION

II.1 METHODES ET OBJECTIFS

Nous avons souligné au chapitre 0 le caractère factuel de la base, les raisons pour lesquelles nous nous sommes orientés vers des techniques documentaires, ainsi que la nécessité d'une analyse préalable des données et des questions par des méthodes linguistiques.

L'étude générale du langage nous a permis de proposer (ch. I) un modèle linguistique et nous avons vu que le modèle d'état finis est suffisant pour représenter les contraintes syntaxiques et une grande partie des contraintes sémantiques du langage. Nous avons ensuite caractérisé les relations existant entre les questions et les descriptions contenues dans la base.

Nous présentons ici les méthodes utilisées dans le logiciel documentaire, celles-ci étant déduites de l'étude précédente, ainsi que les objectifs généraux fixés pour la réalisation.

II.1.1. Le processus d'acquisition

Le processus d'acquisition est l'ensemble des opérations que subissent les descriptions avant leur stockage dans la base.

Le processus se décompose en trois phases :

- contrôle
- normalisation
- indexation

Nous allons tout d'abord définir le but de l'indexation dans le cas particulier qui nous concerne qui est de produire un modèle de représentation de la connaissance.

II.1.1.1. L'indexation

Le but de l'indexation est de faciliter ultérieurement la recherche d'information. Les descriptions sont courtes et concises et l'association de la syntaxe à la sémantique est très simple; les catégories sémantiques associées aux mots d'une description peuvent être déterminées par le biais de la syntaxe.

L'indexation permet donc de fournir un descriptif complet (représentation de la connaissance) tout en se libérant des contraintes syntaxiques du langage et en autorisant des requêtes d'interrogation partielles.

Le texte normalisé d'une description est conservé à travers une liste reflétant son contenu sémantique; celle-ci est obtenue par transduction (cf. I.5). L'interprétation sémantique est réalisée par des couples descripteur-valeur, un descripteur dénotant une catégorie sémantique.

Exemple. Si une description normalisée s'écrit $x_1x_2...x_n$, x_i étant un mot du langage auquel est attaché le descripteur D_i,
le descriptif associé est : $((D_1,x_1)(D_2,x_2)...(D_n,x_n))$

La prise en compte de ces descripteurs lors de l'indexation est indispensable car elle permet de diminuer considérablement le coût de la recherche et d'autre part le bruit. En effet, en raison du principe même de la codification, à un même mot peuvent correspondre des contenus sémantiques différents, selon le contexte grammatical dans lequel il se présente. L'indexation porte donc à la fois sur les mots et leurs descripteurs associés.

II.1.1.2. Rôles de l'analyse grammaticale des descriptions à indexer

Nous avons déjà précisé les raisons qui nous ont conduit à choisir le niveau grammatical (ch. I) (la syntaxe véhicule une part importante de la sémantique du langage).

- *contrôle*. Le premier rôle de l'analyse consiste en un contrôle des descriptions lors de la saisie : une description est déclarée correcte si elle est reconnue conforme à la grammaire.

Une validation sémantique totale ne pouvant être effectuée (la grammaire décrit un sur-langage de l'ensemble réel des descriptions) le contrôle n'a donc pas un caractère absolu : certaines descriptions, bien que conformes, peuvent ne pas avoir d'interprétation (c'est-à-dire ne pas correspondre à un objet existant).

- *indexation*. Les mots considérés comme relevant d'une même catégorie sémantique sont regroupés en classes auxquelles sont associées des étiquettes identifiant ces catégories.

- *normalisation*. Le terme normalisation recouvre ici deux notions :
  a - Normalisation d'une description
  b - Normalisation des termes d'indexation

  a - Un objet possède un nombre fini de descriptions strictement équivalentes à sa description originelle, ceci étant dû à l'utilisation de termes synonymes (cf. I.7.1). La connaissance des classes de synonymie et de leurs représentants permet de transformer toute description acceptable en une forme dite normalisée qui est l'identificateur de l'objet. Il s'agit ici de ramener à une même forme des mots différents dont on sait qu'ils recouvrent le même concept.
  La reconnaissance et le remplacement des mots synonymes sont réalisés en faisant intervenir dans le dictionnaire la notion de synonymie.
  Cette normalisation est externe car envisagée du point de vue de l'utilisateur.

  b - La normalisation des termes consiste à éliminer ceux dont on sait qu'ils sont sémantiquement non significatifs et de ramener à une même forme les différentes représentations d'un même terme dues à des variantes grammaticales.
  Un cas particulier fréquent de mot vide, est un mot utilisé comme séparateur. Seul le critère grammatical permet la détermination de ces mots vides.

  Exemples. Le caractère "-" est séparateur dans 4V-15A,
  il ne l'est pas dans (-55 +125) (tolérance d'un circuit).
  Le caractère "/" est séparateur dans M3×16/12
  il ne l'est pas dans 44/1213-30-9

Sont également éliminés des mots tels que les mots de liaisons (et, à de, avec, en, ...) dans des références non codées.
Notons cependant que certains antonymes de mots vides, bien qu'ayant un statut grammatical identique, ne doivent pas être éliminés.

Exemple. Cas de "AVEC" et "SANS"

(1) BOITE AVEC PRESSE ETOUPE

(2) BOITE SANS PRESSE ETOUPE

"AVEC" est éliminé dans (1), "SANS" est conservé dans (2).

La reconnaissance des différentes formes d'un même terme, dues aux variantes grammaticales, se limite dans cette application à la reconnaissance et la suppression des suffixes de pluriels et de féminin dans les termes empruntés au langage naturel (ex. : noir, noire, noirs).

II.1.2. La recherche

La représentation d'une question est identique à celle d'une description (liste de couples descripteur-valeur). Notre objectif est qu'un objet puisse être retrouvé s'il existe dans la base (à condition qu'il ait été correctement enregistré). L'important est alors que dans l'ensemble des objets constituant la réponse à une question, se trouve l'objet cherché. On cherche donc une couverture de la solution réelle, ce qui en termes documentaires implique de choisir des méthodes qui favorisent l'obtention d'un taux de rappel élevé. Cet objectif est limité par le fait qu'une question n'est pas toujours interprétable.

II.1.2.1. Analyse d'une question

La grammaire relative aux questions correspond à un sous-ensemble de la trace du langage des descriptions acceptables à la saisie.
Le rôle de l'analyse est :

- fournir un diagnostic sur une requête (ceci pouvant être exploité comme aide à la reformulation d'une question)
- normaliser une question

- analyser le contenu d'une question, c'est-à-dire associer à chaque mot d'une question, un descripteur sémantique.
  La validation sémantique est reportée à la comparaison directe de la question avec l'ensemble des descriptions contenues dans la base.
  Si la question est grammaticalement correcte, les descripteurs associés dénotent des catégories sémantiques. Sinon, des critères autres que grammaticaux doivent être recherchés.

La méthode que nous avons choisie lors de la réalisation consiste à utiliser un second module d'analyse plus général, de type morphologique, où seuls des critères de forme sont utilisés (cf. II.2.5.2).

II.1.2.2. Mode de recherche

Le mode de recherche est déduit de l'étude présentée en I.7. Le descriptif associé à une question est considéré comme une expression booléenne de "et" implicites.

Le processus de recherche, c'est-à-dire la suite des opérations permettant d'aboutir à un résultat considéré comme final est présenté en II.3.2.

II.1.3. Objectifs généraux fixés pour la réalisation

Dans le domaine d'application concerné, les utilisateurs ne sont pas informaticiens. Le point fondamental est donc de ne pas entraver leurs activités par de trop fortes contraintes liées à la mise en oeuvre des logiciels.

Ces logiciels doivent offrir un interface d'exploitation le moins contraignant possible ainsi qu'une complexité minimale dans le déroulement des opérations offertes.

- Les langages de descriptions ne sont pas figés. La prise en compte de phénomènes nouveaux se traduit soit par une modification des spécifications d'un langage particulier (caractère évolutif des langages), soit par la spécification d'un nouveau langage. Ces interventions sont fréquentes et seul l'utilisateur final possède la compétence pour les réaliser.

- Si la question originale formulée par l'utlisateur n'aboutit pas à un résultat jugé satisfaisant, celui-ci ne doit pas se trouver en situation de blocage; il doit pouvoir examiner précisément les résultats obtenus et pouvoir modifier ou affiner sa question en conséquence.

L'objectif essentiel que nous nous sommes fixé est donc de faciliter la communication homme-machine. Ceci se traduit par la conception de logiciels conversationnels caractérisés par des commandes simples.

Deux logiciels ont été réalisés au stade de prototypes :

- un logiciel d'analyse : PIAFBCN
- un logiciel de constitution et d'interrogation d'une base de donnée de nomenclature.

Ces deux produits ont été réalisés séparément, dans des langages différents. Pour la recherche, un interface assure la communication entre les deux logiciels. Cet interface serait le même pour les opérations de constitution et de mise à jour de la base, mais pour des raison de sécurité dans une phase expérimentale, nous avons choisi un mode de fonctionnement indépendant.

La suite de ce chapitre est consacrée à la présentation des deux logiciels.

II.2. PRESENTATION DU TRANSDUCTEUR PIAFBCN

Nous avons vu qu'il était important de disposer d'un logiciel interactif d'analyse. D'autre part, en raison du caractère dynamique du langage, nous avons recherche un outil paramétré.

Nous présentons ici un logiciel construit à l'aide du transducteur général d'état finis utilisé dans le système PIAF (Programme Interactif d'Analyse du Français) [COU]. PIAFBCN en constitue un cas d'application particulier. Les principes généraux du système PIAF sont présentés en Annexe I, la description détaillée pouvant être trouvée dans [COU] [CDG].

II.2.1. Objectifs généraux du logiciel

Le premier objectif est la définition d'un outil permettant à l'utilisateur un enrichissement progressif de la grammaire. PIAFBCN bénéficie de possibilités d'interaction semblables à celles du logiciel PIAF :

- définition et mise au point interactive des paramètres linguistiques
- utilisation conversationnelle de l'outil.

Bien que la définition des modèles linguistiques soit relativement complexe pour un non informaticien, la mise au point est rapide et commode (possibilité de visualiser les résultats d'une mise à jour de la grammaire immédiatement et sur autant d'exemples qu'on le désire).

D'autre part, cet outil permet d'effectuer l'opération de normalisation en définissant des classes d'équivalence de chaînes dans le dictionnaire et des modèles de transduction.

II.2.2. Restrictions d'implémentation

La première restriction réside dans le fait que le même outil (logiciel et paramètres) est utilisé à la fois lors de la saisie et lors de la recherche. Ceci a comme principal inconvénient d'affaiblir le contrôle à la saisie, puisque le langage d'interrogation est un sur-langage de celui des descriptions. Cette solution a été choisie, dans une première phase de réalisation, pour des raisons de simplicité.

D'autre part, les omissions sont rares à la saisie (on trouve en général plutôt des redondances).

Une seconde limitation du contrôle provient de l'absence de possibilité de définir des ensembles de valeurs par des propriétés formelles (types). Toutes les chaînes du langage sont actuellement définies par énumération (dictionnaire). Le problème que cela pose concerne essentiellement les mot codés numériquement. Pour le résoudre (du mbins partiellement), nous avons été conduits à définir plusieurs niveaux de transduction (cf. II.2.3). Cette solution est justifiable dans la mesure où les erreurs les plus probables au niveau des mots codés ne sont de toute façon pas décelables. Ce cas se produit lorsque :

- une erreur dans une description identifiant un objet produit une autre description, sémantiquement correcte mais identifiant un autre objet
- une description est grammaticalement correcte mais sans interprétation (cf. I.3.3).

II.2.3. Les différents niveaux de transduction

La solution que nous avons adoptée consiste à définir plusieurs niveaux de traitement, chacun correspondant à une famille de langages ayant des propriétés communes. Les transducteurs sont hiérarchisés par niveau de contrôle décroissant.

Les familles de base sont les suivantes :

1 - famille "descriptive". Elle est définie par l'ensemble des descriptions dans lesquelles tous les mots ont une interprétation indépendante du contexte et de leur position. Cet ensemble est la réunion des sous-langages commutatifs (cf. I.7.2);

2 - famille codée. Elle est définie par l'ensemble des descriptions entièrement codées et est la réunion des sous-langages non commutatifs. Elle est elle-même constituée de deux sous-familles disjointes :

- les codes "décomposables", où les mots ont une interprétation dépendant à la fois du contexte et de leur position dans le code
- les codes "indécomposables", où les mots sont dénués de toute interprétation directe (ils jouent un rôle d'identificateur).

Tout autre famille est qualifiée d'hybride. Nous rappelons ici quelques exemples :

- famille "descriptive" :
  BANDE LAITON UZ10 EP.0,6 LARG.100 ECROUI 1/4DUR
- famille codée :
  décomposable : 44/1221-30-0/2-9 (fil)
  indécomposable : 1N5646A (diode)
- famille hybride :
  CE33-2U 270PF ±10% (condensateur)
  (code)

II.2.3.1. Les deux types de transduction

L'analyse fournit pour chaque mot (ou groupe de mots) interprétable un descripteur dénotant la classe sémantique à laquelle il appartient.

Dans le but de minimiser le silence lors de la recherche, les chaînes non interprétables sont segmentées. En effet, ces codes peuvent être aussi sujets à des erreurs à l'interrogation. Les mots sont extraits selon des critères purement formels (cf. I.2.2.2) et les descripteurs qui leur sont associés dénotent un type.

En pratique, une telle chaîne est segmentée en fonction du type des caractères qui la composent (lettre → typeα, chiffre → type N), les mots étant déterminés soit par un changement de type, soit par un séparateur. Les descripteurs dénotent le type des mots (Pα ou PN, P ∈ N).

Exemple. 1N5646A $\xrightarrow{\text{analyse}}$ ((1N,1)(1α,N)(4N,5646)(1α,A))

II.2.3.2. Langage reconnu par la grammaire

Lors de la définition de la grammaire, nous avions à prendre en compte les faits suivants :

- la diversité des sous-langages de description.
  Le nombre de ces sous-langages s'élève à plusieurs centaines ;

- l'analyse grammaticale doit fournir une interprétation (lorsque cela est possible) des différents mots d'une description aussi bien à la saisie qu'à la recherche. Il faut donc pouvoir accepter certaines omissions, redondances ou permutations de mots.

La solution que nous avons adoptée consiste à considérer une description, non plus comme un élément d'un sous-langage particulier, mais comme une liste de mots sur laquelle est définie une relation d'ordre a priori partielle.

Exemple. Dans CE33-2U 270PF ±10% (famille hybride)
la relation d'ordre est la suivante :
"CE < "33" < "2U"
où le symbole "<" signifie "précède immédiatement"

Les cas extrêmes sont : la relation d'ordre est totale (famille codée), il n'y a pas d'ordre défini (famille "descriptive")

On considère dans cette liste, les sous-listes maxima dans lesquelles la relation d'ordre sur les mots est totale. Tous les mots restants constituent des sous-listes réduites à un élément.

Exemple. Soit D ≡ m1 m2 m3 m4 m5 m6 m7
et la relation d'ordre partielle :
m3<m4<m5<m6
On obtient alors 4 sous-listes : (m1)(m2)(m3 m4 m5 m6)(m7)

La grammaire définie est la grammaire reconnaissant toute combinaison de ces sous-listes. Elle exprime la partie de la syntaxe suffisante à l'interprétation d'une description. Cette méthode a comme avantage de réduire considérablement la taille de la grammaire.

II.2.3.3. Les mots codés interprétables

En raison du principe même de la codification, le vocabulaire des codes est très restreint et les cas de polysémie sont donc fréquents. (La majorité des codes sont numériques et les informations sont en général codées sur 1 ou 2 caractères).
D'autre part, pour éviter d'énumérer tous les mots codés interprétables dans le dictionnaire (chaque mot apparaissant autant de fois qu'il a d'interprétations possibles selon le contexte), nous avons défini les classes de mots codés de façon générique (définition d'un type) plutôt que par extension.

En reprenant les notations précédentes, les types élémentaires utilisés sont : Pα, PN. Si un caractère n'est ni une lettre, ni un chiffre, son type est représenté par le caractère lui-même (ex. : /, -, +, ...).

Ces types élémentaires peuvent être combinés : PαQN (P lettres suivies de Q chiffres), PNQα, Pα/QN, ...

Le type d'une classe est l'union des types des mots qui la composent. Ainsi, si une classe contient uniquement des mots composés d'un caractère numérique et des mots composés de deux caractères numériques, son type est 1N ou 2N.

Exemple. Le code des connecteurs, série UN27-801 (origine : ATI) est défini par : *UN27-801* NBR_CONT TYPE_CONT TYPE_FIX
La classe TYP_FIX (type de fixations) contient les éléments :
A, B, C, D, E, F, GC2, HC2, OC2, I, J, K, L, J/N, I/N
Le type de cette classe est : 1α ou 2α1N ou 1α/1α

Au lieu d'énumérer tous les mots d'une classe dans le dictionnaire, on introduit simplement son type en le décrivant d'une façon symbolique.

Exemple. Si une classe a comme type 1α ou 2α1N ou 1α/1α, on introduit dans le dictionnaire les chaînes : "A", "AA0","A/A" où le caractère "A" symbolise les lettres et "0" les chiffres.

II.2.3.4. Image αN d'une chaîne

Nous appelons "image αN" d'une chaîne, la chaîne obtenue, à partir de la chaîne réelle d'entrée, par les transformations suivantes :

- les caractères spéciaux ainsi que les caractères représentant une information discriminante pour l'analyse sont conservés. Ces derniers sont les préfixes des codes qui désignent un type ou un numéro de série (ex. : "44" dans 44/1221-20-0/2-9, "UN27-801" dans UN27-80153ZJ)
- les autres caractères sont représentés par un symbole particulier dénotant leur type (α,N).

Exemple. L'image αN de 44/1221-20-0/2-9 est 44/NNNN-NN-N/N-N

II.2.3.5. Les trois niveaux de transduction

Trois niveaux de transduction ont été définis selon le niveau d'interprétation et la façon dont sont définies les classes de mots. Aucune hypothèse ne pouvant a priori être faite sur la nature d'une chaîne, ces transductions sont hiérarchisées par niveau du contrôle décroissant.

- T1 correspond aux classes des mots interprétables définies par énumération
- T2 correspond aux classes de mots interprétables définies par un type générique
- T3 correspond aux mots non interprétables.

L'analyse produit pour toute description $D \equiv m1\ m2 \ldots m_p$ une transduction finale de la forme $((D1,m1)(D2,m2)\ldots(Dp,m_p))$, D_i étant un descripteur. Selon la nature de m_i, la transduction partielle (D_i,m_i) est produite par le niveau T1,T2 ou T3.

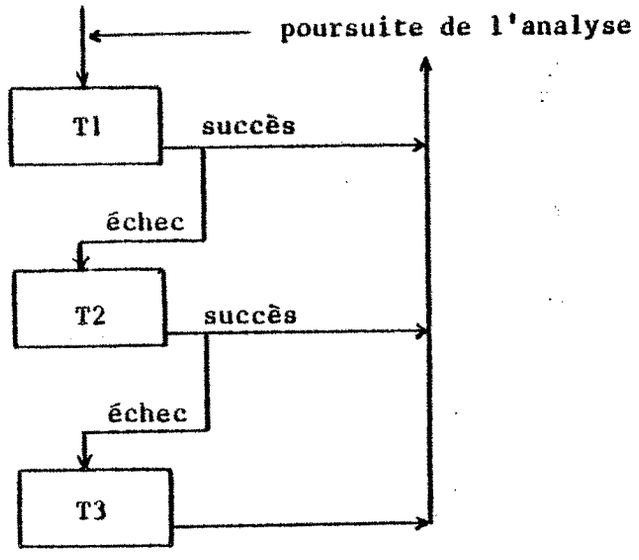
La chaîne d'entrée correspondant à chaque niveau et la nature de la transduction produite sont récapitulées dans le tableau suivant :

	chaîne d'entrée	transduction produite
T1	chaîne réelle	D_i dénote une catégorie sémantique
T2	image αN	D_i dénote une catégorie sémantique
T3	image αN	D_i dénote un type

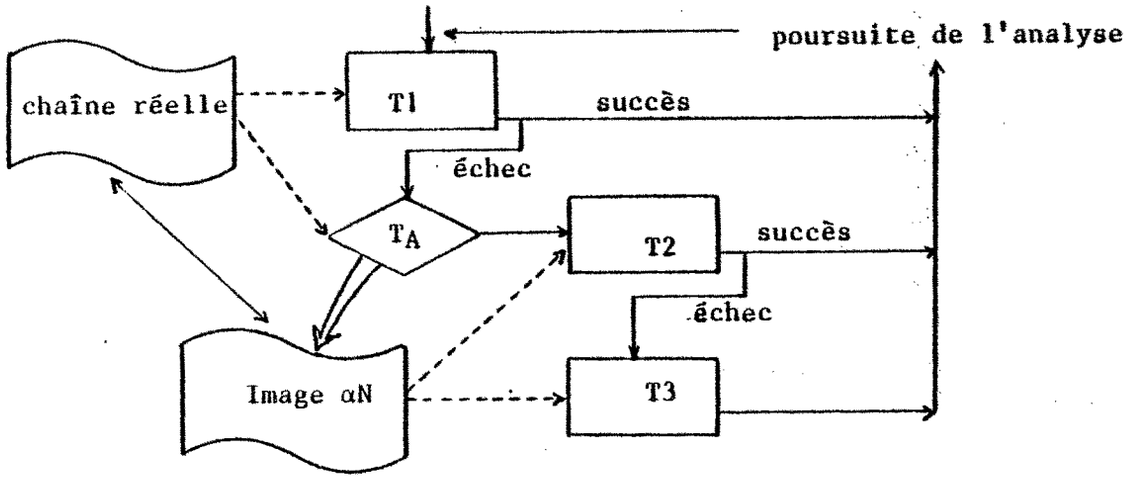
II.2.4. Fonctionnement du transducteur PIAFBCN

II.2.4.1. Enchaînement des différents transducteurs

L'analyse d'une description se déroule en parcourant successivement les trois niveaux jusqu'à l'obtention d'un résultat.
En dernier ressort, toute chaîne est acceptée au niveau le plus bas (T3) puisqu'il n'y a aucun contrôle.



L'image αN d'une chaîne est produite par un transducteur auxiliaire T_A



Le premier transducteur T_1 a en entrée la chaîne réelle. En cas d'échec de T_1, le transducteur auxiliaire T_A reprend la chaîne origine et la transforme en une image αN. Celle-ci est l'entrée des transducteurs T_2 et T_3, la chaîne origine étant conservée en parallèle pour produire la transduction. En cas d'échec de T_2, T_3 est alors activé. L'analyse de la chaîne restante est ensuite poursuivie en reprenant au niveau T_1.

II.2.4.2. Transductions produites

Les transducteurs T_A, T_2, T_3 produisent un résultat unique. Par contre, le résultat de T_1 ne l'est pas forcément. En effet, l'ignorance du contexte d'un mot crée des ambiguïtés (polysèmes, ou ambiguïtés de découpage).

Exemples. (1) Le mot "U" est une forme pour un profilé mais c'est une finition pour une vis.

(2) "RNF100-1" et "RNF100" sont deux types de gaines (origine : RAYCHEM).
Si la chaîne est "RNF100-1" est suivie d'une dimension (ex. RNF100-1 1/2..., celle-ci étant exprimée en pouce) cette chaîne est un mot représentant effectivement le type de la gaine. Sinon, le type est "RNF100" et "1" représente la dimension.

II.2.4.3. Les paramètres linguistiques

a. Le dictionnaire

Le dictionnaire contient outre les caractères (lettres, chiffres, symboles spéciaux) (sous-ensemble correspondant en particulier au niveau TA)

- les symboles de base correspondant au niveau Tl
- les groupes de caractères discriminants dans les codes interprétables
- les définitions symboliques des types génériques (cf. II.2.3.3)

Nous rappelons que tout élément du dictionnaire fait obligatoirement référence à un modèle de transduction (Annexe I).

La commande utilisée pour introduire un élément dans le dictionnaire permet de mettre en évidence les attributs de cet élément :

/ élément$_1$ [(S)] [(*)] / modèle / [élément$_2$ /]

Les éléments entre crochets sont facultatifs."S" et "*" sont deux indicateurs. Leur signification est la suivante :

- il est possible de définir des classes d'équivalence d'éléments (mots ou segments) dans le dictionnaire en indiquant le représentant choisi. Ces classes ont été définies pour la relation de synonymie. Le représentant (ou synonyme préférentiel) traduit la forme normalisée de l'élément. Cette relation est représentée symboliquement par l'indicateur "S".

Exemple / ALU(S) / MAT / ALUMINIUM /

le mot "ALU" est indexé sur le modèle MAT. Il a comme synonyme "ALUMINIUM" (qui est un autre mot du dictionnaire).

- nous avons vu qu'à l'issue d'une transduction T_1, le résultat n'est pas forcément unique. En particulier ayant trouvé un résultat, il peut en exister un autre à partir d'un élément du dictionnaire plus court.

Pour interdire cette possibilité, on utilise l'indicateur "*". Celui-ci indique que la chaîne est indécoupable.

Exemple / CLASSE 18-10MO (*) / NUAN /
/ CLASSE 18 / NUAN /

b. Généralités sur la grammaire

La définition des modèles fait intervenir des validations de règles qui représentent les divers états de l'automate. A toute règle validée correspond une opération de transduction particulière. Le mécanisme de transduction est différent selon le niveau auquel la règle correspond. Les informations traduites sont :

- au niveau TA : une image αN
- aux niveaux T_1, T_2, T_3 :
  . un descripteur (dénotant une catégorie sémantique (T_1,T_2) ou un type (T_3)), cette opération étant réalisée soit pour toute règle validée

(T_2,T_3) soit lorsque l'automate atteint un état final (T_1)

. une chaîne, celle-ci étant formée de la concaténation de sous-chaînes d'origine ou synonymes.

Le principal général de la grammaire est le même que dans le système PIAF [COU].

Nota Un modèle particulier "VIDE" permet de déclarer des mots vides. Ces mots ne figurent pas dans la chaîne de sortie (leur traduction est le caractère vide ε).

III.2.4.4 Exemples d'analyse

(1) L'analyse de BANDE LAITON CUZN10 EP.0,6 1/4 DUR produit

((DENO,BANDE)(MAT,LAITON)(NUA,UZ10)(DIM,0,6)(ELA,1/4DUR))

Le mot CUZN10 a été normalisé en UZ10.

Les descripteurs correspondent dans l'ordre à : dénomination, matière, nuance, dimension, élaboration.

(2) VIS HC BT CUV. M3×8 U ACIER QUAL.8.8 CD8B

↳ ((DENO,VIS)(FORP,HC)(FORC,BOUT CUVETTE)(DIM,M3×8)
[(FINI,U)(FORP,U)] (MAT,ACIER)(NUA,QUAL.8.8)(TRAI,CD8B))

Les ambiguïtés (U : forme ou finition) sont énumérées. L'utilisateur doit alors les lever de manière interactive.

descripteurs : dénomination, forme principale, forme complémentaire, dimension, finition, matière, nuance, traitement.

(3) UN27-804-29ZF-J

↳ ((TYPE,UN27-804)(NCON,29)(TCON,ZF)(TFIX,J))

descripteurs : type, nombre de contacts, type de contacts, type de fixation.

(4) 1N5665A

↳ ((1N,1)(1α,N)(4N,5665)(1α,A))

Dans les exemples (1) et (2), la transduction est produite par le niveau T_1 uniquement ; dans l'exemple (3) par le niveau T_2 et dans l'exemple (4) par le niveau T_3

II.2.5 Utilisations de PIAFBCN

II.2.5.1 Saisie

Dans le but de minimiser le silence lors de la recherche, une redondance est créée au niveau de l'analyse lors de la saisie.

Une seconde traduction est produite par un modèle d'analyse plus général, utilisant des critères autres que grammaticaux. En pratique, nous avons utilisé le transducteur T_3 pour la produire. Cette double analyse est envisagée systématiquement, excepté dans le cas d'une description codée indécomposable (car les deux transductions seraient identiques).

Deux descriptifs sont donc associés en général à une description : pour la première transduction, l'utilisateur doit obligatoirement lever les ambiguïtés s'il en existe et la seconde est unique.

II.2.5.2 Recherche

Deux cas peuvent se présenter :

- La requête est grammaticalement correcte. L'analyse se déroule de façon identique mais on accepte que l'utilisateur ne puisse pas être en mesure de lever les ambiguïtés qui se présentent. Plusieurs descriptifs sont alors produits.
- La requête est incorrecte. Elle est alors analysée automatiquement au niveau T_3. La redondance créée à la saisie est alors utilisée pour effectuer la recherche (cf. II.2.5.1).

II.2.5.3 Utilisation générale

Un ensemble de commandes et de langages permet l'utilisation en mode conversationnel du transducteur PIAFBCN. Les possibilités offertes diffèrent selon le contexte d'utilisation (saisie, recherche).

Les commandes d'utilisation permettent principalement :

- L'accès au dictionnaire et à la grammaire en cours ou non d'exécution (pour interrogation ou mise à jour de manière conversationnelle)
- La correction de la chaîne d'entrée en cours d'exécution
- Un traitement assisté des ambiguïtés (besoin de dialogue)

La consultation et la mise à jour du dictionnaire et de la grammaire se font à l'aide de langages dont la définition est donnée dans [CDG].

L'ensemble de ces commandes et leur enchaînement possible est donné par le diagramme de l'annexe II ; des exemples de sessions sont donnés en annexes III et IV.

II.2.6 Caractéristiques de l'implémentation

a. PIAFBCN a été développé à partir d'une version opérant sous CP/CMS de PIAF. Le logiciel est écrit en PL360. L'analyse d'une description nécessite environ 0,04 s de temps CPU sur un IBM 370, la longueur moyenne d'une description étant de 25 caractères.

   L'expérimentation s'est déroulée sur un échantillon d'environ 10.000 références-articles d'origine et de nature diverses. Les ambiguïtés que nous y avons rencontrées sont rares.

b. Réalisation de la communication PIAFBCN-SOCRATE

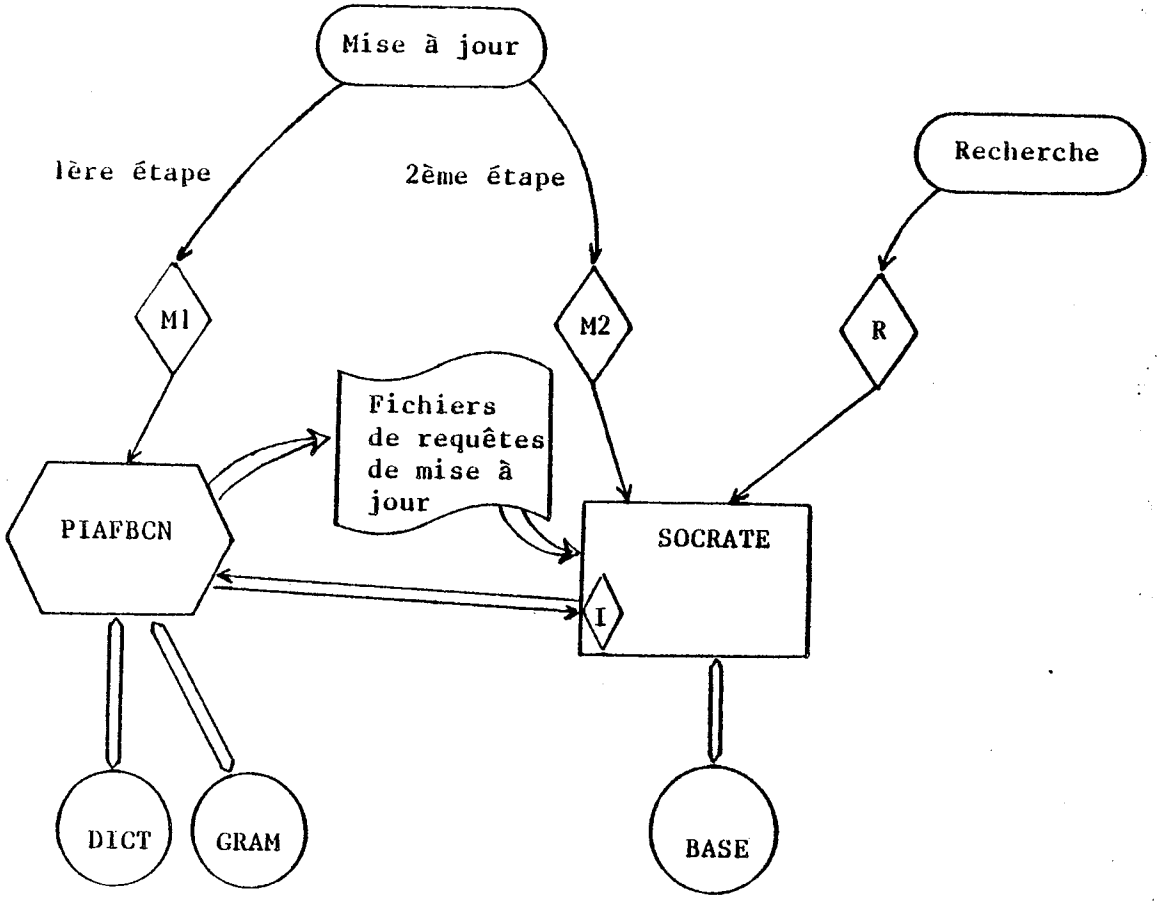
   La mise à jour de la base est une opération se déroulant en deux étapes :

   - La première correspond à l'analyse des descriptions d'origine. Celle-ci est interactive et fournit un fichier de requêtes.
   - La seconde correspond à la mise à jour proprement dite de la base à partir du fichier précédent.

Pour la recherche, un système global a été défini, le fonctionnement des deux logiciels restant indépendant. Les données échangées (SOCRATE fournit un texte de description à PIAFBCN qui en retour lui soumet une requête) transitent par un interface qui résoud aussi les problèmes d'adressage.

Diverses configurations d'entrée-sortie ont été réalisées, selon le mode d'utilisation, tout en conservant l'unicité des différents programmes et paramètres ce qui permet d'assurer la cohérence du système.

Ceci correspond à l'architecture suivante :



M1 et M2 sont les modules d'appel de PIAFBCN et de SOCRATE pour la mise à jour. R et I sont respectivement le module d'appel à SOCRATE et l'interface pour la recherche.

II.3 CONSTITUTION ET EXPLOITATION DE LA BASE

Le SGBD SOCRATE [ABR] a été utilisé pour l'implémentation du logiciel de constitution d'une base de données de nomenclature car il permet une définition rapide et commode de prototypes et la validation de structures d'information complexes. Il offre de plus tous les outils de sécurité nécessaires et des outils statistiques permettant d'analyser les conditions de l'espace mémoire secondaire par les données.

II.3.1 Le système SOCRATE

Le SGBD SOCRATE permet la définition et l'exploitation conversationnelle ou par lots de bases de données de type hiérarchique ou réseau. Les fonctions de la base sont accessibles via des langages de haut niveau qui sont essentiellement :

- Un langage de description. La description des données au niveau logique (schéma conceptuel) et au niveau physique (schéma interne) sont regroupées dans la structure.
- un langage de requêtes qui permet la mise à jour et l'interrogation des données enregistrées dans la base.

Un macro-générateur permet l'expansion de code en langage de requêtes. Il donne la possibilité de :

- Définir un langage de commande simple, spécifique d'une application et ne nécessitant aucune connaissance ni de la syntaxe du langage de requêtes ni de la structure des données de la part de l'utilisateur. Ce langage est constitué d'un ensemble de macro-instructions.
- Définir des programmes permettant de cataloguer une suite de requêtes sous forme compilée.

Le macro-générateur est utilisé pour la définition de toutes les commandes d'exploitation. Par ailleurs, la possibilité de définir et d'appeler des programmes écrits en langage d'assemblage a permis de réaliser l'interface PIAFBCN-SOCRATE.

II.3.2 Présentation du logiciel

II.3.2.1 Définition de la structure

Un objet de la base est défini par les caractéristiques suivantes :

- Un numéro ou référence interne (caractéristique discriminante)
- Un texte contenant des informations relatives à l'objet (origine, dénomination, référence-article ou description...)
- Le code. Cette caractéristique n'a de valeur que si la description de l'objet est entièrement codée. Elle est utilisée pour un accès rapide à l'objet.

- L'alternant. Cette caractéristique permet de connaître les autres sources de fabrication de l'objet (s'il en existe).

Au niveau de la classification des termes d'indexation, toute classe est représentée par son nom et son ensemble de valeurs : deux statuts ont été définis :

- Les classes spécifiques (statut "S"). Elles correspondent aux descripteurs sémantiques définis lors de l'analyse ;
- Les classes génériques (statut "G") correspondant aux types génériques. Ces classes sont disjointes.

Pour restreindre le nombre de valeurs dans le dictionnaire, on autorise l'appartenance d'un terme à la fois à une classe spécifique et une classe générique. Cette redondance est utilisée pour améliorer les résultats de la recherche (cf. II.2.5.1, II.2.5.2).
Toute référence-article est totalement inversée par rapport aux termes constituants.

II.3.2.2 La mise à jour

L'analyse PIAFBCN fournit une requête de mise à jour.
Les principales fonctions de la mise à jour sont :

- Vérification de la non-existence d'un objet ayant le même numéro dans la base ;
- Création d'un nouvel objet ;
- Création éventuelle de nouvelles classes ;
- Création éventuelle de nouveaux termes et création ou mise à jour des fichiers inverses correspondants.

Toute requête de mise à jour est elle-même composée de trois sous-requêtes en général :

- La première sous-requête contient les informations relatives à l'objet dans un format particulier (origine, dénomination, description...) ;
- La seconde correspond au descriptif fourni par l'analyse grammaticale ;

- La troisième, qui existe uniquement si la description n'est pas un code indécomposable, correspond au second descriptif associé. (cf.2.5.1).

II.3.2.3 Interprétation d'une requête d'interrogation

L'aboutissement de l'analyse d'une question est une requête. Celle-ci est en général constituée d'une seule sous-requête, mais elle peut l'être de plusieurs lorsqu'une ambiguïté n'a pas été résolue par l'utilisateur. Celles-ci correspondent alors aux différentes interprétations du contenu de la question. Chaque sous-requête est considérée comme une expression booléenne de "et" implicites, et est interprétée comme une suite d'intersections de fichiers inverses localisés pour chaque couple (classe,terme).

exemple : une sous-requête SR ≡ ((D1,X1)(D2,X2)...(Dn,Xn))
est interprétée {D1X1} ∩ {D2X2} ∩ ... ∩ {DnXn} où {DiXi} est l'ensemble des descriptions contenant le mot Xi avec l'interprétation (Di,Xi), Di dénotant un descripteur.

Si une requête est composée de plusieurs sous-requêtes, celles-ci sont interprétées comme des alternatives possibles.

exemple : l'analyse d'une question produit une requête R composée de trois sous-requêtes SR1, SR2, SR3.
R est alors interprétée {SR1} ∪ {SR2} ∪ {SR3}
où {SR1} est l'ensemble des descriptions localisées par la sous-requête SR1 conformément au principe énoncé ci-dessus.

a. Vecteur d'une sous-requête

Le coût d'une recherche dépend de l'ordre dans lequel sont effectués les intersections de fichiers inverses, car il dépend de leurs tailles. En effet l'algorithme d'intersection est le suivant :

```
Soient F1, F2 les fichiers inverses de départ tel que card(F1) ≤ card(F2)
et F3 = F1 ∩ F2
pour tout objet ∈ F1 faire
   N:= numéro (objet)
   si ∃ (objet ∈ F2 / numéro(objet)=N) alors
         ajouter(objet ∈ F3)
   fin faire
```

La fonction ∃ (objet ∈ F2 / numéro(objet)=N) est évaluée en faisant un accès direct dans F2 par le numéro de réalisation. Donc la taille de F1 est déterminante pour les performances d'évaluation d'une intersection (autant de recherches que d'éléments de F1) ; par contre, la taille de F2 ne l'est pas.

Pour minimiser le temps nécessaire à ces intersections, il faut donc déterminer dans une sous-requête quelles sont les informations les plus sélectives. L'évaluation du pouvoir discriminant de chacune des classes (cf. III.2) permet d'améliorer les performances globales de l'algorithme d'intersection. Cette évaluation conduit à affecter des poids à chacune des classes ; à partir de ces valeurs, les éléments d'une sous-requête sont d'abord réordonnés par ordre décroissant des poids des classes puis l'intersection est évaluée.

Exemple : SR ≡ ((D1,X1)(D2,X2)(D3,X3))
et poids (D2) > poids (D1) > poids (D3)
on construit le vecteur [(D2,X2)(D1,X1)(D3,X3)]
on évalue l'expression {D2X2} ∩ {D1X1} ∩ {D3X3}

b. Cas particulier d'une question totalement codée

Une description totalement codée est plus rarement sujette à des erreurs que les autres. On recherche donc tout d'abord si elle existe telle quelle dans la base ; si ce n'est pas le cas, on fait alors appel à son interprétation.

Une classe virtuelle CODE a été définie ; elle n'est pas enregistrée dans la liste des classes. Lorsqu'elle est reconnue dans une sous-requête, on lui fait correspondre un accès direct dans la base. Si l'objet est retrouvé, le reste de la requête est ignoré, sinon il est interprété comme précédemment. La classe CODE a un poids maximum.

Exemple : SR ≡ ((CODE,X1X2X3)(D1,X1)(D2,X2)(D3,X3))
vecteur associé : [(CODE,X1X2X3)(D2,X2)(D1,X1)(D3,X3)]
Si la recherche directe de l'objet dont la caractéristique "CODE" a la valeur X1X2X3 (cf. II.3.2.1) n'aboutit pas, l'expression {D2X2} ∩ {D1X1} ∩ {D3X3} est alors évaluée.

II.3.2.4 Processus de recherche

a. Affinage d'une question

Si, à l'issue d'une recherche, l'utilisateur juge le résultat trop volumineux, il a alors la possibilité de compléter sa question par une requête d'affinage, dont l'interprétation est identique à une requête de recherche et ceci autant

de fois qu'il le désire. Chaque requête d'affinage sélectionne les objets à partir du résultat acquis à l'étape précédente.

b. Affichage du résultat

Le résultat de toute requête est une liste d'objets dans le cas général. Les informations concernant ces objets sont alors affichées (numéro, description...).

En cas d'échec (liste vide), une justification sous forme de trace d'exécution est fournie à l'utilisateur.

Exemple : Soit la question ((D1,X1)(D2,X2)(D3,X3)(D4,X4))
et le vecteur de requête [(D2,X2)(D1,X1)(D4,X4)(D3,X3)]
si {D2X2} ∩ {D1X1} ∩ {D4X4} = ∅
la trace signale que :

1. {D2X2} ∩ {D1X1} ∩ {D4X4} = ∅
2. l'information (D3,X3) n'a pas été exploitée
3. {D2X2} ∩ {D1X1} = R, R étant donc le résultat immédiatement antérieur au résultat vide.

Il semble donc assez naturel de penser qu'un utilisateur a moins de chances de se tromper sur une information discriminante que sur une information de moindre importance. Le fait que les informations les plus discriminantes sont exploitées en premier, augmente alors la probabilité pour que l'objet cherché soit dans R (résultat antérieur).

II.3.3 Utilisation du logiciel

II.3.3.1 Les commandes relatives à la mise à jour

L'activation de PIAFBCN s'effectue par la commande PIAFBCN. On peut alors sélectionner de manière conversationnelle le support physique des données à traiter et celui de la sortie des résultats (console ou fichier).

- La commande CREA permet d'effectuer l'analyse d'un ensemble d'articles ;
- La commande INTERRO, utile pour la mise au point, permet d'analyser une description frappée au terminal ;
- les commandes DICT et GRAM permettent d'accéder respectivement aux environnements dictionnaire et grammaire.

Les commandes relatives au fonctionnement de l'analyseur sont celles citées en Annexe II.

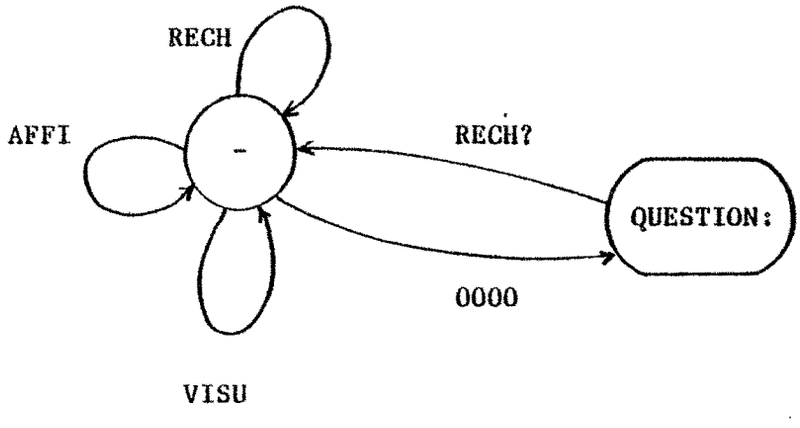
L'enregistrement des articles dans la base, s'effectue par la commande SOCB<nom-fichier> <taille-fichier>
nom-fichier étant le nom du fichier de requêtes fourni par PIAFBCN et taille-fichier le nombre d'enregistrements de ce fichier.
Un compte-rendu de cette opération est donné dans le fichier SORTIE qui contient :

- Les numéros des articles enregistrés ;
- Des messages d'erreurs éventuels.

II.3.3.2 Les commandes relatives à la recherche

L'activation de la recherche s'effectue par la commande RECHBCN. Sous l'état "QUESTION:" de Socrate, la commande RECH ? permet de lancer la recherche d'un objet à parir d'une description.
Le requêtes de recherche sont :

- RECH <description>
- AFFI <description>
- VISU <np>. Si le nombre d'objets sélectionnés est supérieur à 5, seul ce nombre est affiché dans le résultat. La commande VISU permet d'afficher les p objets de la liste constituant le résultat à compter du nième.
- 0000 fin de la recherche et retour à l'état "QUESTION:"



Un ensemble de commandes, que nous ne décrirons pas ici a été défini pour accéder à des informations générales sur la base (utiles pour des études statistiques par exemple). Ces commandes sont accessibles sous l'état "QUESTION:". La liste en est donnée en Annexe V.

II.3.4 Caractéristiques de l'implémentation

Le logiciel de gestion de la base de données a été implémenté sur la version 1.7 de SOCRATE. Ce logiciel est portable dans la mesure où les nouvelles versions de SOCRATE implémentées (ou en cours d'implémentation) sur d'autres sites offrent sensiblement les mêmes outils de base.

La recherche d'un objet dans la base nécessite en moyenne 0,5 s de temps CPU sur l'IBM 370-158.

CHAPITRE III

MESURES DANS LE SYSTEME

III.1 INTRODUCTION

Nous proposons ici diverses mesures destinées à améliorer les performances et la fiabilité de la recherche :

- pour optimiser l'élaboration d'une réponse ; l'ordre dans lequel sont traitées les intersections de fichiers inverses est déterminant de ce point de vue (cf. II.3.2.3). On peut augmenter la rapidité de l'algorithme d'intersection en traitant les caractéristiques par pouvoir discriminant décroissant. Les résultats de l'analyse du pouvoir discriminant sont aussi intéressants pour juger du choix des descripteurs fait lors de l'analyse.
  Cette mesure est dérivée de la théorie de l'information. Elle tient compte à la fois du nombre et de la répartition des termes d'une classe [PIC] [CH2].
- pour optimiser l'indexation d'une description ; cici revient à améliorer la classification des termes. En particulier l'étude de la redondance peut permettre de minimiser le nombre des classes.
- pour présenter l'ensemble des objets constituant le résultat d'une question, lorsque celui-ci est relativement volumineux, par ordre de pertinence décroissante. Celle-ci peut être définie à partir d'une mesure de similarité entre une question et une description.

III.2 EVALUATION DU POUVOIR DISCRIMINANT D'UNE CLASSE

III.2.1 Définitions et notations

X est un ensemble fini d'objets

$X = \{Xi\} \quad 1 \leq i \leq N$

C est un ensemble fini de classes sémantiques

$C = \{Ci\} \quad 1 \leq i \leq P$

T est un ensemble fini de termes

$T = \{ti\} \quad 1 \leq i \leq n$

Une classe est un ensemble fini de termes. On considère que :

- tout terme appartient au moins à une classe. L'appartenance du terme ti à la classe Ck est notée t_k^i.
- toute classe Ck possède une valeur particulière notée nil_k

On note nk = card (Ck) et $T' = T \cup \{nil_i\}$ $1 \leq i \leq P$

La représentation R d'un objet est une application bijective de X dans $Y \subseteq (C \times T')^P$.

III.2.2 Gain d'information

On considère maintenant : X comme une variable aléatoire pouvant prendre les valeurs X1, X2... Xn avec des probabilités égales ($\forall i$ pXi = $\frac{1}{N}$), c'est-à-dire que tous les objets sont considérés a priori comme équiprobables.
Et Ck comme une variable aléatoire pouvant prendre les valeurs t_k^1, t_k^2 ... t_k^{nk} avec respectivement les probabilités p_k^1, p_k^2 ... p_k^{nk} (la valeur nil_k est comprise dans cet ensemble). $\sum_{i=1}^{nk} p_k^i = 1$ (système complet)

Intuitivement, pXi représente la probabilité de rechercher l'objet Xi dans la base et p_k^i la probabilité de rechercher un objet dont la représentation contient le couple (Ck, t_k^i).
Si une représentation ne contient pas de couple de classe Ck, Ck a alors la valeur nil_k.

On veut connaître le gain d'information sur X (ou diminution de l'indétermination sur X) résultant de l'observation de Ck.

Ce gain d'information, noté I (X//Ck) s'exprime en fonction de l'entropie [REN] :

(a) I (X//Ck) = H(X) - H(X/Ck)

H(X) est la quantité moyenne d'information donnée par X (entropie de X)
H(X/Ck) est l'entropie de X connaissant Ck

On a la propriété suivante : I(X//Ck) = I(Ck//X) et d'après (a) :

I(Ck//X) = H(Ck) - H(Ck/X)

Par définition de l'entropie conditionnelle :

$$H(Ck/X) = - \sum_{i=1}^{nk} \sum_{j=1}^{N} p\ (X=Xj)\ p\ (Ck=v_k^i/X=Xk) \quad \log_2\ p\ (Ck=v_k^i/X=Xk)$$

or $\forall j\ p(X=Xj)=\frac{1}{N}$ (répartition uniforme)

et $p(Ck=v_k^i/X=Xk) = 0$ ou 1

donc $H(Ck/X) = 0$ (Intuitivement, celà signifie que lorsqu'on connaît X, on connait la valeur de Ck).

On en déduit que $I(X//Ck) = H(Ck)$

Le gain d'information résultant de l'observation de Ck est égal à l'entropie de Ck.

Le gain est total (l'information est totalement discriminante) si

$$I(X//Ck) = H(X),$$

la répartition étant uniforme $H(X) = \log_2 N$ et puisque $H(Ck) \geq 0$

$$0 \leq I(X//Ck) \leq \log_2 N$$

III.2.3 Pouvoir discriminant d'une classe

On définit le pouvoir discriminant d'une classe par :

$$P(Ck) = \frac{I(X//Ck)}{H(X)}$$

ceci pour ramener P(Ck) à l'intervalle [0,1]

donc $P(Ck) = \frac{H(Ck)}{\log_2 N} \qquad 0 \leq P(Ck) \leq 1$

La formule de Shannon permet d'évaluer la quantité H(Ck) [REN]

$$H(Ck) = - \sum_{i=1}^{nk} p_k^i \log_2 p_k^i, \text{ le système étant complet } (\sum_{i=1}^{nk} p_k^i = 1)$$

On utilise les fréquences de chacune des valeurs pour évaluer a priori les probabilités p_k^i (on considère que le nombre d'observations est grand et qu'il est donc possible d'assimiler les fréquences à des probabilités).

la fréquence f_k^i d'une valeur est égale à $\frac{nk}{N}^i$

où n_k^i est le nombre d'objets ayant la valeur t_k^i dans leur représentation.

$$H(Ck) = \sum_{i=1}^{nk} \frac{n_k^i}{N} \log_2 \frac{N}{n_k^i}$$

d'où (b) $$P(Ck) = 1 - \frac{\sum_{i=1}^{nk} n_k^i \log_2 n_k^i}{N \log_2 N}$$

Cas particuliers

1. P(Ck) est maximal si et seulement si toutes les valeurs t_k^i de Ck sont équiprobables.

$$\forall i \; f_k^i = \frac{1}{nk}$$

La valeur maximale de P(Ck) est $\frac{\log_2 nk}{\log_2 N}$

2. La valeur nulle n'est atteinte que dans le cas limite où la classe ne possède qu'une seule valeur (de probabilité 1).

III.2.4 Application au calcul des poids associés aux classes de la base

Dans la représentation effective d'un objet n'intervient qu'un nombre restreint de classes (en moyenne 6).

L'évaluation du pouvoir discriminant donné par (b) ne peut pas être considérée comme utilisable sur l'ensemble des objets de la base globalement. En effet, si un objet est représenté sur $(C \times T')^P$ (cf. III.2.1 où P est le nombre total de classes), la majorité des classes prennent la valeur que nous avons noté "nil". Cela signifie que la probabilité des valeurs "nil_k" des classes Ck est très forte par rapport aux autres. Ces distributions déséquilibrées, rendent la comparaison des pouvoirs discriminants des classes peu significative.

L'évaluation doit être relative à un ensemble particulier d'objets.On partitionne la base en sous-ensembles d'objets de la même famille (où la représentation fait intervenir les mêmes classes). On évalue le pouvoir discriminant de Ck dans toute famille (correspondant à un sous-langage) où Ck intervient. La probabilité que Ck prenne la valeur "nil_k" est alors faible. Le poids de la classe Ck est ensuite calculé en faisant la moyenne des P(Ck) pour tous les sous-ensembles concernés.

Actuellement dans le prototype, les poids sont attribués empiriquement, la classification automatique des Ck n'ayant pas été réalisée.

L'évaluation du pouvoir discriminant que nous avons envisagé pour les classes sémantiques peut être appliquée,de façon identique, aux classes génériques. Les poids obtenus seront alors faibles car la distribution des valeurs est déséquilibrée (valeurs à très forte ou très faible probabilité).

Pour toutes les classes, cette évaluation doit avoir un caractère dynamique du fait de l'évolution de la base.

III.3 EFFICACITE D'UN CHEMIN D'ACCES

III.3.1. Introduction

L'évaluation du pouvoir discriminant pour un ensemble de classes est coûteuse car il faut évaluer un grand nombre de probabilités conditionnelles (il y a un très grand nombre de combinaisons du fait de la taille des différentes classes, et on ne peut pas considérer que ces classes sont indépendantes).

Nous avons donc envisagé une autre approche qui consiste à évaluer a postériori l'efficacité d'un chemin d'accès en analysant les résultats fournis par le système.

Cette étude offre un double intérêt :

- Aspect qualitatif : déterminer quels sont les chemins effectivement utilisés. On peut alors envisager de calculer les entropies conditionnelles pour les associations les plus fréquentes.
- Aspect quantitatif : optimiser la succession d'étapes et supprimer celles qui sont inutiles (redondance).

III.3.2. Evaluation de l'efficacité d'un chemin

On appelle chemin la liste ordonnée (par ordre décroissant des valeurs des poids) des classes intervenant dans une question.

Pour évaluer l'efficacité d'un chemin, on observe l'évolution du nombre

d'objets sélectionnés à chaque étape (une étape correspondant à une opération d'intersection).

Soit $\mathcal{C}$ = C1C2...Cn un chemin relatif à une question Q.

no=N le nombre total d'objets enregistré dans la base.

n_i le nombre d'objets sélectionnés après la prise en compte de la valeur de Ci.

On calcule l'amélioration du taux de réponses k_i à l'étape Ci par

$$k_i = \frac{n_{i-1} - n_i}{n_{i-1}} \quad 0<i\leq n, \quad 0\leq k_i \leq 1$$

La valeur de ce coefficient pour I questions comportant cette étape est calculée en prenant la moyenne de tous les coefficients k_i^j, $1\leq j\leq I$

$$1\leq i\leq n \quad K_i = \frac{\sum_{j=1}^{I} k_i^j}{I} \quad 0\leq K_i \leq 1 \qquad (1)$$

Cette moyenne reflète l'efficacité qu'un utilisateur peut attendre de cette étape.

Une seconde méthode peut être utilisée pour évaluer ce coefficient.

$$1\leq i\leq n \quad K_i' = \frac{\sum_{j=1}^{I} n_{i-1}^j - \sum_{j=1}^{I} n_i^j}{\sum_{j=1}^{I} n_{i-1}^j} \quad 0\leq K_i' \leq 1 \qquad (2)$$

Cette expression peut être interprétée de la façon suivante : à partir des I questions originales on considère une unique question hypothétique construite de telle sorte que le nombre d'objets sélectionnés à l'étape i soit la somme des nombres d'objets sélectionnés par les I questions originales.

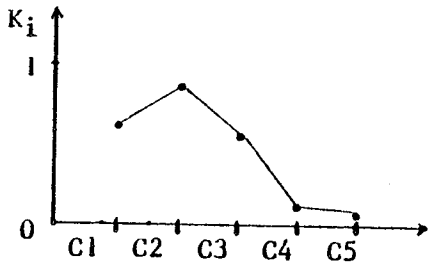
Il est probable que ces deux méthodes donnent des résultats proches pour I assez grand mais les expériences de recherche dans la base n'ont pas été suffisammentvariées et nombreuses pour vérifier cette assertion.
L'efficacité d'un chemin complet (succession d'étapes) peut être évaluée en prenant la moyenne des coefficients K_i (ou K_i') :

$$EF(\mathcal{C}) = \frac{\sum_{i=1}^{n} K_i}{n} \quad 0\leq EF(\mathcal{C})\leq 1$$

Interprétation

- S'il existe i, tel que la valeur de K_i soit voisine de 1, il en ressort que l'ensemble (C1,C2...Ci) est fortement discriminant.
  Si i≠n, les informations apportées par les valeurs de Ci+1...Cn sont redondantes pour la discrimination.
- Si la valeur de K_i est voisine de 0, la classe Ci est redondante avec l'une des classes qui la précède dans le chemin (pas d'amélioration du taux de réponses).

Exemple : Considérons une question faisant intervenir les classes C1,C2...C5 où P(C1)≥P(C2)≥,...≥P(C5) et les coefficients K_i du chemin d'accès donnés par :



Il est vraisemblable que, dans cet exemple, les classes C4 et C5 sont redondantes.

III.3.3. Applications

Un premier point est d'optimiser l'indexation d'une référence-article (limitation du nombre de couples produits à l'issue d'une analyse, choix des classes). Pour celà, l'étude de la redondance entre différentes classes permet d'éliminer ou de restructurer certaines classes, ceci impliquant une modification de l'analyseur.

Cette opération doit être effectuée en tenant compte du pouvoir discriminant propre à chaque classe et de leur redondance mutuelle.

Mais l'élimination de la redondance ne doit pas s'effectuer au détriment des possibilités de recherche. En effet si on supprime une classe redondante, on gagne de la place mais on se prive d'une possibilité d'accès. Il en est de même si l'on concatène plusieurs classes (concaténation des termes respectifs)

car la conséquence d'une erreur est alors plus importante.

Il y a donc un conflit entre ces deux points de vue et une "bonne" optimisation n'est pas chose facile à réaliser.

Les résultats obtenus peuvent présenter également un intérêt pour l'utilisateur. Lors de la formulation d'une question celui-ci a tendance à être économe (choix des termes les plus discriminants et abandon des termes superflus) tout en restant suffisament précis. L'évaluation de l'efficacité des chemins, permet de caractériser des ensembles minima de termes pour obtenir un résultat satis faisant.

III.4 EVALUATION D'UNE SIMILITUDE ENTRE UNE QUESTION ET UN OBJET

III.4.1 Introduction

Dans le résultat d'une question, les objets sont actuellement présentés à l'utilisateur dans un ordre quelconque. Un ordre peut être établi en évaluant une similarité entre la question et les différents objets constituant le résultat, et en présentant les objets par valeur de similarité décroissante. Cette mesure permet d'évaluer la pertinence des réponses ; on rejoint donc ici certaines notions de la documentation automaitque [SALT1]. Cette approche n'offre un intérêt que dans le cas où la taille du fichier résultat est relativement grande.

III.4.2 Mesure de similarité

a. Première approche

Soit une question Q et un objet identifié par sa description X.
Soit TQ (resp. TX) l'ensemble des termes de la représentation de Q (resp. X)
On peut mesurer la similarité entre Q et X en évaluant :

$$s(Q,X) = \frac{card(TQ \cap TX)}{card(TQ \cup TX)}$$

$s(Q,X)$ a les propriétés suivantes :

- $0 \leq s(Q,X) \leq 1$
- symétrie

$s(Q,X)=1$ ssi Q et X sont identiques à des permutations de mots ou des synonymies près.

L'avantage ce cette mesure est sa simplicité de mise en oeuvre.
Mais, puisque ces critères sont purement linguistiques, elle ne rend pas compte de l'importance relative des différents termes. Par exemple un terme omis peut être très important ou peu important.

b. Affinement de la mesure

On considère qu'à chaque terme est associé un poids reflétant son importance. Ce poids est la valeur du pouvoir discriminant de la classe à laquelle il appartient.

On affine la mesure précédente en prenant en compte ces poids

$$s'(Q,X) = \frac{\sum_{t \in Q \cap X} p(t)}{\sum_{t \in Q \cup X} p(t)}$$

où $p(t)$ est le poids du terme t.

Les propriétés de s' sont identiques à celles de s. L'évaluation de la valeur de $s'(Q,X)$ reste très simple.

CHAPITRE IV

CONCLUSION

Sur le plan pratique, l'objectif que nous nous étions fixé était de pouvoir réaliser rapidement un prototype pour mettre en évidence la faisabilité et les caractéristiques d'un logiciel de traitement des références, et d'autre part de tester ce prototype auprès d'utilisateurs réels. Pour ce faire, une base relativement importante et représentative de la variété des langages a été constituée.

Notre travail a été plus orienté vers la recherche d'informations que le contrôle de saisie ; le système défini peut être utilisé pour toute investigation particulière, autre que la recherche d'un objet unique.

Des améliorations sont souhaitables aussi bien pour renforcer les moyens de contrôle à la saisie et élargir les possibilités d'exploitation de la base que pour simplifier le processus de constitution de celle-ci de façon à le rendre plus accessible aux utilisateurs. Il reste d'autre part à intégrer le logiciel d'analyse au logiciel de constitution et d'exploitation de la base de données afin d'améliorer les performances globales du système et d'en augmenter la fiabilité. Ces différents points sont développés dans la suite de ce chapitre.

Enfin, compte-tenu des impératifs de temps, certains aspects plus théoriques du problème n'ont pas pu être suffisamment développés et validés. La validation des mesures proposées nécessite en effet un effort d'implémentation supplémentaire ainsi qu'une période relativement longue d'utilisation de la base dans un contexte réel d'exploitation.

IV.1 EVALUATION DES LOGICIELS ET PERSPECTIVES

IV.1.1 Simplification de la définition des grammaires

Les résultats fournis par PIAFBCN se sont avérés satisfaisants pour le sous-ensemble "descriptif" du langage tant par leur qualité par rapport aux objectifs définis que par les performances de l'exploitation. Il est à noter cependant l'absence, dans la grammaire, de fonctions de calcul qui permettraient la normalisation automatique des champs numériques selon des unités de mesure standard.

Ces fonctions existent dans les développements de PIAF en PASCAL.

Toutefois la définition des règles et des modèles linguistiques est relativement complexe et le formalisme utilisé est contraignant. Si ce n'est pas un problème réel dans le cas d'une application pour laquelle la grammaire est plus ou moins figée, il le devient lorsque les mises à jour sont des opérations fréquentes, effectuées par des utilisateurs non informaticiens.

Une simplification de la définition des automates transducteurs est donc souhaitable. L'idéal serait qu'elle soit totalement transparente à l'utilisateur. Une approche consisterait à rendre le métalangage de définition des descriptions plus proche de la conception que l'utilisateur a de celles-ci (par exemple par visualisation sous forme graphique). Il serait également intéressant de voir si on peut envisager une solution par inférence grammaticale [PAO].

IV.1.2 Renforcement des moyens de contrôle

Nous avons souligné dans cette étude, l'importance du contrôle, à la saisie notamment. Un des aspects essentiels est que toute requête soit considérée dans un contexte conversationnel afin de permettre une correction immédiate des informations. Des améliorations sensibles pourraient être apportées à ce niveau pour offrir à l'utilisateur une véritable aide dans la formulation de ses requêtes (en particulier, visualisation dans un format "simple" de la grammaire correspondant à un type de requête, lorsque c'est possible).

D'autre part, une limite importante des contrôles provient du fait que l'outil utilisé a été conçu pour analyser des chaînes de caractères. Or une description contient fréquemment des données numériques ne pouvant pas raïsonnablement être prises en compte de la même façon que les chaînes. Il faut donc pouvoir définir des ensembles de valeurs soit formellement (par un type) soit par énumération. Plus généralement, il faudrait décider de manière prédicative de la catégorie d'un mot.

IV.1.3 Traduction des codes

Actuellement, l'utilisation des informations contenues dans la base est limitée par l'absence d'opérations de traduction des expressions codées. Une telle

traduction offre un intérêt de plusieurs points de vue :

- L'accès à la traduction d'un code peut être considéré comme un moyen de contrôle. Cela permet en effet de rectifier, de façon interactive, certaines erreurs dues à la méconnaissance de la codification des caractéristiques d'un objet.
- Tout utilisateur peut prendre connaissance de l'information contenue dans un code et ceci automatiquement.
- Celà rend possible la recherche d'un objet à partir de ses caractéristiques sans connaître la façon dont celles-ci sont codées dans la référence-article de l'objet.

Le problème réside dans le grand nombre des systèmes de codification existants sur un vocabualaire de code très résistant. La structure de la base doit donc être partiellement modifiée et il faudrait probablement envisager la constitution d'un thésaurus pour représenter les relations entre les mots exprimant les caractéristiques des objets. Cette orientation, de type réellement documentaire, nous semble offrir une recherche intéressante.

IV.2 INTEGRATION DES LOGICIELS

Les deux produits ont été réalisés séparément et leur fonctionnement est indépendant. Une intégration totale semble difficile à réaliser pour des raisons techniques sur le matériel que nous avons utilisé, notamment parce qu'il n'existe pas de langage de programmation de haut niveau pouvant efficacement prendre en compte des problèmes aussi éloignés que l'analyse de chaînes de caractères et la gestion d'une base de données ; il est toutefois intéressant d'effectuer une intégration partielle des deux logiciels.

Actuellement beaucoup d'informations sont dupliquées, en particulier dans les dictionnaires relatifs aux deux logiciels.
Outre le gain de place qui résulterait de l'unicité du dictionnaire, ceci renforcerait la cohérence de ces informations.

On peut escompter d'une intégration partielle de meilleures performances globales (dues à une meilleure répartition des différentes fonctions du système)

et des améliorations qualitatives (fiabilité plus grande due en particulier à la suppression des redondances).

IV.3 MESURES DANS LE SYSTEME

Nous sommes conscients que les méthodes d'évaluation quantitatives et qualitatives proposées ne sont qu'une première approche. Elles ont la mérite d'être simples à mettre en oeuvre. L'analyse du pouvoir discriminant offre un intérêt dans tous les types d'applications où aucun critère discriminant ne peut être défini a priori [CH2]. L'exploitation des résultats ainsi obtenus pour optimiser le découpage des références nous semble être une perspective intéressante ; la limitation de la redondance doit être étudiée en tenant compte néanmoins des problèmes liés à la recherche.

IV.4 REFLEXION SUR LE CODAGE

Au terme de cette étude, nous conclurons également par une réflexion sur le codage. L'objectif du codage est de représenter de façon plus condensée un ensemble d'informations, de façon à pouvoir les communiquer plus rapidement et de manière normalisée (non ambigüe). Les contraintes que le codage représente, se traduisent par un effort accru d'apprentissage et d'attention de la part des personnes manipulant ces informations.

C'est une représentation qui est donc utile mais aussi source d'erreurs car il n'y a plus d'expression naturelle de l'information.

Excepté le cas de codes très simples (type numéro de sécurité sociale), il semble que le codage, contrairement à ce qu'on pourrait penser au premier abord, soit en définitive un obstacle important à l'informatisation des traitements.

ANNEXE I

RAPPELS SUR LE SYSTÈME PIAF

Le système PIAF (Programme Interactif d'Analyse du Français) a été conçu et réalisé au Laboratoire IMAG pour le traitement assisté des langues naturelles [COU].

Il est composé de deux modules principaux : un transducteur général d'états finis (TGEF) conçu pour l'analyse morphologique et un système d'analyse syntaxique. Le TGEF étant le seul utilisé dans cette étude, nous ne décrirons pas l'analyseur syntaxique.

1. Caractéristiques principales du TGEF

Une originalité de ce système est de permettre la définition de tout automate transducteur d'états finis à partir de trois types de paramètres :

- un dictionnaire, un ensemble de modèles et un ensemble de règles de transduction.

C'est un ensemble interactif qui permet un dialogue entre l'utilisateur et la machine. Un éditeur lexicographique [CDG] permet à l'utilisateur :

- la création et la modification en mode conversationnel des paramètres linguistiques ;
- la détection et la correction manuelle des erreurs ;
- le traitement assisté des ambiguïtés ou des insuffisances (choix de polysémie, ajout d'informations).

La possibilité de définir dans le dictionnaire des familles de mots synonymes en indiquant l'élément préférentiel, confère au transducteur un rôle de normalisateur de formes.

La souplesse et la conception très générale du TGEF ont conduit à de nombreuses applications :

- réalisation d'un système documentaire, utilisant comme outil linguistique PIAF pour l'extraction et la normalisation des mots clefs [GRA] ;
- transducteur phonétique [CH3] ;
- traduction de programmes d'un langage dans un autre langage de la même famille [CHA] ;
- traduction d'un texte en Braille abrégé [MAT].

2. Présentation du TGEF

Les objectifs de l'analyse morphologique sont les suivants :

- segmenter une phrase pour obtenir des mots ou des groupes de mots ;
- déterminer pour chaque chaîne fournie par la segmentation un certain nombre de renseignements linguistiques.

Un mot peut être lui-même composé de plusieurs segments. Ces segments sont répertoriés dans un dictionnaire, le rôle de la grammaire étant de vérifier la cohérence de leur concaténation pour constituer le segment cherché.

. Description des paramètres linguistiques

a. Le dictionnaire

Un élément du dictionnaire est constitué d'une chaîne de caractères (segment élémentaire candidat au découpage des mots), de références à d'autres éléments du dictionnaire, de références à des modèles de comportement (ex. modèle morphologique) et éventuellement d'indicateurs particuliers relatifs à l'application des règles de la grammaire.
Ceci rend en particulier possible la définition, au niveau du dictionnaire, de classes d'équivalence pour une relation donnée (ex. synonymie).

b. La grammaire

La grammaire est une grammaire à validations et saturations équivalente à une grammaire d'états finis [COU]. Son rôle est de :

- contrôler la concaténation des différents constituants d'un mot ;
- effectuer la transduction des informations linguistiques désirées.

- les modèles

Les modèles réalisent l'interface entre le dictionnaire et la grammaire. Afin de ne pas encombrer inutilement le dictionnaire, chacun de ses éléments fait référence à un modèle, celui-ci étant le représentant d'une classe d'équivalence dont tous les éléments ont un même comportement linguistique.
Chaque modèle est en liaison avec une ou plusieurs règles de la grammaire.

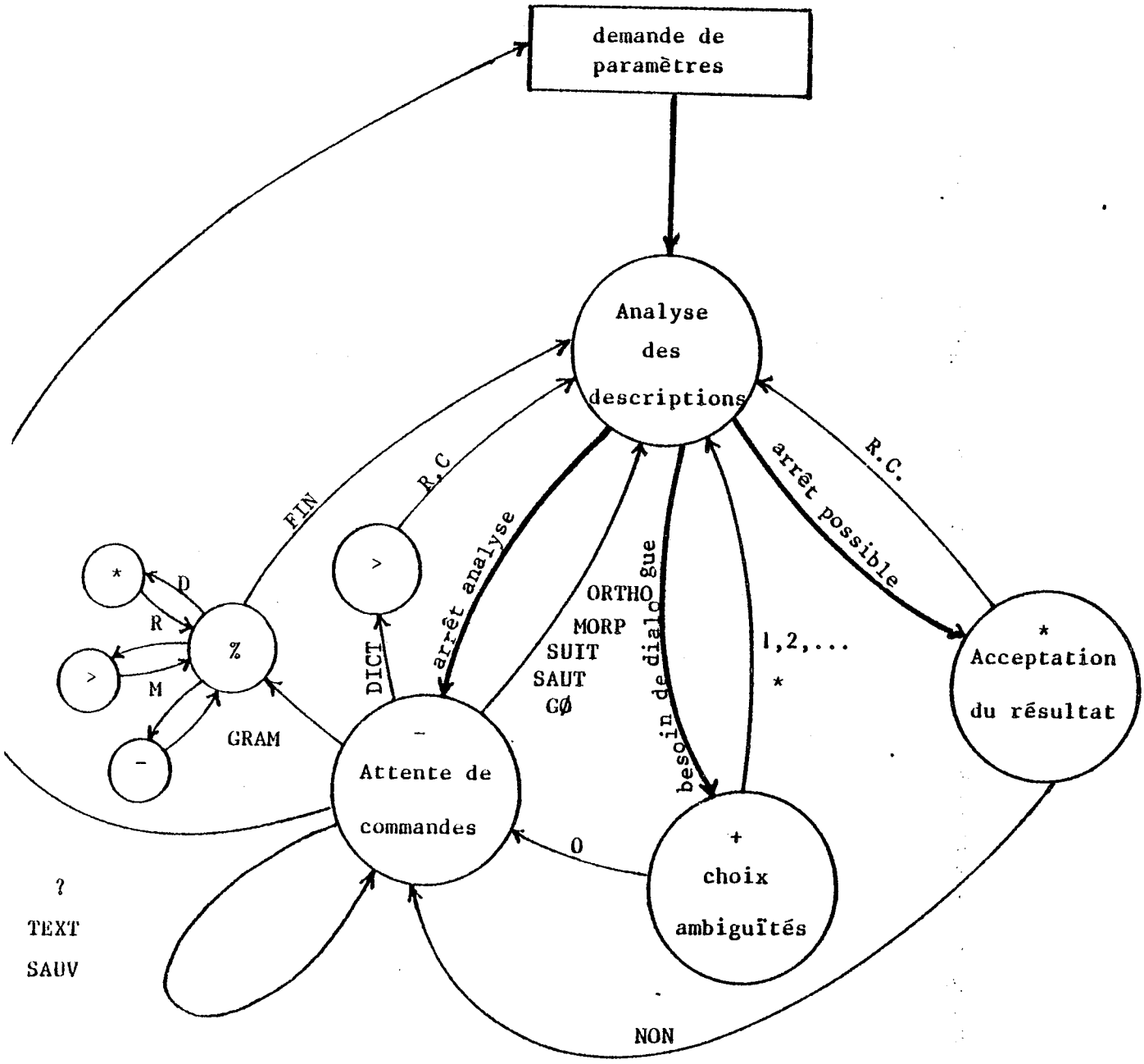
- les règles

L'application d'une règle conduit à calculer l'état courant de l'automate et à transducter des informations à partir des propriétés de l'état précédent et de celles de l'état courant.

ANNEXE II

UTILISATION DE PIAFBCN

Diagramme de fonctionnement de PIAFBCN



Les différents états sont matérialisés par un caractère spécial qui signale à l'utilisateur l'attente d'une réponse.

Dans ce qui suit, les commandes ou les états spécifiques à la saisie (c'est-à-dire non accesibles à la recherche) ou spécifiques à la recherche sont marqués par les symboles (S) et (R).

Etat "-" : attente de commandes

On se trouve dans cet état lorsque l'analyse n'a pu être réalisée ou lorsque le résultat produit est jugé non satisfaisant :

- chaîne ne contenant pas de numériques non analysée au niveau T1. Une telle chaîne, en effet, n'est probablement pas un code. C'est le seul cas où l'analyse est interrompue prématurément avant la production d'un résultat ;
- choix d'homographes non satisfaisant (S) ;
- désaccord avec le résultat produit (S). En d'autres termes,, cela signifie qu'une chaîne a été analysée au niveau T3 "à tord".

Dans cet état, l'utilisateur dispose des commandes suivantes :

- ? Cette commande (disponible aussi dans tous les autres états) permet d'obtenir la liste de toutes les commandes possibles dans l'état où l'on se trouve.
- TEXT Impression du texte complet de la description en cours d'analyse.
- ORTHO Cette commande permet de modifier le texte de la description en cours d'analyse en faisant passer l'utilisateur dans l'environnement "modification du texte d'entrée".
- SAUV (S) Cette commande, qui est une sécurité, permet de sauvegarder les fichiers du dictionnaire et de la grammaire pour conserver les modifications faites au cours de l'analyse.
- SUIT Abandon de l'analyse de la description en cours.
- SAUT Saut de l'analyse du mot en cours et passage à l'analyse des mots suivants.
- MORP Cette commande a pour objet de relancer l'analyse au début de la description.

- GO Cette commande permet de forcer l'analyse aux niveaux suivants d'une chaîne ne contenant pas de numériques.
- $FIN (S) Sortie de l'environnement analyse.
- DICT Accès à l'environnement dictionnaire. L'utilisateur peut alors consulter le dictionnaire ou le mettre à jour (S) (extension, suppression ou remplacement).
- GRAM (S) Accès à l'environnement grammaire. L'utilisateur peut alors interroger ou mettre à jour les déclarations, les règles ou les modèles.

Etat "+" : choix ambiguïtés (besoin de dialogue)

On se trouve dans cet état, lorsqu'un cas d'ambiguïté se présente.
Un choix est proposé à l'utilisateur sous la forme :

1. Classe1 mot 1
2. Classe2 mot 2

.
.
.

Celui-ci peut alors :

- choisir l'interprétation correcte en répondant par le numéro de la ligne correspondant.
- accéder à l'état "attente de commandes" en répondant "0" (S).
- ne pas être en mesure de choisir (R). La réponse "*" permet de prendre en compte toutes les propositions (production de plusieurs résultats).

Etat "*" : acceptation du résultat produit (S)

On se trouve dans cet état lorsque le niveau T3 a été atteint au moins une fois au cours de l'analyse. Un accord avec le descriptif proposé est alors demandé à l'utilisateur. La réponse "non" lui permet d'accéder à l'état "attente de commandes".

ANNEXE III

EXEMPLES D'UTILISATION

DU LOGICIEL PIAFBCN

```
piafbcn
EXECUTION BEGINS...
ACTIVER PROGRAMME ? (GRAN,DICT,INTERRO,CREA)
$
interro                                  /entrée et sortie terminal/
>
tube laiton uz36 dia.3,2x0,25 ecroui 1/2 dur

SDEN TUBE  SMAT LAITON  SNOA CU-ZN36  SDIM DIA.3,2X0,25  SELA ECROUI  SELA 1/2DUR
>                                        /normalisation de "uz36" et "1/2 dur"/

16t 200pf +-20pct 1kv

STYP 16T  SCAP 200PF  SPRE +-20%  STEN 1KV
>
vis f/90 m4x8 u acier qual.3.6 cd8b

1   SFIN U                               /choix polysèmes/
2   SFOR U
CHOIX  AMBIGUITE  DE GRAM0 ? (NO LIGNE OU 0)
*
1
SDEN VIS  SFOR F/90  SDIM M4X8  SFIN U  SMAT ACIER  SMAT QUAL.3.6  STRA CD8B
>
rnf100-1 1/8 long.20 repere s noir

1   STYP RNF100-1                        /ambiguïté de découpage/
2   STYP RNF100
CHOIX  AMBIGUITE  DE GRAM0 ? (NO LIGNE OU 0)
*
1
STYP RNF100-1  SDIM 1/8  SDIM LG.20  SREP REPERE-S  SCOL NOIR
>
122169-40-22bc1                          /code interprétable/

SNOR L22169 SDIM 40 SDIM 22 SMAT BCL
```

```
>
1n5236a                                  /code non interprétable/

G1N  1 G1A  N G4N  5236 G1A  A
ACCORD ? (O/N)
*                                        /acceptation du résultat produit/

>
vis h si 8x35/20 acier xc38 cd8b

SDEN VIS  SFOR H  SDIN SI8X35/20  SMAT ACIER  G2A  XC G2N  38 STRA CD8B
ACCORD ? (O/N)
*
n
-
dict                                     /accès au dictionnaire/
>
?/xc38/                                  /interrogation du mot "xc38"/

/XC35F/BOIS//XC42/BOIS/
/XC35/BOIS/
/X/X/X /
/X/CROI/
/X(S)/LET/Y/
/X/1SEP/
>
/xc38/bois/                              /introduction du mot "xc38" lié au modèle BOIS/
>

SDEN VIS  SFOR H  SDIN SI8X35/20  SMAT ACIER  SNUA XC38  STRA CD8B
>
$fin
```

ANNEXE IV

EXEMPLES DE RECHERCHE DANS LA BASE

```
rechbcn
*** RECH. D' ANTERIO. ***
........ > RECH ?
COMMANDE : RECH MOT-1 MOT-2 .... MOT-N
........ : AFFI MOT-1 MOT-2 .... MOT-P
........ : VISU D L ( DEBUT LONGUEUR )
........ : 0000 ( FIN DE REQUETES )
........ : HELP
*** INTERRO DE LA BASE ***
........ > POUR AVOIR LA LISTE DES COMMANDE : LCOMM ?
*SOCRATE A VOTRE SERVICE*
$
ACC-NUMBER:
NAME:
BASE:
MODE:(NVC,UPD,INT)
MOT-DE-PASSE:
SOUS-STRUCTURE :
OPLS :
LOGIN BASE BASE1
PRET  LE 16.10.81  A 09.14.13
$
QUESTION:
rech ?
*-
rech vis h si 8x35/20
                                        /description incomplète/

RECH SDEN VIS  SFOR H  SDIM SI8X35/20
* SOUS REQUETE: NOMBRE DE REPONSES:   1
*** RESULTAT FINAL; 1      REFERENCES TROUVEES **

  NO THOMSON:  99018792
  AFNOR VIS H SI 8X35/20 ACIER XC38 CD8B
-
rech cf5 0,022mf +-10pct 1000v gaine pvc    /description surabondante (redondance)/

RECH STYP CF5  SCAP 0,022MF  SPRE +-10%  STEN 1000V  SDEN GAINE  SMAT PVC
* SOUS REQUETE: !! REPONSE VIDE !! TRACE D EXECUTION:
    DESCRIPTEUR: STEN VALEUR: 1000V
    DESCRIPTEUR: SCAP VALEUR: 0,022MF
    DESCRIPTEUR: STYP VALEUR: CF5
    DESCRIPTEUR: SPRE VALEUR: +-10%
    DESCRIPTEUR: SMAT VALEUR: PVC
  !! DESCRIPTEUR FATAL !! REPONSES ANTERIEURES: 1
    DESCRIPTEUR: SDEN VALEUR: GAINE          NON UTILISE
*** RESULTAT FINAL; 1      REFERENCES TROUVEES **

  NO THOMSON:  99000212
  RTC CF-5 0,022MF +-10PCT 1000V...
```

```
-
rech vis h xc38

RECH SDEN VIS  SFOR H  SNUA XC38
* SOUS REQUETE: NOMBRE DE REPONSES:   61
*** RESULTAT FINAL; 61     REFERENCES TROUVEES **
-
affi cd8b                          /précision de la description initiale/

AFFI STRA CD8B
* SOUS REQUETE: NOMBRE DE REPONSES:   59
*** RESULTAT FINAL; 59     REFERENCES TROUVEES **
-
visu 1 4                           /visualisation d'une partie du résultat/

  NO THOMSON:  99065648
 AFNOR VIS H M5X28/28 ACIER XC38 CD8B

  NO THOMSON:  99062425
 AFNOR VIS H M6X35/35 ACIER XC38 R 80KG/MM2 CD8B

  NO THOMSON:  99052631
 AFNOR VIS H M14X30/30 AC.XC38 TRAITE R80KG/MM2 CD8B

  NO THOMSON:  99062755
 AFNOR VIS H M16X60 ACIER XC38 R 80KG/MM2 CD8B

-
rech t

1   SFIN T
2   SFOR T
CHOIX  AMBIGUITE  DE GRAM0 ? (NO LIGNE OU *)
*
*                                  /prise en compte des deux interprétations/
RECH SFIN T
SUITSFOR T
* SOUS REQUETE: NOMBRE DE REPONSES:   2
*** RESULTAT FINAL; 2      REFERENCES TROUVEES **

  NO THOMSON:  99055691          /résultat correspondant à la 1ère interprétation/
  AFNOR VIS F/90 M8X35 T CUPRO-NICKEL-ZINC

  NO THOMSON:  99109743
  BNA RIVET TUB.2PIECES T/PLAT.R93506 3X9 T E24 BR.MILIT
* SOUS REQUETE: NOMBRE DE REPONSES:   1   /résultat correspondant à la 2ème inter-
*** RESULTAT FINAL; 3      REFERENCES TROUVEES **    prétation/

  NO THOMSON:  99055691                  /résultat final/
  AFNOR VIS F/90 M8X35 T CUPRO-NICKEL-ZINC

  NO THOMSON:  99035482
  AFNOR T CU-ZN36 10X10X1,5 ECROUI 1/4 DUR

  NO THOMSON:  99109743
  BNA RIVET TUB.2PIECES T/PLAT.R93506 3X9 T E24 BR.MILIT
```

```
-
rech raccord union 12.10/12.17 cd8nb co

GRAM0: A PARTIR DE :UNION
-
dict                                    /interrogation du dictionnaire/
>
?/union/
/UNION DOUBLE(*)(S)/R/UNION-DOUBLE//UNION SIMPLE(*)(S)/R/UNION-SIMPLE/
/U(S)/LET/V/
/U/U/
/U/R/PROFILE U/
>

GRAM0: A PARTIR DE :UNION
-
ortho                                   /modification de la description initiale d'après
>                                        les renseignements du dictionnaire/
/union/union simple/

RECH SDEN RACCORD  SFOR UNION-SIMPLE  G2N  12 G2N  10 G2N  12 G2N  17 STRA CD8NB  SFCO CO
* SOUS REQUETE: NOMBRE DE REPONSES:   1
*** RESULTAT FINAL; 1      REFERENCES TROUVEES **

  NO THOMSON:  91300602
  ERMETO RACCORD UNION SIMPLE 12.10/12.17 CO CD8NB N


-
rech fil aug21 paroi mince              /description incomplète avec un terme erroné
                                         (fil' au lieu de 'gaine')/
RECH SDEN FIL  SDIM AUG21  SFCO PAROI-MINCE
* SOUS REQUETE: !! REPONSE VIDE !! TRACE D EXECUTION:
    DESCRIPTEUR: SDIM VALEUR: AUG21
    DESCRIPTEUR: SFCO VALEUR: PAROI-MINCE
    DESCRIPTEUR: SDEN VALEUR: FIL
  !! DESCRIPTEUR FATAL !! REPONSES ANTERIEURES: 1
*** RESULTAT FINAL; 1      REFERENCES TROUVEES **

  NO THOMSON:  91257280
  FILOTEX CD360 TFE PAROI MINCE AUG21 TRANSLUCIDE
-
0000
QUESTION:
non
$
logout
LOGOUT BASE: BASE1
LOGOUT  LE 16.10.81  A 09.32.58
R; T=7.83/10.25 09:33:00
```

ANNEXE V

LISTE DE QUELQUES COMMANDES

D'INTERROGATION DE LA BASE

1) Recherche d'un produit par son numéro interne

- PROD <numéro> ? affiche les renseignements concernant le produit ayant le numéro interne donné.

2) Evaluation de la taille de la base

- NBPROD ? affiche le nombre de produits de la base
- NBVAL ? affiche le nombre de valeurs contenues dans la base
- NBDESCR ? affiche le nombre de descripteurs utilisés.

3) Renseignements concernant les descripteurs

- NOMDESCR ? affiche le nom de tous les descripteurs
- IDESCR <nom-du-descripteur> ? affiche les différentes valeurs d'un descripteur, le nombre de produits référrés par chacun, le nombre total de produits concernés
- DESCRI <nom-du-descripteur> ? affiche les renseignements concernant un descripteur donné
- DESCRNO <numéro> ? affiche les renseignements concernant un descripteur de numéro donné (numéro du descripteur dans la base).

4) Renseignements concernant les valeurs

- VALUE <chaîne> ? affiche les renseignements concernant une valeur donnée
- VALNO <numéro> ? affiche les renseignements concernant une valeur de numéro donné (numéro de la valeur dans la base).

5) Renseignements concernant les produits

- NOTH <nom-du-descripteur> ? affiche le numéro interne de tous les produits dont le descriptif contient le descripteur donné
- LRVAL <descripteur valeur> ? affiche le numéro interne de tous les produits dont le descriptif contient le couple (descripteur,valeur) donné
- NBREF <nom-du-descripteur> ? affiche le nombre de produits dont le descriptif contient le descripteur donné.

6) Renseignements concernant les poids

- IPOIDS <nom-du-descripteur> ? affiche le poids associé à un descripteur
- PDSDESCR <nom-du-descripteur poids> ? affectation d'un poids à un descripteur
- IPOIDS * ? affiche le poids de tous les descripteurs.

ANNEXE VI

EXEMPLES D'ARTICLES

NOART	LC	NOMOD	LC	REFERENCE ARTICLE
99030490	M	14196	K	VIS FB/90 SI 5X12/12 ACIER 30NC11 CD8B
99070061	P	14196	K	VIS FB/90 M5X12/12 ACIER XC38 R42/65KG/MM2 CD8B
99076777	C	14196	K	VIS FB/90 SI 3X6 ACIER XC38 CD8B
99014920	E	14202	C	VIS C SI 4X8 U ACIER QUALITE 12.9 CD8B
99014922	X	14202	C	VIS C SI 4X12 U ACIER QUALITE 12.9 CD8B
99014927	R	14202	C	VIS C SI 5X25/25 U ACIER QUALITE 8.8 CD8B
99070059	B	14202	C	VIS C M5X15 U ACIER QUALITE 8.8 NON PROTEGE
99081105	V	14202	C	VIS C M5X16 U ACIER QUALITE 14.9 NON PROTEGE
99081106	D	14202	C	VIS C M5X10 U ACIER QUALITE 14.9 CD8B
99097525	Y	14202	C	VIS C M1,6X4 U ACIER QUALITE 12.9 CD8B
99099794	S	14202	C	VIS C M3X12 U ACIER QUALITE 10.9 CD8B
99099795	A	14202	C	VIS C M5X10 U ACIER QUALITE 10.9 CD8B
99099796	J	14202	C	VIS C M5X16 U ACIER QUALITE 10.9 CD8B
99100148	P	14202	C	VIS C M1,6X6 U ACIER QUALITE 10.9
99100234	D	14202	C	VIS C M1,6X8 U ACIER QUALITE 10.9
99103633	M	14202	C	VIS C M4X14 U ACIER QUALITE 5.6 CD8B
99104026	X	14202	C	VIS C M3X14 U ACIER QUALITE 5-6 CD8B
99108721	W	14186	F	VIS F/90 M6X16 U ACIER QUALITE 6.6 CD8B
99108722	E	14186	F	VIS F/90 M2X6 U ACIER QUALITE 6.6 CD8B
99108723	N	14186	F	VIS F/90 M3X16 U ACIER QUALITE 6.6 CD8B
99108724	X	14186	F	VIS F/90 M3X10 U ACIER QUALITE 6.6 CD8B
99108725	F	14186	F	VIS F/90 M3X10 U ACIER QUALITE 8.8 CD8B
99108726	P	14186	F	VIS F/90 M3X12 U ACIER QUALITE 6.6 CD8B
99108727	Y	14186	F	VIS F/90 M3X5 U ACIER QUALITE 6.6 CD8B
99108728	G	14186	F	VIS F/90 M3X6 U ACIER QUALITE 6.6 CD8B
99108889	T	14186	F	VIS F/90 M5X20/20 T ACIER QUALITE 6.6 CD8B
99108890	X	14186	F	VIS F/90 M6X16 T ACIER QUALITE 6.6 CD8B
99108891	F	14186	F	VIS F/90 M6X16 U ACIER QUALITE 12.9 CD8B
99110748	H	14186	F	VIS F/90 M6X10 U ACIER QUALITE 6.6 CD8B
99112565	V	14186	F	VIS F/90 M8X16 U ACIER QUALITE 6.6 CD8B
99114587	N	14186	F	VIS F/90 M2,5X16/16 U ACIER QUAL. 5.6 CD8B

CLAS	TYPE	NOFAB	NOMFAB	NOMOD	NOART	DENOM	REFERENCE	ARTIC
441	01	M000578	SOCAPEX	U	91374602	01	CONNECTE	LJTF06RT19-11S01
441	01	M000578	SOCAPEX	M	91382652	01	CONNECTE	LJTF06RT21-16S01
441	01	M000578	SOCAPEX	W	91382653	01	CONNECTE	LJTF06RT21-75P01
441	01	M000578	SOCAPEX	R	91374772	01	CONNECTE	LJTF06RT21-75S01
441	01	M000578	SOCAPEX	N	91456042	01	CONNECTE	LJTF07RT17-05S014
441	01	M000578	SOCAPEX	C	91374605	01	CONNECTE	LJTF07RT19-11P01
441	01	M000578	SOCAPEX	E	91382654	01	CONNECTE	LJTF07RT21-16P01
441	01	M000578	SOCAPEX	G	91497632	01	CONNECTE	LJTF07RT21-39PA01
441	01	M000578	SOCAPEX	Y	91497631	01	CONNECTE	LJTF07RT21-39PB01
441	01	M000578	SOCAPEX	P	91497630	01	CONNECTE	LJTF07RT21-39PC01
441	01	M000578	SOCAPEX	Y	91500652	01	CONNECTE	LJTF07RT21-39PL01
441	01	M000578	SOCAPEX	Z	91374773	01	CONNECTE	LJTF07RT21-75P01
441	01	M000578	SOCAPEX	G	91500655	01	CONNECTE	LJTF07RT25-29PA01
441	01	M000578	SOCAPEX	R	91500654	01	CONNECTE	LJTF07RT25-29PB01
441	01	M000578	SOCAPEX	Z	91500655	01	CONNECTE	LJTF07RT25-29PC01
441	01	M000578	SOCAPEX	H	91500656	01	CONNECTE	LJTF07RT25-29PD01
441	01	M000578	SOCAPEX	R	91497633	01	CONNECTE	LJTF07RT25-29P014
494	01	M000578	SOCAPEX	E	91456041	01	RACCORD	LJTN-SA-09-1-014
494	01	M000578	SOCAPEX	J	91374765	01	RACCORD	LJTN-SA-11-1-014
494	01	M000578	SOCAPEX	Y	91456427	01	RACCORD	LJTN-SA-13-1-014
494	01	M000578	SOCAPEX	A	91374764	01	RACCORD	LJTN-SA-15-1-014
494	01	M000578	SOCAPEX	Z	91469397	01	RACCORD	LJTN-SA-15-7-014
494	01	M000578	SOCAPEX	R	91374922	01	RACCORD	LJTN-SA-17-1-014
494	01	M000578	SOCAPEX	M	91464554	01	RACCORD	LJTN-SA-17-3-014
494	01	M000578	SOCAPEX	D	91464553	01	RACCORD	LJTN-SA-17-7-014

442	R	R000214	SOURIAU	H	91332909	CONNECTE	8141-500-221
442	R	R000214	SOURIAU	V	91379766	CONNECTE	8141-500-221
442	R	R000214	SOURIAU	V	91370868	CONNECTE	8141-500-421
442	R	R000214	SOURIAU	T	91237309	CONNECTE	8141-600-121
442	R	R000214	SOURIAU	Z	91421401	CONNECTE	8141-600-211
442	R	R000214	SOURIAU	R	91451483	CONNECTE	8141-700-121
442	R	R000214	SOURIAU	Z	91451484	CONNECTE	8141-700-211
442	R	R000214	SOURIAU	P	91237312	CONNECTE	8141-800-111
442	R	R000214	SOURIAU	Y	91237313	CONNECTE	8141-800-221
442	R	R000214	SOURIAU	T	91487300	CONNECTE	8141-010-111
442	R	R000214	SOURIAU	V	91458223	CONNECTE	8142-000-143S
442	R	R000214	SOURIAU	X	91295561	CONNECTE	8142-000-313A
442	R	R000214	SOURIAU	Y	91336882	CONNECTE	8142-000-325
442	R	R000214	SOURIAU	R	91336884	CONNECTE	8142-000-328
442	R	R000214	SOURIAU	G	91336883	CONNECTE	8142-000-331
442	R	R000214	SOURIAU	Y	91299387	CONNECTE	8142-000-913
442	R	R000214	SOURIAU	G	91299388	CONNECTE	8142-000-932
442	R	R000214	SOURIAU	B	99009475	CONNECTE	8142-100-201
442	R	R000214	SOURIAU	J	99027535	CONNECTE	8142-300-101
442	R	R000214	SOURIAU	D	91311808	CONNECTE	8142-400-100
442	R	R000214	SOURIAU	Y	91311549	CONNECTE	8142-400-255

ANNEXES VII

EXTRAITS D'UNE NORME AFNOR

ET D'UN CATALOGUE FABRICANT

Norme française

enregistrée

BOULONNERIE
NOTATIONS ABREGEES
(Vue d'Ensemble)

NF
E 27-001
Octobre 1968

Remarque importante :

La présente norme donne une vue d'ensemble des notations abrégées et modes de désignation prévus dans les différentes normes d'articles de boulonnerie. Les symboles prévus comme notation abrégée ne sont à employer que dans le cas où l'on est suffisamment sûr qu'ils ne risqueront pas d'être une cause d'erreurs ou de confusions. A défaut de cette certitude, faire toujours usage des désignations en langage clair, particulièrement sur les bons de commande.

I - VIS - ECROUS - RIVETS

a) SYMBOLES :

Symboles de forme principaux	Hexagonale	H	Pour têtes et écrous de forme hexagonale.
	Crénelées	HK	Pour les écrous crénelés hexagonaux.
	Carrée	Q	Pour têtes et écrous de forme carrée.
	Cylindrique	C	Pour têtes et écrous de forme cylindrique, y compris les rivets à tête plate.
	Ronde	R	Pour tête en segment de sphère.
		Ra	Pour tête ronde de rivets avec arrondi sous tête
		Rb	Pour tête ronde de rivets avec bavure et arrondi sous tête.
	Goutte de suif .	G	Pour certains rivets à tête bombée.
	à oreille	O	Pour écrous à oreilles.
	Fraisée	F	Pour têtes fraisées plates, l'angle au sommet se place en dénominateur.
	Fraisée bombée .	FB	Pour têtes fraisées bombées, l'angle au sommet s place en dénominateur.
	Japy	J	Pour les vis à tête bombée aplatie, collet carré (dite : Japy).
	Soc	S	Pour vis de boulons pour soc de charrue.
Symboles de forme complémentaires	Ergot	E	Pour indiquer qu'une vis comporte un ergot.
	Collet carré ...	X	Pour indiquer qu'une vis comporte un collet carr
	Foré	f	Pour indiquer que l'extrémité de tige d'un rivet est forée.
Symboles de dimension relative	Large (ou haut)	L	Pour désigner les têtes plus larges (ou les écro crénelés plus hauts) que dans la série usuelle.
	Haut	h	Pour désigner les écrous plus hauts que dans la série usuelle.
	Moins large	P	Pour désigner les têtes carrées circoncrites au diamètre de tige.
	Minimum	m	Pour désigner les têtes les moins larges et les écrous les moins hauts.
	(Usuel).....	u	Pour mieux distinguer, s'il y a lieu, l'écrou de hauteur 0,8 d de l'écrou symbolisé h.

Symboles de finition	Réduit	Z	Pour désigner les têtes ou écrous de boulonne réduite.
	Ajustable	A	Pour désigner les corps de boulons ajustables (corps lisses à parachever par l'usager).
	Ajusté	AA	Pour désigner les corps de boulons ajustés (di mètre de corps lisse supérieur au diam. fileté
	Brute	N	Pour indiquer la boulonnerie brute.
	Semi-fine	T	Pour indiquer la boulonnerie semi-fine.
	Fine	U	Pour indiquer la boulonnerie fine.

b) ENONCIATION :

Enoncer toujours les caractères dans l'ordre suivant :

1° Désignation de la pièce en langage clair
2° Symbole de forme principal
3° Symbole de forme complémentaire
4° Symbole de dimension relative
5° Eventuellement désignation en langage clair de la forme d'extrémit (bout pointu par exemple)
6° Dimensions en millimètres (voir page 2)
7° Symboles de finition
8° Pour mémoire : Matière et indice de la norme dimensionnelle.

souriau

références connecteurs sans raccords

racine _____ 8M/-7C3-04SY-002-A

matière
- A : boîtier alliage léger AG5
- I : boîtier bronze marine UA9 Nfe (nous consulter)

type de boîtier
- 0 : embase à fixation par vis
- 1 : prolongateur
- 2 : embase à fixation (sans possibilité de raccord)
- 4 : traversée de cloison (nous consulter)
- 6 : fiche
- 7 : embase à fixation par écrou (montage par l'arrière de la cloison)
- 8 : embase à fixation par écrou (montage par l'avant de la cloison)

mode de raccordement des contacts
- S : soudure (contacts ø 5 et 10 mm standard)
- C : sertissage (contacts ø 1 - 1,5 - 2,5 - 3,5 mm standard)
- W : connexions enroulées (contacts ø 1 seulement)
- R : contact à sertir, rétention par clips (contacts ø 1 seulement)

taille de boîtier : 1 à 9

tiret : pour connecteurs avec contacts standard
F : pour connecteurs avec contacts coaxiaux

brochage : voir tableau

contact
- P : mâle
- S : femelle

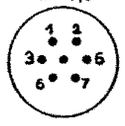
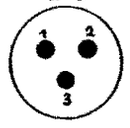
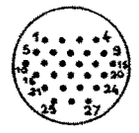
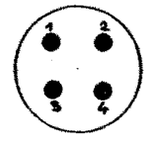
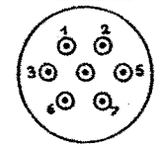
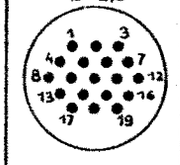
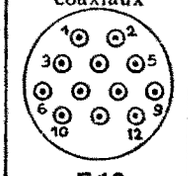
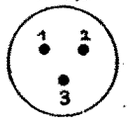
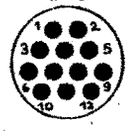
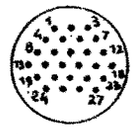
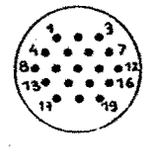
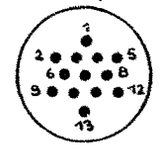
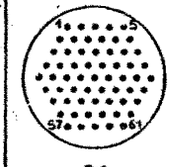
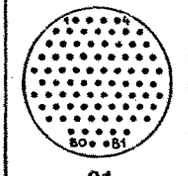
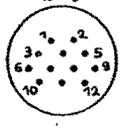
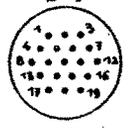
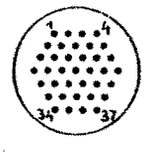
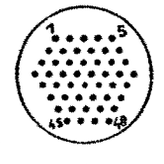
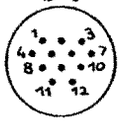
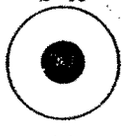
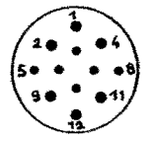
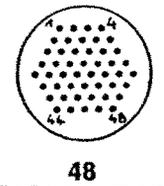
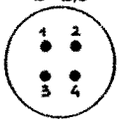
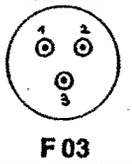
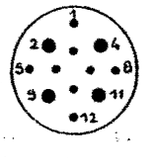
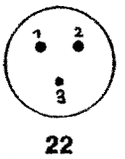
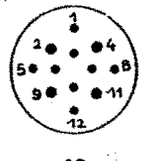
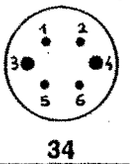
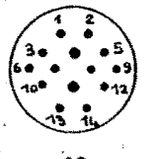
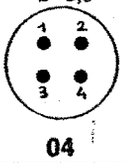
positionnement
- normal; aucune lettre dans la référence
- X, Y, Z, voir tableau

spécifications
- 001 : insert sans grommet
- 002 : boîtiers oxydés bleu (sauf pour version bronze)
- 003 : contacts coaxiaux 50 PPN
- 004 : contacts coaxiaux 75 PPN
- 005 : contacts coaxiaux 50 RPN
- sans spécification, aucun chiffre dans la référence

A - indice obligatoire

brochages

boîtiers

1	2	3	4	5	6	7	8	9
3 contacts ø 1  03	3 contacts ø 1,5  03	7 contacts ø 1,5  07	3 contacts ø 5  03	27 contacts ø 1  26	4 contacts ø 5  04	7 contacts coaxiaux  F 07	19 contacts ø 2,5  19	12 contacts coaxiaux  F 12
	7 contacts ø 1  07	3 contacts ø 2,5  03	12 contacts ø 5  12	27 contacts ø 1  27	19 contacts ø 1,5  19	13 contacts ø 2,5  13	61 contacts ø 1  61	81 contacts ø 1  81
		12 contacts ø 1  11	19 contacts ø 1  19		37 contacts ø 1  37	48 contacts ø 1  47		
		12 contacts ø 1  12	1 contact ø 10  01		6 contacts ø 1,5 6 contacts ø 2,5  44	48 contacts ø 1  48		
		4 contacts ø 2,5  04	3 contacts coaxiaux  F 03		8 contacts ø 1,5 4 contacts ø 3,5  43			
		1 contact ø 1,5 2 contacts ø 2,5  22	7 contacts ø 2,5  07		8 contacts ø 1,5 4 contacts ø 2,5  42			
			4 contacts ø 1,5 2 contacts ø 3,5  34		12contacts ø 1,5 2 contacts ø 2,5  46			
			4 contacts ø 3,5  04					

Bibliographie

[ABR1] J.R. ABRIAL, J.P. COHEN et al.
Projet Socrate, spécifications générales
Université Scientifique et Médicale de Grenoble, 1972

[ABR2] J.R. ABRIAL
Data Semantics
IFIP Working Conference, Cargèse, Corse, Avril 1974

[AFN] ASSOCIATION FRANCAISE DE NORMALISATION (AFNOR)
Recueil des normes de boulonnerie-visserie
AFNOR, 1974

[AHU] A.V. AHO, J.D. ULLMAN
The theory of parsing, translation and compiling (Volume I : Parsing)
Prentice-Hall, 1972

[BAR] R. BARBOSA
Contribution à l'étude sémantique des bases de données. Application au système Socrate
Thèse de Docteur Ingénieur à l'Université Scientifique et Médicale de Grenoble, 1975

[BOC] L. BOURRELY, E. CHOURAKI
Le système documentaire SATIN
CNRS URADCA Aix-Marseille, Janvier 1975

[BON] A. BONNET
Baobab, a parser for a rule-based system using a semantic grammar
Computer science department, Stanford University
report n° STAN-CS-78-668, Septembre 1978

[BUR] R. BURTON
Semantic grammar, an engineering technique for constructing natural language understanding systems
BBN report n° 3453, Décembre 1976

[CAC] E.F. CODD, R.S. ARNOLD, J-M. CADIOU, C.L. CHANG, N. ROUSSOPOULOS
Rendez-vous version I : an experimental english-language query formulation system for casual users of relational data bases.
Research Report, IBM Research Laboratory
San Jose, California 95193, 1978

[CDG] J. COURTIN, D. DUJARDIN, E. GRANDJEAN
Editeur lexicographique pour les langues naturelles
Document interne, Grenoble 1976

[CHA] C. CHASSAGNE
Etude et réalisation d'une méthode de transport : traduction de programmes PL360 en LP80.
Thèse de Docteur Ingénieur, Institut National Polytechnique de Grenoble, Janvier 1979

[CHO] N. CHOMSKY
Aspects of theory of syntax
MIT Press Cambridge, 1965

[CH1] Y. CHIARAMELLA
Traitement des données ambigües dans un système de base de données. Application aux bases de données démographiques.
Thèse de Doctorat es Sciences Mathématiques,
Université Scientifique et Médicale de Grenoble, Juin 1981

[CH2] Y. CHIARAMELLA
Evaluation du pouvoir discriminant d'un ensemble d'informations. Application aux bases de données.
Congrès AFCET Informatique, Novembre 1980

[CH3] Y. CHIARAMELLA
Détection automatique des variations orthographiques sur des noms propres. Définition d'un transducteur morpho-phonétique interactif.
3ème Congrès International de linguistique computationnelle,
Ottawa, 1976

[COD1] E.F. CODD
A relational model of data for large shared databanks.
Communications ACM, Vol. 13, N° 6, Juin 1970

[COD2] E.F. CODD
Further normalization of the data base relational model
IBM Research RJ909, 1971

[COU] J. COURTIN
Algorithmes pour le traitement interactif des langues naturelles
Thèse de Doctorat es Sciences Mathématiques,
Université Scientifique et Médicale de Grenoble, 1976

[DAT1] C.J. DATE
An introduction to Database Systems
Addison Wesley, 1977

[DAT2] C.J. DATE
Referential Integrity
7th International Conference on Very Large Data Base, 1981

[DEL] C. DELOBEL
Les systèmes de base de données
Université Scientifique et Médicale de Grenoble, Juin 1975

[FLU] C. FLUHR
Algorithmes à apprentissage et traitement automatique des langues
Thèse de Doctorat es Sciences, Université de Paris-Sud,
Centre ORSAY, 1977

[GRA] E. GRANDJEAN
Application d'un système de traitement de langues naturelles
pour l'indexation automatique
Avril 1978

[GUI] S. GUIASU
Information theory with applications
Mc Graw-Hill Inc., 1977

[HAL] M. HAMMER, D.J. MCLEOD
Semantic integrity in a relational data base system
First Very Large Data Base Conference, 1975

[HAM] J. HAMEON
Indexation et classement en bureautique
Thèse de Docteur de 3ème cycle, Institut National Polytechnique de Grenoble, 1981

[HAR] Z. HARRIS
Methods in structural linguistics
Chicago, 1955

[HEN] G.G. HENDRIX
The LIFER manual A guide to building practical language inferfaces
SRI, Artificial Intelligence Center, Technical note 138, 1977

[HRS] M.A. HARRISON
Introduction to formal language theory
Addison-Wesley, 1978

[HTM] J. HARTMANIS, R.E. STEARNS
Regularity preserving modifications of regular expressions
Information and Control 6, 1963

[KAF] J. KATZ, J.A. FODOR
The structure of a semantic theory dans The Structure of Language
Prentice-Hall Inc. 1964

[LAU] J. LAURENT
Exploitation de la terminologie technique normalisée à l'association française de normalisation (AFNOR)
3ème congrès européen sur les systèmes et réseaux documentaires "Franchir la barrière linguistique",
VD, Luxembourg, Mai 1977

[LYO] J. LYONS
Introduction to theorical linguistics
Cambridge University Press, 1971

[MAT] B. MATHIEU
Etude et réalisation d'un système permettant la construction de réseaux d'automates d'états finis ; Application à la production de documents en braille abrégé.
Thèse de Docteur Ingénieur, Institut National Polytechnique de Grenoble, Septembre 1980

[MAY] M. MACHTEY, P. YOUNG
An introduction to the general theory of algorithms
Elsevier North-Holland, Inc. 1978

[McL] D. MCLEOD
Abstraction in databases
Proceedings of workshop on data abstraction, databases and conceptual modelling
ACM, Pingree Park, Colorado, Juin 1980

[MYL] J. MYOLOPOULOS
An overview of knowledge representation
Proceedings of workshop on data abstraction, databases and conceptual modelling
ACM, Pingree Park, Colorado, Juin 1980

[PAO] T.W.L PAO
A solution of the syntactical induction -inference problem for a non trivialsubset of context- free language
University of Pennsylvania Ph.D., Computer Science, 1969

[PIC] C. PICARD
Théorie des questionnaires
Gauthiers-Villars, 1965

PL3 D. SUTY
Le langage PL360, Janvier 1971

[RAP] A. CULET, Y. CHIARAMELLA, E. GRANDJEAN, M. YVAIN, J. LEBEAU
Rapport du contrat d'étude THCSF/SCTF-IMAG
Maquette PIAFBCN d'analyse interactive, désignations des articles
BCN, Juin 1981

[REN] A. RENYL
Calcul des probabilités
Dunod, 1966

[SAL] A. SALOMAA
Theory of Automata
Pergamon Press, 1969

[SALT1] G. SALTON
Dynamic information and library processing
Prentice-Hall, 1975

[SALT2] G. SALTON
The smart Project
Prentice-Hall, 1971

[SCH] M.J. SCHULZ
Une banque de données terminologiques au service du traducteur ; possibilités d'interrogation dans le système TEAM.
3ème Congrès européen sur les systèmes et réseaux documentaires. "Franchir la barrière linguistique", VD, Luxembourg, Mai 1977

[SOC] SOCRATE : Manuel d'utilisation
Eca Automation, Janvier 1979

[STO] M. STONEBRAKER
Implémentation of integrity constraints and views by query modification
Proceedings of the IFIP 1977 Congress

[TAR] L. TARNOCZI
Essai de critique au structuralisme
Linguistics 57 Mouton, 1970

[UTE] Recueil des composants sous assurance de qualité NF-UTE, Service National de la qualité des composants électroniques CNET, 1977

[WAG] D. WALTZ, B. GOODMAN
Writing a natural language data base system
Proc. of 5th IJCAI, Cambridge, Mass, Août 1977

dernière page de la thèse

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU les rapports de présentation de Messieurs

. G. VEILLON, professeur à l'ENSIMAG
. V. JOLOBOFF, Ingénieur de recherche

Mademoiselle Annie CULET

est autorisée à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR INGENIEUR, Spécialité "Informatique".

. Fait à Grenoble, le 14 octobre 1981,

Le Président,

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble