



Institut National Polytechnique

Félix HOUPHOUËT-BOIGNY

CCI CAMES 2025

Candidature pour une inscription sur la liste : **LAFPT**
CTS : **Sciences et Techniques de l'Ingénieur (STI)**
Spécialité : **Informatique**

THESE

BROU Konan Marcellin

Maître de Conférences

Institut National Polytechnique Félix HOUPHOUËT-BOIGNY (INP-HB)

B.P 1093 Yamoussoukro, Côte d'Ivoire

Tel : (+225) 07 08 58 30 00, Email : marcellin.brou@inphb.ci

EFQPC
Ecole de Formation Continue et
de Perfectionnement des Cadres

☒ 1093 YAMOOUSSOUKRO (R.C.I.)
☎ (225) 30 64 66 66 FAX: (225) 30 64 66 60

ESA
Ecole Supérieure
d'Agronomie

ESCAE
Ecole Supérieure de Commerce
et d'Administration des Entreprises

site web: www.inphb.edu.ci
email : polytec@inphb.edu.ci

ESI
Ecole Supérieure
d'Industrie

ESMG
Ecole Supérieure des
Mines et de Géologie

☒ V 79 ABIDJAN (R.C.I.)
☎ (225) 22 44 51 67 FAX: (225) 22 44 32 96

ESTP
Ecole Supérieure
des Travaux Publics

Diplômes

1. Rapport de thèse
2. Thèse



Institut National Polytechnique

Félix HOUPHOUËT-BOIGNY

CCI CAMES 2025

Candidature pour une inscription sur la liste : **LAFPT**
CTS : **Sciences et Techniques de l'Ingénieur (STI)**
Spécialité : **Informatique**

RAPPORT DE THESE

BROU Konan Marcellin

Maître de Conférences

Institut National Polytechnique Félix HOUPHOUËT-BOIGNY (INP-HB)

B.P 1093 Yamoussoukro, Côte d'Ivoire

Tel : (+225) 07 08 58 30 00, Email : marcellin.brou@inphb.ci

EFPC
Ecole de Formation Continue et
de Perfectionnement des Cadres

ESA
Ecole Supérieure
d'Agronomie

ESCAE
Ecole Supérieure de Commerce
et d'Administration des Entreprises

ESI
Ecole Supérieure
d'Industrie

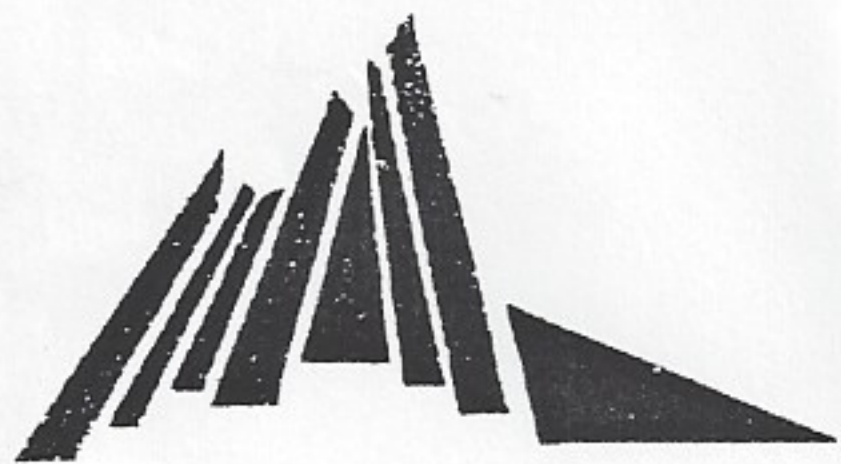
ESMG
Ecole Supérieure des
Mines et de Géologie

ESTP
Ecole Supérieure
des Travaux Publics

☒ 1093 YAMOOUSSOUKRO (R.C.I.)
☎ (225) 30 64 66 66 FAX: (225) 30 64 66 60

site web: www.inphb.edu.ci
email : polytec@inphb.edu.ci

☒ V 79 ABIDJAN (R.C.I.)
☎ (225) 22 44 51 67 FAX: (225) 22 44 32 96



UNIVERSITÉ de PAU et des PAYS de L'ADOUR
CENTRE UNIVERSITAIRE de RECHERCHE SCIENTIFIQUE

RAPPORT SUR LA SOUTENANCE DE DOCTORAT DE
Monsieur Marcellin Konan BROU

Monsieur Marcellin Konan BROU a présenté son travail intitulé "Bases de concepts: Contribution à la représentation et à l'exploitation hypertexte de concepts- le système CoDB-Web"

Son exposé a été extrêmement clair, bien préparé et bien construit. M. Brou a montré sa parfaite connaissance des sujets abordés qu'il s'agisse de Bases de Données, d'Hypertextes ou de Systèmes Experts.

Cet exposé a montré la qualité et l'importance du travail réalisé par M. Brou. Il a également démontré des qualités pédagogiques avec une grande maîtrise de la langue. Il a répondu pertinemment aux multiples questions posées.

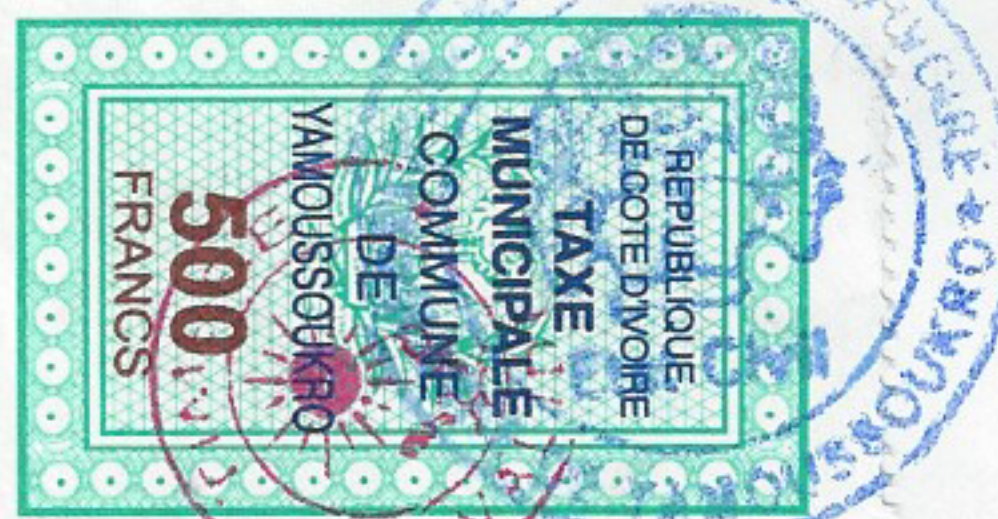
2312
Son rapport écrit est clair et rigoureux. Le travail est validé par une application concrète d'importance sur l'utilisation des plantes médicinales.

C 0055 185862
23/11/2009
ON I
yako
19/12/2016
Pour ces raisons, le jury lui a décerné sans hésitation la mention très honorable.

Lo Maire

QUOIRH SYLVAIN
Directeur des Services
Administratifs
Officier d'Etat Civil P/D

Pau le 20 Novembre 1997



Bernard CAUSSE

JM PINON

A. TESTE

CE DU CATEAU

A. Houine

B. Cay



Institut National Polytechnique

Félix HOUPHOUËT-BOIGNY

CCI CAMES 2025

Candidature pour une inscription sur la liste : **LAFPT**
CTS : Sciences et Techniques de l'Ingénieur (STI)
Spécialité : **Informatique**

THESE

BROU Konan Marcellin

Maître de Conférences

Institut National Polytechnique Félix HOUPHOUËT-BOIGNY (INP-HB)

B.P 1093 Yamoussoukro, Côte d'Ivoire

Tel : (+225) 07 08 58 30 00, Email : marcellin.brou@inphb.ci

EFPCP
Ecole de Formation Continue et
de Perfectionnement des Cadres

ESA
Ecole Supérieure
d'Agronomie

ESCAE
Ecole Supérieure de Commerce
et d'Administration des Entreprises

ESI
Ecole Supérieure
d'Industrie

ESMG
Ecole Supérieure des
Mines et de Géologie

ESTP
Ecole Supérieure
des Travaux Publics

☒ 1093 YAMOOUSSOUKRO (R.C.I.)
☎ (225) 30 64 66 66 FAX: (225) 30 64 66 60

site web: www.inphb.edu.ci
email : polytec@inphb.edu.ci

☒ V 79 ABIDJAN (R.C.I.)
☎ (225) 22 44 51 67 FAX: (225) 22 44 32 96

N° d'ordre: 384

THÈSE

présentée à

l'Université de Pau et des Pays de l'Adour

en vue de l'obtention du

DOCTORAT D'INFORMATIQUE

par

Marcellin Konan BROU

Base de concepts : Contribution à la représentation et à
l'exploitation hypertexte de concepts -
le système CoDB-Web

Soutenue le 20 novembre 1997 devant le jury composé de:

Bernard CAUSSE

Charles-François DUCATEAU

Amrane HOCINE

Jean-Marie PINON

Alain TESTE

Professeur à l'IUT de Bayonne

Professeur à l'Université René Descartes (Paris V)

Maître de conférences à l'Université de Pau

Professeur à l'INSA de Lyon

Maître de conférences à l'Université de Pau

A ma famille

Remerciements

Je souhaite ici très sincèrement remercier tous les membres du jury de l'honneur qu'ils m'ont accordé, et en particulier :

M. Charles-François DUCATEAU, Professeur à l'I.U.T. de l'Université René Descartes de Paris, qui a accepté d'être le rapporteur de cette thèse; qu'il soit remercié pour ses critiques et suggestions après une lecture approfondie de ce mémoire, et pour la séance de travail que nous avons eue durant son séjour en Côte d'Ivoire cet été.

M. Jean-Marie PINON, Professeur à l'INSA de Lyon qui a également accepté d'être le rapporteur de ce mémoire; qu'il soit particulièrement remercié pour ses remarques et propositions très pertinentes, et pour m'avoir fait profiter de ses compétences au terme d'une lecture minutieuse de mon manuscrit.

M. Bernard CAUSSE, Professeur à l'I.U.T. de Bayonne, pour avoir accepté d'être mon Directeur de recherche et pour ses encouragements.

M. Amrane HOCINE, Maître de conférences à l'Université de Pau et des Pays de l'Adour, qui m'a suivi toutes ces années en assurant l'encadrement scientifique de mes travaux; sa bonté, sa disponibilité de tous les instants, son soutien constant, ses précieux conseils ainsi que ses compétences m'ont permis d'entreprendre et de mener à bien ce travail; à travers lui, j'aimerais aussi remercier tous les membres de l'équipe ICOG en particulier M. Alain TESTE et M. Jean-Claude SAILLARD pour leur soutien constant.

Que soient associées à ses remerciements toutes les personnes qui, d'une manière ou une autre m'on aidé ou encouragé et notamment le chef de la Mission Française de Coopération et d'actions culturelles en Côte d'Ivoire pour son soutien financier.

Enfin je désire remercier de tout cœur tous mes parents pour le soutien moral qu'ils m'ont sans cesse prodigué et qui m'a été d'un secours inestimable dans la poursuite et l'achèvement de travail.

Résumé

Le travail présenté dans cette thèse porte sur les bases de concepts (BdC).

«Nous définissons une base de concepts comme étant une base de données de concepts qui utilise des facilités hypertextuelles pour la gestion et l'exploitation des concepts d'un domaine.»

L'approche que nous avons adoptée pour la modélisation d'une BdC, s'appuie sur les concepts et techniques de la terminologie. Une BdC offre les fonctionnalités suivantes :

- l'organisation des concepts du domaine en réseaux de concepts,
- la description de leur sémantique par des informations terminologiques,
- le traitement automatique des concepts grâce à leurs caractères définitoires,
- la représentation et l'exploitation hypertextuelle du réseau de concepts afin de capter toute la richesse sémantique des concepts.

La mise en place d'une telle BdC est basée sur l'exploitation corrélée d'un système de gestion de base de données qui assure la gestion classique des données, et d'un système hypertexte qui offre une interface conviviale de consultation ainsi que sa souplesse de représentation des informations. Une telle approche présente les avantages suivants :

- Elle permet d'assurer la cohérence sémantique des informations terminologiques après une mise à jour de son contenu informatif. Un processus d'hypérisation répercute automatiquement la mise à jour sur la base d'hyperdocuments.
- Elle permet également de vérifier la cohérence au niveau des liens qui n'aboutissent pas parce que le système ne trouve pas le nœud destination. La détermination de ces nœuds est réalisée par la comparaison des liens qui arrivent et qui partent d'un documents avec ceux de la structure qui contient les liens du document.
- Elle permet aussi de créer une vue de la base d'hyperdocuments correspondant au choix de l'utilisateur des informations terminologiques ou caractères définitoires qu'il souhaite manipuler.

Le système MEDTRAD qui est une base de concepts sur les plantes médicinales utilisées en médecine traditionnelle valide notre approche de BdC qui est utilisée dans ce cadre comme composante explicative d'un système expert.

Mots clés : concept, base de concepts, terminologie, informations terminologiques, caractères définitoires, hyperdocument, hypérisation, MEDTRAD, système expert.

Table des matières

Introduction	1
1 Introduction aux bases de concepts	7
1.1 Introduction	7
1.2 Base de concepts	8
1.2.1 Quelques définitions	8
1.2.2 Architecture d'une base de concepts	10
1.3 Notre approche de base de concepts	19
1.3.1 Une représentation de données basée sur le modèle relationnel	20
1.3.2 Une exploitation intégrant des facilités hypertextes	21
1.4 Conclusion	22
2 Base de données et hypertextes: état de l'art	25
2.1 Introduction	25
2.2 Base de données relationnelle	25
2.2.1 Le modèle de données	26
2.2.2 Le modèle de requêtes	26
2.2.3 SGBD avancées	27
2.3 Les hypertextes	28
2.3.1 Les concepts de base	29
2.3.2 Exploration du réseau	32
2.3.3 Quelque systèmes Hypertextes	33
2.3.4 Problèmes des systèmes hypertextes	36
2.3.5 Quelques solutions	38
2.4 Systèmes Hypertextes et SGBD	39
2.5 Conclusion	44
3 Le modèle de la base de concepts	47
3.1 Introduction	47
3.2 Logiques terminologiques	47

3.2.1	Le formalisme	47
3.2.2	Comparaison entre logique terminologique et base de concepts	51
3.3	Représentation formelle d'un concept	51
3.3.1	Définition	52
3.3.2	Propriétés et opérateurs d'un concept	53
3.4	Le modèle de données de la base de concepts	55
3.4.1	Classification des informations terminologiques	55
3.4.2	Classification des caractères définitoires	57
3.4.3	Instanciation des propriétés des concepts	58
3.4.4	Le modèle conceptuel de la BdC	60
3.4.5	Le schéma relationnel de la base de concepts	65
3.5	Conclusion	67
4	Un modèle de représentation et gestion hypertexte des concepts	69
4.1	Introduction	69
4.2	Choix du modèle de données hypertexte	69
4.2.1	Les unités informatives	70
4.2.2	Les prédicats	71
4.2.3	Les liens hypertextes	72
4.3	Un modèle hypertexte pour une base de concepts	75
4.3.1	Les unités informatives	75
4.3.2	Les liens hypertextes	78
4.4	Exploitation hypertexte de la BdC	81
4.4.1	Manipulation de la BdC	82
4.4.2	Le processus d'hypérisation	90
4.4.3	Maintenance de la BdC	93
4.5	Conclusion	101
5	L'implémentation de la BdC	103
5.1	Introduction	103
5.2	Architecture générale de la BdC	103
5.2.1	Présentation fonctionnelle	103
5.2.2	Circulation des informations entre CoDB-Web et son environnement	105
5.3	L'environnement de développement de la BdC	106
5.3.1	Le Système hypertexte WWW	106
5.3.2	Le langage HTML	108
5.3.3	Le langage d'implémentation de la BdC	115
5.3.4	Le système de gestion de base de données relationnelle	120

Table des matières	v
5.4 Méthodologie d'instantiation de la BdC	122
5.5 Conclusion	123
6 Application : le système MEDTRAD	125
6.1 Introduction	125
6.2 Présentation de la BdC de MEDTRAD	126
6.2.1 Délimitation du domaine	126
6.2.2 Collecte des informations	127
6.2.3 Modélisation du domaine sous forme d'arbres conceptuels	127
6.2.4 Modélisation de l'application sous forme d'un schéma conceptuel . . .	130
6.2.5 Etablissement des fiches	130
6.2.6 Utilisation de MEDTRAD	133
6.2.7 La page d'accueil de Medtrad	135
6.2.8 Accès à Medtrad par navigation	136
6.2.9 Accès à Medtrad par une recherche initiant la navigation	142
6.3 Extension future de MEDTRAD : MEDTRAD+	145
6.3.1 Architecture du système MEDTRAD+	146
6.3.2 Fonctionnement du système MEDTRAD+	146
6.4 Conclusion	149
Conclusion	151
Bibliographie	153
Annexe	160

Table des figures

1.1	Relations entre concept, objet et terme [CAM 96].	8
1.2	Schéma conceptuel du domaine.	11
1.3	Arbres conceptuels du domaine.	13
1.4	Relation d'association entre deux concepts.	14
1.5	Gestion de la hiérarchie entre concepts.	18
1.6	Architecture générale d'une BdC	20
2.1	La relation Lien.	26
2.2	Exemple de réseau de nœuds et de liens.	32
3.1	Arbre de concepts d'un domaine D.	52
3.2	Relation entre deux concepts liés d'un domaine D.	53
3.3	Association entre deux concepts liés.	59
3.4	Caractères définitoires de type concept.	59
3.5	Caractères définitoires de type relation.	60
3.6	Schéma conceptuel de donnée de la BdC.	61
3.7	Structure d'un hyperdocument.	63
3.8	Table d'interprétation des liens.	65
3.9	Instanciation de la BdC pour le domaine enregistrement de voitures	66
4.1	Schéma conceptuel d'un document hypertexte.	70
4.2	Lien permanent à destination fixe.	72
4.3	Lien permanent à destination contextuelle.	73
4.4	Lien contextuel à destination fixe.	74
4.5	Lien contextuel à destination contextuelle.	74
4.6	Structuration des unités informatives de la BdC.	76
4.7	Exemples d'UIIT et d'UICD.	77
4.8	Les liens de référence.	80
4.9	Interface de consultation de la BdC	82
4.10	Notion de point de vue.	89
4.11	Instances d'un même objet documentaire.	89

4.12	Structure logique d'un document.	93
4.13	Lien vers une partie d'un document.	97
4.14	Modélisation des liens en HTML.	98
4.15	Suppression de la première UI d'un document.	99
4.16	Après la suppression de la première UI d'un document.	99
4.17	Après la suppression d'une UI d'indice différent de 1 et n d'un document.	99
4.18	Après suppression de la dernière UI d'un document.	100
4.19	Suppression d'une UI qui contient ou qui est la destination d'un lien de référence.	100
4.20	Suppression d'une partie d'une UI.	100
5.1	Schéma de fonctionnement général de la BdC	104
5.2	Architecture logicielle de CoDB-Web.	105
5.3	Circulation des informations entre CoDB-Web et son environnement.	105
5.4	Architecture client/serveur.	107
5.5	Code HTML de la page d'accueil de Medtrad	111
5.6	Page d'accueil de Medtrad.	112
5.7	Environnement de développement d'Intrabuilder	119
6.1	Relations sémantiques entre les concepts du domaine.	128
6.2	Arbres conceptuels du domaine.	129
6.3	Schéma conceptuel de l'application.	130
6.4	Description des informations terminologiques.	131
6.5	Liste des caractères définitoires.	132
6.6	Base d'hyperdocuments de Medtrad.	133
6.7	le menu général de Medtrad.	135
6.8	Fiches terminologiques du concept Plante.	136
6.9	Fiches terminologiques du concept Plante ligneuse.	137
6.10	Fiches terminologiques du concept Acacia senegal.	138
6.11	Activation du lien hypertextuel Définition.	139
6.12	Activation du lien hypertextuel Image.	140
6.13	Activation d'un lien de description.	141
6.14	Changement d'arbre conceptuel.	142
6.15	Interface de requête.	143
6.16	Requête sur les informations terminologiques.	144
6.17	Requête sur les caractères définitoires.	145
6.18	Architecture du système MEDTRAD+.	146
19	Menu général général	175
20	Formulaire fiches terminologiques.	176
21	Fiches terminologiques de Acacia senegal.	177

22 Caractères définitoires de type relation de Acacia senegal. 178

Introduction

Position du problème

Le projet base de connaissances généralisées (BCG) sur lequel travaille notre équipe est né du constat de la convergence de plus en plus flagrante de trois domaines : base de données, base de connaissances et hypertexte. Une BCG peut être alors définie comme l'extension d'une base de données, d'un système d'information de structure complexe telle que les hypertextes, intégrant des caractéristiques et/ou des systèmes experts.

Dans le cadre de ce projet, [FOR 94] a proposé un premier modèle d'hyperdocument (document hypertexte) expert fondée sur la coopération d'une base de données et d'une base d'hyperdocuments experts, sans modification fondamentale du modèle relationnel. Un hyperdocument expert est défini comme un hyperdocument dans lequel des connaissances sont introduites au niveau des unités informatives permettant ainsi d'intégrer un aspect dynamique à leur contenu informatif.

[DES 95] a ensuite spécifié et normalisé un modèle hypertexte expert; ce modèle étend et généralise le premier modèle.

Le modèle de base de concepts décrit dans ce mémoire étend le modèle d'hyperdocument expert par la possibilité : (i) de stocker dans une base de données le contenu informatif et la structure logique d'un hyperdocument, (ii) de générer dynamiquement l'hyperdocument par un processus d'hypérisation.

Une méthode et un outil d'aide à la structuration et à la représentation des concepts du domaine de l'application hypertexte, fondés sur le modèle précédent, sont proposés à l'utilisateur. Les arbres conceptuels du domaine sont reliés entre eux par des relations sémantiques formant ainsi un réseau de concepts qui permet de décrire complètement un domaine.

Cette approche que nous proposons est fondée sur la notion de base de concepts qui s'appuie sur les concepts et techniques de la terminologie [GOU 92].

La terminologie qui est l'étude scientifique du langage, est fondée sur un modèle tripartite dont les mots clés sont : le concept ou la notion, l'objet et le terme [CAM 96]. Un concept est un savoir sur un objet qui a pour nom un terme. La démarche terminologique consiste à analyser les concepts et à les structurer en arbres de concepts ou de domaine en fonction de leurs caractères. Les caractères sont les propriétés des objets conceptualisés. Ils permettent de rapprocher ou de différencier les objets entre eux.

Cette structuration d'un domaine en un réseau de concepts, dans lequel chaque concept permet de révéler ses rapports avec les autres concepts grâce aux liens sémantiques ou notionnels qui les relient, est utilisée dans certains domaines de l'informatique. Parmi ces domaines on peut citer :

- *Base de données*: certaines recherches sur l'intégration des schémas de BD utilisent une structuration des concepts [BLA 93] [BON 94] [MIR 96] [LAR 96] [SAO 96] pour exprimer la sémantique des schémas à fédérer et pour découvrir les correspondances et les conflits existant entre leurs éléments.
- *Système de recherche d'information documentaires (SRI)*: la réponse à une requête dans les SRI est basée sur la mise en correspondance de la représentation sémantique du texte des document (ensemble de termes d'indexation) et la représentation des termes de la requête [MEN 94]. Ces termes d'indexation sont structurés en arbres de dépendance sémantique [SAV 90], et les fonctions de mise en correspondance sont traduites par la synonymie ou par des liens notionnels.
- *Enseignement assisté par ordinateur*: en EAO, l'apprenant ne maîtrise véritablement le domaine étudié que lorsqu'il est parvenu à résoudre des problèmes types de ce domaine et à assimiler les concepts théoriques correspondants [FOR 93]. Ces concepts théoriques du cours sont structurés en hiérarchies de concepts.
- *Systèmes experts*: la modélisation de l'expertise explicative des concepts d'une base de connaissances [BRO 97] [HOC 91] [MOE 96] permet de rompre avec les explications classiques des systèmes experts qui consiste à présenter les règles déclenchées. Cette modélisation est basée sur la structuration des concepts du domaine en hiérarchies de concepts.

Le principal inconvénient de ces différents modèles de structuration des concepts est leur spécificité à chaque domaine. Ils sont donc non exportables.

Objectif de la thèse

Notre objectif est de mettre à la disposition de tous les domaines dont la structuration des concepts est un élément important un modèle de base de concepts.

«Nous définissons une base de concepts comme étant une base de données de concepts qui utilise des facilités hypertextuelles pour la gestion et l'exploitation des concepts d'un domaine.»

Elle doit offrir les fonctionnalités suivantes :

- l'organisation des concepts du domaine en réseaux de concepts,
- la description de leur sémantique par des informations terminologiques,
- le traitement automatique des concepts grâce à leurs caractères,
- la représentation et l'exploitation hypertextuelle du réseau de concepts afin de capter toute la richesse sémantique des concepts.

La mise en place d'une telle base de concepts, nécessite l'utilisation de deux systèmes de gestion de données : un SGBD et un système de gestion hypertexte. Le SGBD assure la gestion classique des données et le système hypertexte offre son interface conviviale de consultation ainsi que sa souplesse de représentation des informations.

On peut aborder cette coopération entre SGBD et système hypertexte selon deux approches :

- la création d'une interface de consultation permettant la navigation hypertextuelle dans le schéma d'une BD [MEN 94].
- l'intégration d'informations d'une BD dans les hypertextes [FOR 94].

Aucune de ces deux approches ne proposent un mécanisme de traitement hypertextuelle des informations de la BD qui sont constituées d'un texte important ou d'une image. Elles n'abordent pas aussi le problème de la maintenance et de la cohérence de la base d'hypertextes.

Contributions de la thèse

Notre recherche porte sur la proposition d'un modèle de base de concepts accessible à tous les utilisateurs informaticiens ou non. Pour réaliser cette exigence nous avons exploré deux axes de recherche : le premier porte sur les SGBD afin de proposer un modèle conceptuel rigoureux de la base de concepts; le second concerne les systèmes hypertextes afin d'offrir une interface privilégiant la navigation lors de la consultation des informations de la base de concepts.

- Notre première contribution est la modélisation conceptuelle d'une base de concepts fondée sur une approche terminologique.
- Notre deuxième contribution porte sur l'exploitation d'une base de concepts utilisant la coopération entre deux systèmes de gestion de données :
 - un SGBD pour la gestion des données et
 - un système de gestion hypertexte pour fournir, non seulement une interface de consultation graphique par "association d'idées" mais aussi permettre de représenter certaines informations du modèle conceptuel d'une base de concepts de façon hypertextuelle.
- Notre dernière contribution concerne la conception des documents hypertextes. Il est important que la mise à jour des hyperdocuments et la vérification de la cohérence des la base de concepts puisse se faire de manière simple et facile. Pour atteindre cet objectif, nous avons opté pour une représentation séparée des informations et des liens.

Organisation du mémoire

Ce mémoire est organisé en trois parties :

La première partie relative à l'état de l'art et à l'architecture d'une base de concepts est constituée des chapitres I et II.

Le chapitre I présente d'abord les différentes composantes d'une base de concepts, ensuite insiste sur l'approche que nous avons adoptée pour la conception et l'exploitation d'une base de concepts.

Le chapitre II établit d'abord un état de l'art sur les concepts et techniques hypertextes puis présente des extensions hypertextes de bases de données afin de situer le cadre de notre

étude.

La seconde partie relative à la modélisation conceptuelle et à l'exploitation hypertextuelle d'une base de concepts est constituée des chapitres III et IV.

Le chapitre III commence par la présentation d'une base de concepts de manière formelle, puis décrit le modèle conceptuel dans le formalisme Entité/Association et termine par la représentation du modèle relationnel associé au modèle conceptuel de données.

Le chapitre IV présente le modèle hypertexte pour une base de concepts. Il décrit l'exploitation de la base de concepts en insistant sur les mécanismes de maintenance et de vérification de la cohérence dans la base d'hyperdocuments.

La dernière partie relative à l'implémentation et à l'utilisation du système CoDB-Web (Concept DataBase Web) est constituée des chapitres V et VI.

Le chapitre V présente les fonctionnalités et l'environnement de développement de CoDB-Web qui est le système de gestion et d'exploitation hypertexte de la BdC fondée sur l'utilisation du Word Wide Web.

Le chapitre VI présente une mise en œuvre du système CoDB-Web dans le cadre d'une application appelée MEDTRAD portant sur la MEDecine TRADitionnelle en Afrique. L'objectif de cette application est de capitaliser la connaissance africaine sur les plantes médicinales.

La conclusion fait un bilan du travail réalisé et présente les extensions futures.

Chapitre 1

Introduction aux bases de concepts

1.1 Introduction

La maîtrise d'un domaine n'est réalisée que lorsqu'on est parvenu à assimiler les concepts de ce domaine. Ces concepts sont analysés et classifiés par les terminologues qui en précisent leur sens par l'intermédiaire de leurs propriétés ou caractères.

La problématique des bases de concepts a été abordé dans [BON 94] pour l'intégration de schémas de bases de données. Dans cette approche, une base de concepts est caractérisée par trois niveaux : terminologique, caractérisation formelle et représentation.

Notre approche de base de concepts s'inspire de cette architecture; cependant, hors mis l'extension des informations du niveau terminologique, elle se différencie sur les points suivants :

- la définition d'une Bdc
- la signification du niveau représentation et
- la gestion et l'exploitation d'une Bdc

Dans ce chapitre nous introduisons d'une part la notion de concept et d'autre part nous présentons l'approche hybride (base de données et hypertexte) que nous avons retenu pour structurer et exploiter de manière efficace les concepts d'un domaine donné.

1.2 Base de concepts

Une base de concepts désigne la base de données qui permet de gérer l'ensemble des concepts d'un domaine donné. Nous préciserons cette première définition un peu plus loin.

1.2.1 Quelques définitions

La terminologie est la science qui étudie les termes d'un domaine en précisant les relations qui existent entre ces différents termes. Elle est fondée sur un modèle tripartite dont les mots clés sont: le concept, l'objet et le terme [CAM 96].

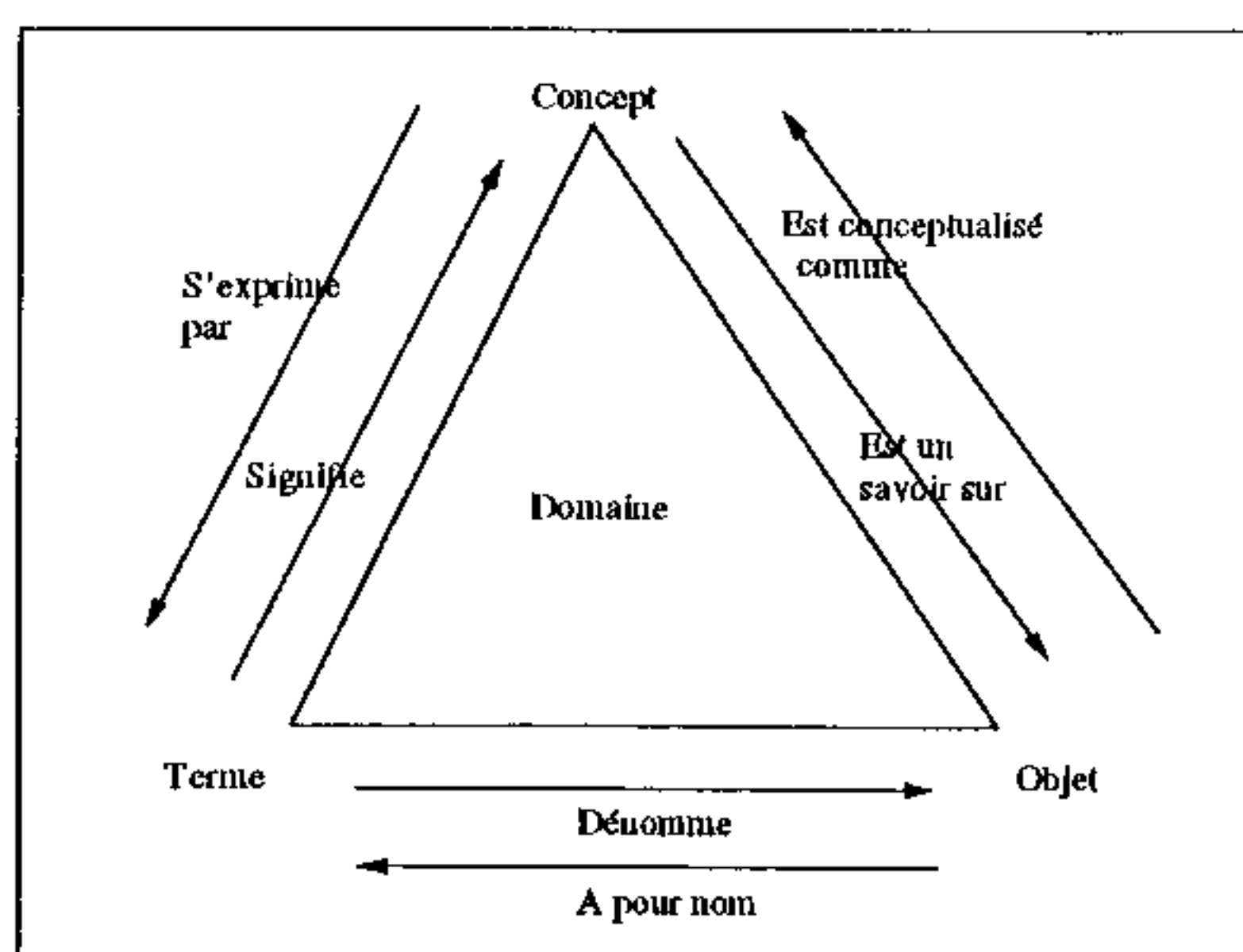


FIG. 1.1 – Relations entre concept, objet et terme [CAM 96].

La figure 1.1 montre les relations qui existent entre ces trois entités à l'intérieur d'un domaine.

- **Domaine** : la notion de concept est étroitement liée à la notion de domaine qui peut être considéré comme une activité, une problématique ou une discipline.

«Nous définissons un domaine comme étant un ensemble de concepts reliés les uns aux autres par des relations sémantiques».

C'est ce que les linguistes appellent univers du discours.

A l'intérieur d'un domaine une langue de spécialité est choisie et est utilisée par les experts du domaine afin d'avoir une interprétation commune des concepts qu'ils utilisent. En botanique par exemple le latin est la langue véhiculaire pour nommer les plantes.

Un concept n'a de signification que dans un domaine. En effet, un concept peut être défini dans plusieurs domaines mais avec des objets différents. De tels concepts sont dits homonymes.

Par exemple, le concept épi désigne une sorte d'inflorescence, alors que dans la vie courante il désigne une mèche de cheveux.

- Concept : un concept est une réalité abstraite, c'est à dire une idée qui représente un aspect de la réalité isolée par l'esprit [HAB 88], ou encore c'est une unité de pensée constituée d'un ensemble de caractères attribués à un objet ou à une classe d'objets [BON 94].

Pour les terminologues, un concept est une unité de pensée constituée par abstraction à partir des propriétés communes à un ensemble d'objets.

«Nous définissons un concept comme étant une idée abstraite donc générale pour définir un ensemble d'objets ayant des caractères ou propriétés communs».

Par exemple le concept Plante est une idée abstraite qui ne désigne pas une plante particulière, mais plutôt une notion générale qui convient pour désigner toutes les plantes.

- Objet : *«un objet est un élément de réalité qui peut être perçu ou conçu. Il peut être matériel (exemple plante) ou immatériel (exemple maladie)».*

Un objet est une instanciation d'un concept, c'est la représentation concrète d'un concept. Par analogie au modèle objet, il peut être comparé à un objet d'une classe, c'est à dire une instance de classe.

- Terme : *«un terme désigne au moyen d'une unité linguistique un concept défini dans une langue de spécialité».*

Il peut être constitué de un ou plusieurs mots.

Exemple : plante et plante ligneuse sont deux termes qui désignent deux concepts différents.

- Caractère : les concepts sont constitués d'un ensemble de caractères qui correspondent à des propriétés des objets. Ils permettent de différencier ou de rapprocher les objets conceptualisés entre eux.

Par exemple le concept Plante peut avoir les caractères suivants :

famille, durée de vie, hauteur, racine, tige, feuille, fleur, fruit.

«Un caractère est donc une propriété d'un concept; il peut être assimilé à un attribut d'une relation du modèle relationnel ou à un attribut d'un objet du modèle objet ».

1.2.2 Architecture d'une base de concepts

En analysant les différentes définitions que nous venons de présenter, nous constatons qu'un concept doit être décrit à deux niveaux : sa terminologie et ses caractères décrivant les propriétés de son objet.

Dans une base de concepts, il faut également préciser la manière dont sont représentés les éléments de ces deux niveaux.

Une BdC sera donc constituée de trois niveaux qui sont :

- le niveau terminologique,
- le niveau caractère et
- le niveau représentation.

«Nous définissons une Base de Concepts (notée BdC) comme une base de données de concepts qui utilise des facilités hypertextuelles pour la gestion et l'exploitation des concepts d'un domaine.»

La différence avec l'approche de [BON 94], hors mis l'ajout d'autres informations terminologiques se situe au niveau de la partie représentation. En effet, dans son approche le niveau représentation indique comment les concepts sont représentés dans les schémas sous-jacents des bases de données à fédérer (relation, tuple d'une relation, attribut, classe, ...). Dans notre approche, ce niveau correspond à un modèle de représentation de données (hypertexte, BD, ou la coopération entre BD et hypertexte) utilisé pour représenter les informations des niveaux terminologique et caractère.

Dans les paragraphes suivants, nous détaillons les trois niveaux (terminologique, caractère et représentation) constituant une BdC.

Le domaine qui nous servira d'exemple est un système de gestion des enregistrements de voitures d'une entreprise. La figure 1.2 illustre un schéma Entité/Association partiel simplifié des données de l'application.

Une voiture est composée d'une carrosserie, de roues, d'un châssis et d'un moteur (cette agrégation est symbolisée par le Et logique) ; elle est fabriquée par un constructeur unique ; elle peut être utilisée par plusieurs personnes et être réparée dans plusieurs garages.

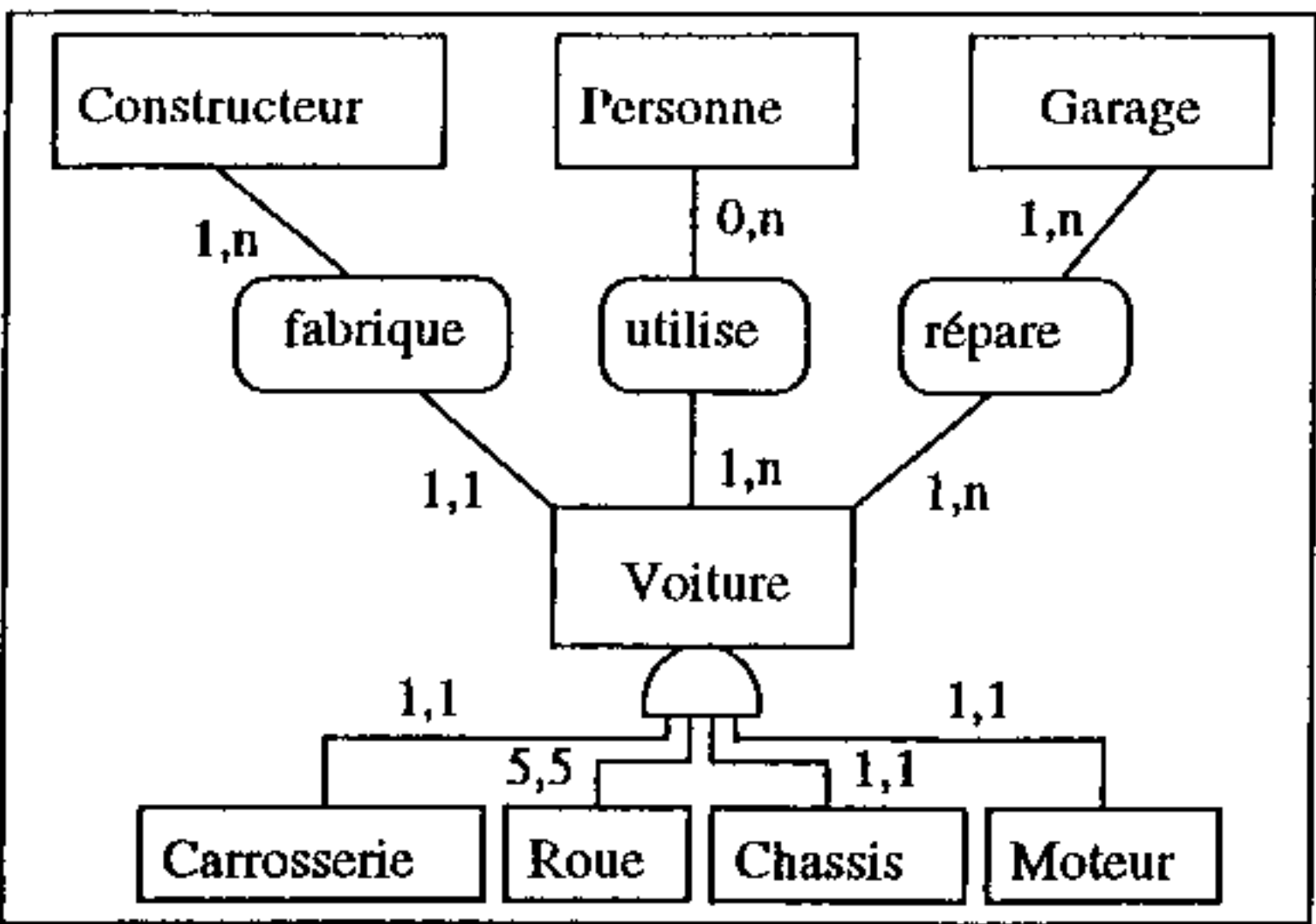


FIG. 1.2 – Schéma conceptuel du domaine.

1.2.2.1 Le niveau terminologique

Le niveau terminologique contient des informations textuelles et graphiques donnant la sémantique des concepts. Ces informations comprennent les notions : terme, définition, concept générique, concept lié et contexte décrites dans [BON 94]. Nous les avons complétées en introduisant les notions :

- langue : qui est un élément indispensable en terminologie multilingues,
- synonyme : permettant d'exprimer les différentes formes de synonymie (synonymie réelle, abréviation, ...),
- homonyme : afin d'éviter les erreurs d'interprétation dues aux concepts qui ont la même orthographe,
- sous-concept : facilite la représentation des arbres conceptuels du domaine, et
- image : qui est un élément important pour la composante hypertexte d'une BdC.

Nous décrivons ci-dessous les différents champs du niveau terminologique.

- **Terme** : comme nous l'avons défini précédemment, un terme est moyen permettant de désigner un concept d'un domaine où est définie une langue de spécialité.

Il peut être représenté par un seul mot (terme simple) ou par plusieurs mots (terme complexe). Les mots d'un terme complexe peuvent aussi désigner des concepts.

Exemple : **voiture de course** est un terme complexe, il est constitué de la conjonction des termes simples **voiture** et **course**.

- **Langue** : dans un domaine, une langue de spécialité est choisie et est utilisée par les experts du domaine. En botanique par exemple, le latin est la langue véhiculaire pour nommer ou pour classer les plantes.

Exemple : les noms des plantes suivantes sont en latin : *Acacia senegal*, *Ocimum basilicum*, *Borreria verticillata*.

- **Synonyme** : un terme peut être accompagné de plusieurs synonymes et homonymes. Deux termes sont synonymes s'ils désignent le même concept. La synonymie représente l'équivalence entre termes réellement synonymes, entre des variantes orthographiques, ou entre des abréviations [MEN 94]. La synonymie peut être décrite au sein d'une même langue ou dans des langues différentes.

Deux termes synonymes ont une même valeur de référencement et des conditions d'utilisation rigoureusement identiques [GOU 92]. Ils sont donc substituables l'un à l'autre. Cette notion de synonymie permet de réduire le nombre d'entrée de la base de concepts.

Exemple : le concept *véhicule à moteur* a pour synonyme *automobile* en français ; *voiture* a pour synonyme *car* en anglais.

- **Homonyme** : deux termes sont homonymes s'ils ont la même représentation orthographique mais désignent des concepts différents. L'homonymie peut exister à l'intérieur d'un même domaine ou deux domaines différents.

Exemple : en botanique le concept *oignon* est une plante potagère à bulbe comestible ; il désigne également la bulbe souterraine de certaines plantes telles que le *lis* et la *tulipe*.

En horlogerie, il désigne une grosse montre et en anatomie, il désigne un durillon se formant à la base du gros orteil.

- **Concepts génériques** : cette notion permet de traduire les relations hiérarchiques entre concepts (généralisation). Elle représente la relation *est-un* dans les modèles à objets. En effet, chaque concept peut être associé à un concept plus général appelé concept générique de manière à former un arbre conceptuel du domaine.

Si le domaine contient des concepts de nature très diverse, correspondant à plusieurs terminologies (par exemple les concepts *Personne* et *Voiture* du domaine de l'enregistrement de voitures), il y aura plusieurs arbres de domaine. Ces différents arbres de domaine sont reliés les uns aux autres par l'intermédiaire de l'information terminologique *Concept lié* que nous définirons plus loin.

Exemple : comme le montre la figure 1.3, le concept *Personne* est le concept générique des concepts *Homme* et *Femme*.

Un concept peut avoir plusieurs concepts génériques; c'est le cas du concept *Solex* qui a les concepts génériques *Moto* et *Vélo*.

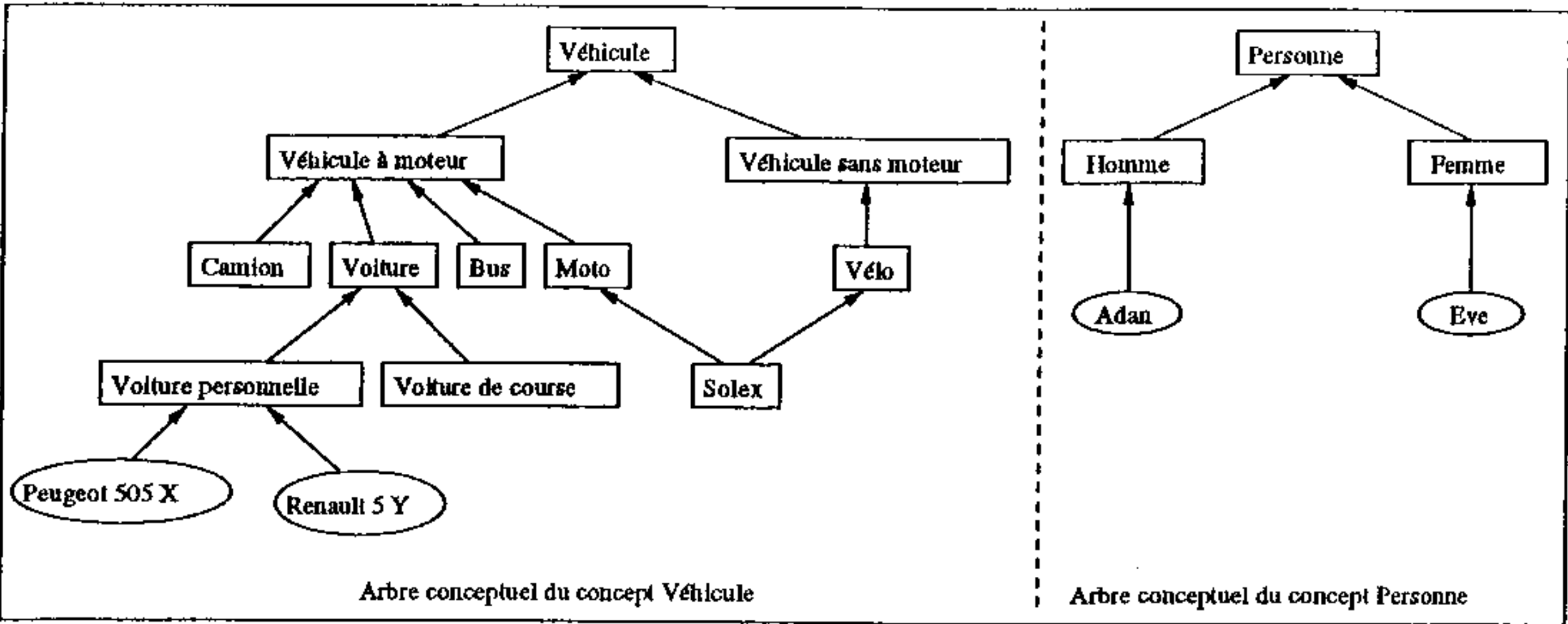


FIG. 1.3 – Arbres conceptuels du domaine.

Dans la figure 1.3, les rectangles représentent les concepts complexes, les cercles les concepts élémentaires (X et Y sont des numéros de voiture), et les flèches traduisent la relation est-un.

On peut rapprocher cette notion de concept générique de la notion de subsumption utilisée dans les logiques terminologiques (cf. 3.2 page 47). Cependant la subsumption est plus générale car en plus de la classification des concepts en hiérarchie de concepts, elle permet d'inférer de nouvelles relations d'héritage entre les concepts.

- **Sous-concepts** : de même qu'à un concept peut être associé un concept plus général, à concept générique va être associés ses concepts spécifiques ou sous-concepts de premier rang.

Cette notion de sous-concepts permet de traduire la relation inverse de la hiérarchie (spécialisation). Ceci facilite la prise de conscience de la structure hiérarchique de la BdC, ainsi que la représentation des arbres conceptuels du domaine.

Exemple : la figure 1.3 nous présente les sous-concepts de *Personne* qui sont : *Homme* et *Femme*.

- **Concepts liés** : mis à part son concept générique, un concept peut être sémantiquement lié à d'autres concepts. Cette information terminologique permet de représenter la proximité sémantique entre concepts. Elle exprime une relation autre que la synonymie, l'homonymie et la hiérarchie. Elle peut permettre de décrire une relation d'association,

d'antonymie ou de voisinage entre concepts par exemple.

Exemple : la plante *Borreria verticillata* est souvent confondue avec une espèce voisine qui est le *Mitracarpus scaber*; nous dirons que ces deux concepts sont liés par une relation de voisinage.

Dans la figure 1.4, les concepts *Personne* et *Voiture* sont liés respectivement l'un à l'autre par les relations d'association *utilise* et *est utilisée*.

Cette relations d'association a la même sémantique que celle qui est utilisée dans le modèle Entité/Association.

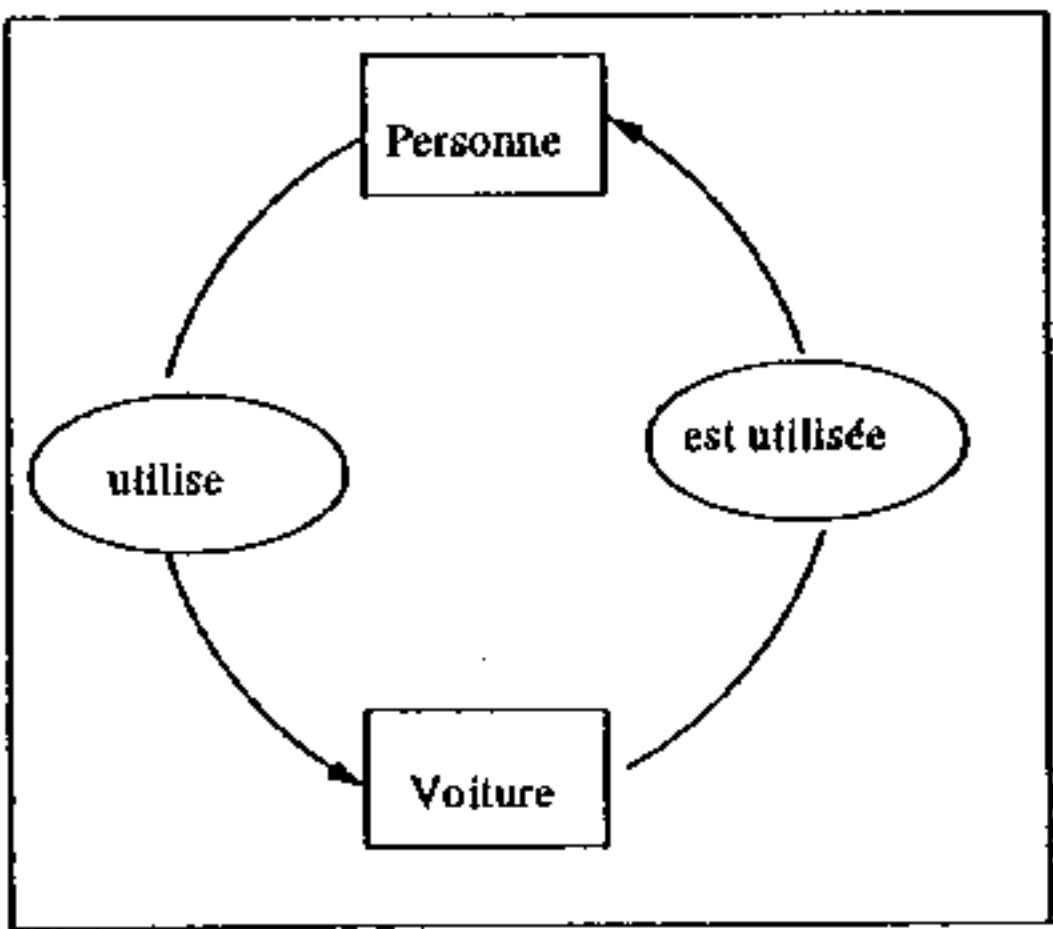


FIG. 1.4 – Relation d'association entre deux concepts.

- Définition : une définition permet l'énonciation des caractères, des qualités propres à un concept; c'est la signification du terme qui le désigne.

Cette définition est donnée sous forme d'un texte en langage naturel, tel qu'on peut le trouver dans un dictionnaire ou sur une fiche terminologique. Ce texte peut aussi comporter des concepts.

Exemple : la définition du concept *Voiture* :

« Une voiture est un véhicule servant au transport des personnes et des choses. Elle possède un moteur qui peut être diesel ou à essence, une carrosserie, et des roues... »

Dans cette définition, les mots soulignés représentent des concepts ou font référence à d'autres concepts. Par exemple le terme *personne* fait référence au concept *Personne*.

- Contexte : le contexte d'un concept est un texte qui précise les conditions de son utilisation. Ce texte a les mêmes caractéristiques que la définition d'un concept, ie il peut faire référence à d'autre concepts.

Cette information terminologique permet de définir des sous-domaines dans lesquels les concepts peuvent être utilisés sans ambiguïté.

Par exemple le contexte du concept Voiture ci-dessous restreint le type de voiture étudié :

«Les voitures dont il est question ici font partie des voitures familiales possédant un moteur diesel avec quatre roues motrices...».

- **Image** : une image ou un graphique permet d'illustrer l'objet d'un concept. Elle facilite la prise de conscience du sens d'un concept.

Cette information terminologique peut permettre de prendre connaissance de la sémantique de plusieurs concepts à la fois. Il est donc intéressant d'avoir une image réactive dont la description permet d'accéder à d'autres concepts de la base et d'en avoir une meilleure compréhension grâce à leur représentation visuelle.

Par exemple l'image du concept Voiture permet de décrire également le concept Carrosserie.

1.2.2.2 Le niveau caractère

Le niveau caractère est aussi appelé dans [BON 94] niveau caractère définitoire ou niveau caractérisation formelle. Dans la suite de ce document, nous utiliserons indifféremment les termes caractère et caractère définitoire; surtout le dernier qui est un néologisme, afin d'insister sur le fait que les caractères sont plus que de simples attributs car ils sont porteurs d'une certaine sémantique permettant de définir ou de caractériser les concepts.

Le niveau information terminologique que nous avons décrit précédemment contient des informations de nature textuelle et graphique qui sont bien adaptées pour prendre connaissance de la sémantique des concepts. Cependant, elles ne conviennent pas pour effectuer des opérations de recherche "fine" sur les concepts.

Le niveau caractère permet de décrire l'ensemble des caractères définitoires que l'on retrouve dans chaque concept; il s'agit de trouver une description de chaque concept basée sur une même structure afin de pouvoir les traiter automatiquement dans une requête.

Un caractère définitoire est une propriété de l'objet qui est conceptualisé comme un concept; on peut le rapprocher à la notion d'attribut d'un objet du modèle objet ou à la

notion d'attribut d'une relation en base de données. Cependant, un caractère définitoire se distingue de ces deux notions par le fait que sa sémantique est aussi exploitée dans la BdC.

Un caractère définitoire peut être élémentaire (non décomposable) ou complexe c'est à dire formé d'autres caractères.

Chaque instance de concept hérite de l'ensemble des caractères définitoires appartenant à ce concept mais avec des valeurs différentes.

Considérons par exemple les concepts suivants et leurs caractères définitoires :

Carrosserie : nombre de portes, couleur

Roue : diamètre, pression

Chassis : numéro

Moteur : nombre de Km, puissance

Constructeur : nom, adresse, téléphone

Garage : nom, adresse, téléphone

Voiture : numéro, marque, modèle, Carrosserie, Roue, Moteur, Constructeur, Garage

Les cinq premiers concepts ont tous des caractères définitoires élémentaires (comme les attributs d'une relation) contrairement au concept Voiture qui est composé par l'agrégation d'autres concepts; ses caractères définitoires numéro, marque et modèle sont élémentaires par contre ses caractères définitoires Carrosserie, Roue, Chassis, Moteur, Constructeur et Garage sont complexes.

Le concept Voiture peut être instancié avec les valeurs suivantes :

numéro(1085 TJ 64),

marque(Peugeot),

modèle(505 GTI),

Carrosserie(5, rouge),

Roue(100, 2.5 Kg),

Chassis(XXXXX),

Moteur(20 000 Km, 11 CV),

Constructeur(Peugeot, 112 rue Championnet 75018 Paris, 01 25 12 12 01),

Garage(Rapido, 32 avenue de l'Université 64000 Pau, 05 59 22 41 05)

Les opérations de recherche pertinente sur les concepts dans la BdC sont basées sur la manipulation de leurs caractères définitoires. Une BdC doit pouvoir répondre aux interrogations suivantes :

sur la sémantique d'un CD, «Qu'est ce qu'un numéro de voiture?»

sur la valeur d'un CD, «Comment interpréter un numéro de voiture?»
sur les concepts, «Quels sont les voitures de marque Peugeot ayant la couleur rouge?».

Le § 4.4.1.2 page 84 est consacré à la recherche d'informations dans une BdC.

1.2.2.3 Le niveau représentation

Le niveau représentation indique le moyen utilisé pour stocker les informations terminologiques et les caractères définitoires des concepts d'un domaine.

Deux types de représentation sont possibles :

- par une approche hypertexte ou
- par une approche base de données

1.2.2.4 Par l'utilisation d'un modèle hypertexte

On peut représenter toutes les informations de la BdC à l'aide d'un hypertexte. En effet, les informations terminologiques sont de nature textuelle, de plus les différents concepts sont sémantiquement reliés entre eux pour former un arbre de concepts.

L'avantage de cette solution est qu'elle est bien adaptée, non seulement, pour exprimer la structure arborescente d'une base de concepts, mais aussi pour fournir une interface de consultation conviviale de la base par "association d'idées". Un nœud hypertexte contiendra les informations terminologiques d'un concept et un autre les caractères définitoires afin de pouvoir prendre connaissance de sa sémantique de façon visuelle.

Cependant elle présente des inconvénients :

- manque d'outils d'aide à la conception des hyperdocuments,
- problème de désorientation, c'est à dire la perte de vue du but de la recherche, dû à la prolifération des liens et
- surtout de maintenance et de maintien de la cohérence de la base d'hyperdocuments.

1.2.2.5 Par l'utilisation d'un modèle de BD relationnel

Dans une base de données relationnelle, chaque concept générique est représenté par une relation et chaque information terminologique ou chaque caractère définitoire par un attribut. Chaque tuple représente une instance du concept.

Exemple : la relation décrivant le concept Voiture :

```
Voiture(terme, concept générique, contexte, définition, image,  
        numéro, marque, modèle, carrosserie, roue, chassis, moteur).
```

Les attributs de cette relation représentent les informations terminologiques et les caractères définitoires du concept Voiture. Chaque tuple représente une instance du concept Voiture.

Les hiérarchies entre concepts se ramènent à un problème de gestion d'arbres en SQL. A première vue cela peut paraître impossible parce que le SQL ne dispose pas de notion de pointeur et de récursion comme dans les L3G.

Cependant [ROZ 96] a montré qu'en SQL un arbre peut être représenté par la relation arbre(père, fils) où l'attribut fils constitue la clé. La notion de pointeur est simulée par cette clé. Quant à la récursion, elle peut être remplacée par une simple itération (dérécursion). Beaucoup de "dialectes" SQL ont été justement étendus afin de prendre en compte l'itération (Transact SQL de Sybase par exemple).

Le calcul de toute la descendance d'un concept générique est donc réalisable en SQL.

Exemple : modélisation de la hiérarchie dans le concept Personne.

Concept			Relation conceptuelle	
numconcept	terme		numconcept1	numconcept2
1	personne		1	2
2	Homme		1	3
3	Femme		2	4
4	Adan		3	5
5	Eve			
.....				

FIG. 1.5 – Gestion de la hiérarchie entre concepts.

Dans la figure 1.5 les deux relations Concept et Relation conceptuelle permettent de gérer la hiérarchie entre concepts. Le lien entre les deux relations est assuré par l'attribut numconcept (numéro du concept père). L'attribut numconcept2 est le numéro du concept fils.

L'avantage de cette solution est que les mises à jours sont prises en charge par le SGBD. De plus son langage de requêtes permet d'accéder efficacement aux informations de la base.

Un inconvénient est qu'il ne dispose pas d'interface qui permet d'accéder naturellement aux informations de la BdC par "association d'idées" comme dans un hypertexte.

1.2.2.6 Par l'utilisation d'un modèle de BD orienté-objet

Compte tenu de la nature hiérarchique de la base de concepts, et de l'existence des notions d'héritage implicite entre concepts génériques et sous-concepts, une BD orientée-objet nous semble aussi bien adaptée à la représentation des informations de la BdC (cf. Fig. 1.3).

Dans un tel modèle, chaque concept générique sera représenté par une classe, et chaque information terminologique ou chaque caractère définitoire par un attribut de la classe. Chaque instance de concept sera représentée par une instance de classe, c'est à dire par un objet de la classe.

Cette solution souffre du même inconvénient que les SGBDR; elle n'offre pas d'interface conviviale de type hypertexte d'accès aux données de la BdC.

1.3 Notre approche de base de concepts

Dans la figure ci-dessous, nous donnons l'architecture générale d'une BdC.

La démarche de conception d'une BdC peut être résumée comme suit :

1. Les concepts du monde réel concernant le domaine à modéliser sont extraits et structurés suivant le modèle de BdC (détermination des informations terminologiques et des caractères définitoires) par l'expert du domaine.
2. Ensuite toutes ces informations sont stockées dans une base de données relationnelle.
3. Enfin l'exploitation de cette BD est réalisée en intégrant des facilités hypertextes, aussi bien pour la consultation que pour la représentation de certaines informations de la BdC (Définition, contexte et Image).

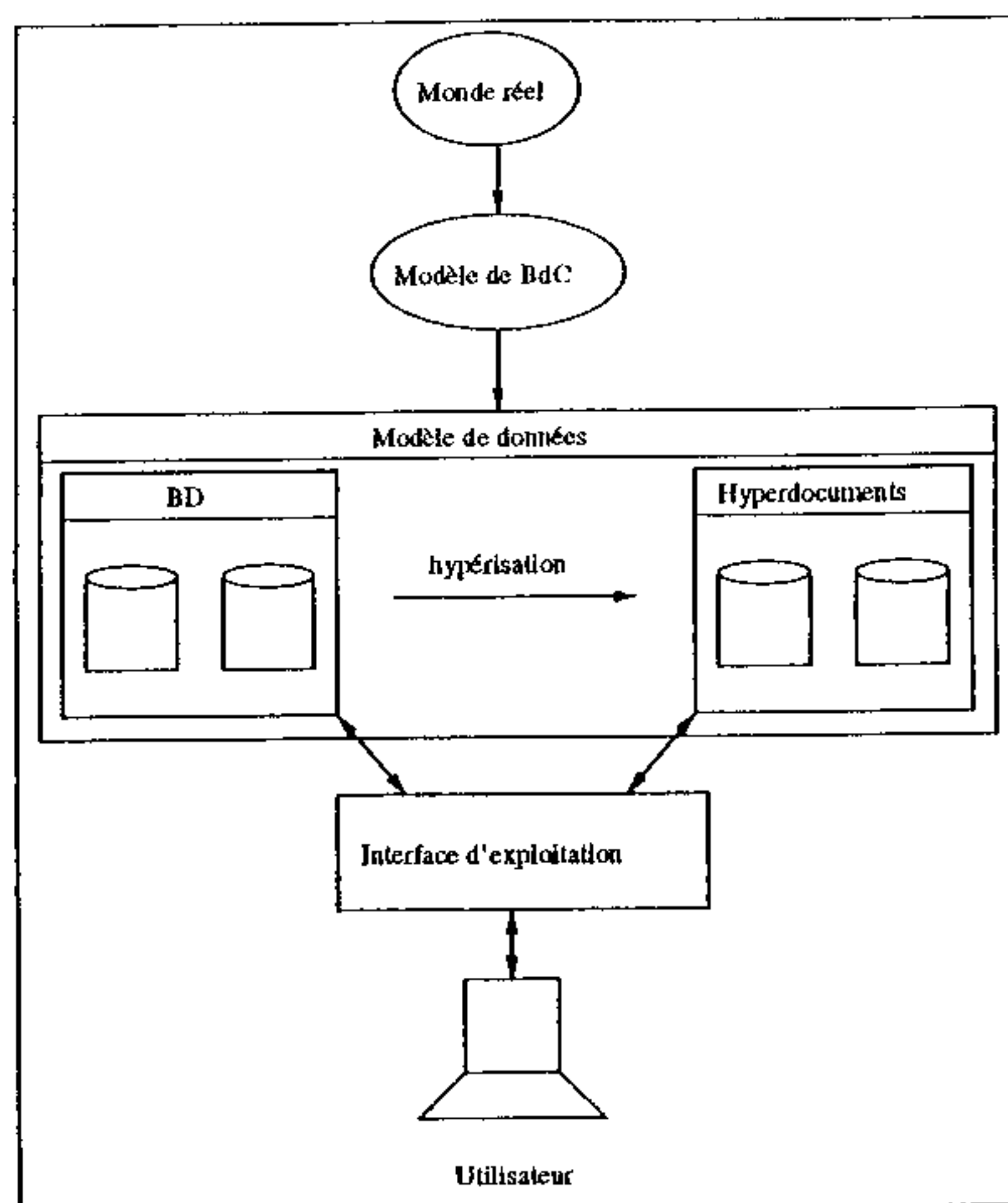


FIG. 1.6 – Architecture générale d'une BdC .

1.3.1 Une représentation de données basée sur le modèle relationnel

Pour représenter les données de la BdC nous avons opté pour le modèle relationnel. Les informations terminologiques et les caractères définitoires y seront stockés.

Ce choix s'est naturellement imposé à nous parce que c'est l'un des systèmes de gestion de données que notre équipe a retenu pour la construction d'une BCG [DES 95] [FOR 94].

Nous pouvons citer entre autres comme qualités d'un SGBD relationnel :

- ils sont les plus utilisées dans le monde industriel,
- l'accès facile aux données grâce à leur langage de requête et
- la gestion transparente des fichiers assurée par le SGBDR.

Cependant, comme les SGBDR n'offrent pas une interface de consultation conviviale des données qu'ils gèrent, nous avons opté pour une exploitation de la BdC intégrant des concepts hypertextes afin d'accéder aisément à la sémantique des concepts.

1.3.2 Une exploitation intégrant des facilités hypertextes

Dans une BdC, chaque concept est relié à au moins un concept générique de manière à former un arbre de concepts. Différents arbres de concepts sont aussi reliés entre eux par des liens sémantiques. Dans le but de capter toute la richesse sémantique d'une BdC, nous utiliserons les concepts et techniques hypertexte pour permettre à l'utilisateur d'aller au delà des simples interrogations classiques de la base de données, en lui permettant :

- d'une part de naviguer dans la base de concepts par "association d'idées",
- et d'autre part, d'avoir accès à une information complète hypertextuelle.

Certaines informations terminologiques telle que Définition d'un concept peuvent être également traitées de manière hypertexte car elles sont de nature textuelle et contiennent des informations faisant référence à d'autres concepts ou à des caractères définitoires (cf. définition de Voiture page 14).

L'apport de l'hypertexte à l'exploitation d'une BdC est double :

- faciliter la création d'une interface hypertextuelle conviviale de consultation, afin que l'utilisateur puisse naviguer librement dans la BdC par "association d'idées".
- permettre la représentation hypertexte de certaines informations terminologiques pouvant être traitées comme des hypertextes.

Il y aura donc une coopération entre le SGBD et le système hypertexte utilisés pour la création et l'exploitation d'une BdC. Dans cette coopération, le SGBD assurera la gestion des données, il facilitera surtout la recherche d'informations. Quant au système hypertexte il apportera sa souplesse de représentation visuelle des informations de la BdC. Comme nous l'avons souligné précédemment, la coopération entre le SGBD et le système hypertexte peut être abordé selon les deux approches :

- la création d'une interface de consultation permettant la navigation hypertextuelle dans le schéma d'une BD,
- l'intégration d'informations d'une BD dans les hypertextes

Aucune de ces deux approches ne propose un mécanisme de traitement hypertextuelle des informations de la BD qui sont constituées d'un texte très volumineux et/ou complexe.

Notre proposition, non seulement, inclut les deux approches ci-dessus, mais aussi fournit une aide à la conception des hypertextes.

En effet dans l'hypertexte représentant les informations terminologiques de type hypertexte, nous faisons la distinction entre leur contenu informatif et l'aspect logique de l'hypertexte. Ces deux aspects sont stockés séparément dans la BdC.

L'intérêt d'une telle approche est la maintenance aisée des hyperdocuments (cf. § 4.4.3 page 93).

Une telle approche présente trois avantages :

- Elle permet d'assurer la cohérence sémantique des hypertextes après une mise à jour de leur contenu informatif. Un processus d'hypérisation (opération de transformation d'un document textuel linéaire en réseau de nœuds hypertextes, (cf. § 4.4.2 page 90) répercute automatiquement la mise à jour sur la base d'hyperdocuments.
- Elle permet également de vérifier la cohérence au niveau des liens (liens intra ou inter-documents, cf. § 4.4.3.2 page 95) qui n'aboutissent pas parce que le système ne trouve pas le nœud destination. La détermination de ces nœuds est réalisée par la comparaison des liens qui arrivent et qui partent d'un document avec ceux de la structure qui contient les liens du document.
- Elle permet aussi de créer une vue de la base d'hyperdocuments correspondant au choix de l'utilisateur des informations terminologiques ou des caractères définitoires qu'il souhaite manipuler (cf. § 4.4.1.3 page 87).

1.4 Conclusion

Nous avons présenté les deux principaux éléments d'une base de concepts : les informations terminologiques et les caractères définitoires.

Les informations terminologiques de nature textuelle, permettent de prendre conscience de la sémantique des concepts d'un domaine.

Quant aux caractères définitoires, ils permettent d'effectuer des recherches de concepts dans la BdC afin de les différencier ou de les rapprocher les uns des autres.

Ces éléments sont stockés dans une BD relationnelle gérée et exploitée de façon hypertextuelle. Cette approche induit la coopération de deux systèmes de gestion de données : le SGBD et le système hypertexte. Le couplage entre ces deux systèmes permet de tirer parti de leurs qualités respectives, la recherche pour le SGBD et la navigation pour le système hypertexte afin d'avoir une gestion et une exploitation efficace de la BdC (cf. § 3.4).

Enfin nous avons proposé une aide à la conception d'hypertextes basée sur la séparation du contenu informatif de la structure logique (structuration logique en un réseau de nœuds et de liens) des hyperdocuments afin de faciliter la maintenance et la vérification de la cohérence de la base d'hyperdocuments(cf. § 4.4.3).

Les chapitres suivants présentent de façon plus détaillée le modèle de données et l'interface de consultation de la BdC.

Chapitre 2

Base de données et hypertextes : état de l'art

2.1 Introduction

Avec l'apparition de nouvelles applications (bureautique, CAO, EAO, génie logiciel, hypermédias, ...) qui manipulent une quantité importante d'information de nature complexe, les SGBD ont montré leurs limites. D'où la nécessité d'étendre la nature des données traditionnelles manipulées par les SGBD à de nouveaux types de données complexes. Cette extension est souvent réalisée par un couplage des SGBD et des systèmes de gestion hypertexte afin de tirer parti de leurs qualités respectives.

Dans ce chapitre, nous commençons par rappeler quelques notions fondamentales des bases de données relationnelle, puis par décrire les systèmes de gestion hypertexte, en insistant sur leurs qualités et leurs insuffisances respectives, nous terminons par la présentation de quelques systèmes nés de la coopération entre ces deux systèmes de gestion de données.

2.2 Base de données relationnelle

Les bases de données relationnelles ont prouvé leur efficacité dans le traitement de l'information des applications classiques de gestion (comptabilité, paye, gestion de stock, ...). Ce succès est dû à la simplicité du modèle relationnel fondé sur la notion mathématique de relation [COD 70].

Parmi les SGBD les plus utilisés, nous pouvons citer, les systèmes : INGRES, ORACLE et DB2 [GAR 90] [ORA 89] [REM 89].

2.2.1 Le modèle de données

Le modèle relationnel représente le monde réel sous forme de relations ou de tables qui sont caractérisées par un ensemble d'attributs. Chaque attribut (nom d'une colonne) est une variable prenant ses valeurs dans un ensemble de valeurs appelé domaine.

Par exemple la relation Concept qui modélise les tous les concepts de la BdC.

numconcept	terme	langue	définition	contexte	image
1	plante	français			pl.jpg
2	Acacia senegal	latin			pl5.jpg

FIG. 2.1 – La relation Lien.

Chaque ligne de cette table, appelée aussi tuple, représente un concept unique caractérisé par : un numéro, un terme, une langue, une défintion, un contexte et une image.

2.2.2 Le modèle de requêtes

Pour la définition, le contrtrôle et la manipulation des données, les SGBBR disposent de langages ensemblistes basés sur l'algèbre relationnelle tels que le langage déclaratif SQL (Structured Query Langage) et le langage graphique QBE (Query By Example).

Exemple de requête :

"Quels sont la définition, le contexte et l'image du concept voiture?"

```
SELECT  définition, contexte, image
FROM    Concept
WHERE   terme = "voiture";
```

Cette requête crée une relation dont les attributs sont ceux de la clause SELECT; la mise à jour de la relation Lien est automatiquement repercutée sur cette relation.

Afin d'assurer la confidentialité des données, certains SGBDR utilisent la notion de vue pour restreindre l'accès aux données partagées. Une vue est une présentation logique d'une partie de la structure de la base de données; on peut la comparer à une table virtuelle car elle ne possède pas de donné propres comme les autres tables de la BD [TEC 96].

Exemple de vue dans Système R :

```
DEFINE VIEW vue1
AS SELECT terme, nomcds, typecds
FROM Concept, Comportecds, CD_simple
WHERE (Concept.numconcept = Comportecds.numconcept) AND
      (Comportecds.numcds = CD_simple.numcds);
```

La vue est une entité dynamique, toute modification des relations dont elle dépend se répercute automatiquement sur elle.

Cette requête crée la nouvelle relation *vue*₁ qui dépend des relations Concept, Comporte_cds et CD_simple. Ses attributs sont ceux qui figurent dans la clause SELECT. Cette relation se présente comme suit :

*vue*₁(terme, nomcds, typecds)

où

terme est le nom d'un concept, nomcds désigne un caractère définitoire simple et typecds son type (texte ou numérique).

Cette vue permet d'accéder uniquement aux caractères définitoires simples de la BdC.

Dans la mise en place de la BdC, nous définissons un langage de requêtes simple basé sur le standard SQL pour l'accès aux informations de la BdC (cf. 4.4.1.2 page 84). Nous proposons également la notion de point de vue qui s'inspire de la vue relationnelle afin de présenter aux utilisateurs une partie restreinte de la base d'hyperdocuments, qui coorespond aux informations terminologiques ou aux caractères définitoires mieux maîtrisés, qu'ils souhaitent manipuler.

2.2.3 SGBD avancées

L'efficacité des SGBDR est due à la simplicité des types de données manipulés (entier, réel, chaîne de caractères, ...). Cette pauvreté des types de données ne permet pas de capter toute la sémantique du monde réel modélisé et de traiter de nouveaux types de données complexes, structurés et volumineux ou de gérer les documents hypermédias [CHE 90] [GAR 90]. C'est le cas par exemple des systèmes : O2 Web et Oracle Web System (cf. § 2.4 page 39).

D'où la création de nouveaux axes de recherche sur les BD qui étendent les fonctionnalités des SGBDR. Ces SGBD sont appelés SGBD avancés [GAR 90].

Parmi ces SGBD avancés, on peut citer :

- **les SGBD orientées objet** : ils utilisent les techniques de la programmation orientée objet pour modéliser les données et introduire un aspect dynamique dans la modification du schéma conceptuel et surtout, ils valorisent les types de données complexes.

Par exemple : les systèmes Generis, Ontos, Gemstone, Orion, O2 [ABI 87] [CHR 88] [KEM 91] [BOU 94].

- **les Bases de données déductives** : elle permettent de déduire des informations nouvelles à partir de la connaissance stockée dans la base de données grâce à des langages de règles issus de la logique des prédicats ou grâce à des langages de l'intelligence artificielle tel que Prolog.

Par exemple : les systèmes Deduce II, Bdgen, Postgres, Sabrina, Epsilon [BON 86] [COR 87] [DES 90] [ELL 87] [GAR 90] [HEB 89] [LEB 93].

- **les Bases de données réparties** : elles permettent de gérer plusieurs BD localisées sur plusieurs sites interconnectés par un réseau.

Par exemple : le système System R^* [GAR 90].

- **les Base de données étendues** : elles cherchent à atteindre les objectifs des BD objets et des BD déductives en restant dans un contexte relationnel [GAR 90]. Elle permettent de gérer de nouveaux types de données tels l'image et le texte.

Par exemple : le système IMAID, VACBI [CHR 88].

Nous nous sommes particulièrement intéressé à l'extension des SGBDR aux nouveaux types de données multimédia. Cette extension permet de combler d'une part la pauvreté de l'interface utilisateur des SGBD, grâce à une interface hypertextuelle qui favorise la recherche d'informations par navigation, le plus naturellement possible. D'autre part, elle permet aussi de manipuler des informations volumineuses, structurés et complexes contenues dans des documents hypermédias qui prennent de l'importance avec le développement rapide du réseau Internet.

2.3 Les hypertextes

Le concept d'hypertexte a été introduit pour permettre une consultation de documents volumineux et complexes [CON 87] [DAN 90] [FIS 89] [JUL 88] [NAR 89] [SAV 90]. On le

compare à une base de données débarrassée de la lourdeur des SGBD [DAN 90].

Sa genèse est attribuée à Vannevar Bush [BUS 45] conseiller scientifique de F. D. ROOSEVELT. Dans un article intitulé "As we may think", il a introduit la notion d'organisation de la documentation de manière à "reproduire artificiellement le processus mental de la pensée humaine qui opère par association d'idées" [DAN 90]. Mais c'est Ted NELSON qui précisa dans les années 60 le concept actuel d'hypertexte. Il définit un système hypertexte comme "l'association d'un langage naturel avec les possibilités de l'ordinateur d'établir des liaisons interactives et des affichages dynamiques d'un texte non linéaire" [DAN 90] [LAR 88].

Pour [SAV 90], les systèmes hypertextes permettent de rompre avec la linéarité des textes traditionnels car il permettent de voyager dans une collection d'informations par association d'idées.

Un système hypertexte est donc un logiciel qui permet la manipulation des documents de nature diverse (texte, graphique, image,...) grâce aux capacités dynamiques des ordinateurs.

Le concept d'hypertexte a été étendu afin de prendre en compte de nouvelles formes de médias, d'où les concepts de systèmes hypermédias et de systèmes multimédias :

- Un système multimédia gère des informations de toutes sortes au travers de plusieurs médias (lecteur audio, vidéo, CD ROM,...).
- Un système hypermédia est un système multimédia qui gère les informations selon les techniques hypertextes.

2.3.1 Les concepts de base

Un système hypertexte offre les trois fonctionnalités [SAV 90] suivantes :

- l'accès à une base d'informations souvent de nature multimédia,
- la structuration de l'information à l'aide d'un réseau et
- la présentation d'une interface de navigation conviviale dans laquelle on peut activer des références.

Ces fonctionnalités des systèmes hypertextes sont essentiellement basées sur les notions de nœuds et de liens.

2.3.1.1 Hyperdocument

Un document hypertexte est appelé hypertexte ou hyperdocument, sa représentation sous-jacente est un réseau de nœuds connectés entre eux par des liens.

Le terme d'hyperdocument est utilisé pour signifier que c'est un réseau d'informations organisé selon l'approche non linéaire des hypertextes, contrairement à un document classique qui structure séquentiellement les informations.

2.3.1.2 Nœud

Un nœud ou granule d'information est un contenant d'information. Il n'y a pas de restriction sur le type d'information. Il peut être du texte, du graphique, une séquence audio ou vidéo, etc.

On distingue deux sortes de nœuds: les nœuds élémentaires et les nœuds composés:

- Un *nœud élémentaire* contient une information homogène considérée comme atomique (indécomposable), i.e. ayant le même type. Il peut être une partie d'une image, d'un graphique, d'un texte, d'une séquence audio/vidéo, etc. C'est la plus petite entité référençable par un lien.
- Un *nœud composé* est un nœud non atomique, il rassemble dans un objet hypertexte unique plusieurs autres nœuds pouvant être élémentaires et/ou composés. Cette notion de nœud composé permet de structurer de façon hiérarchique un hyperdocument.

2.3.1.3 Lien

Un lien connecte un nœud source à un ou plusieurs nœuds destinations; il exprime la relation sémantique qui existe entre ces nœuds ou entre les informations qu'ils renferment. Cette sémantique peut être interprétée comme:

- des renvois à une partie se trouvant à l'intérieur d'un même document,
- des références vers d'autres documents,
- l'exécution d'un traitement de données

On distingue en général trois sortes de liens: les liens hiérarchiques ou organisationnels, les liens de référence et les liens exécutable.

- Les liens hiérarchiques sont des liens qui relient un nœud composé à ses nœuds fils, définissant ainsi une arborescence de nœuds. Il sont aussi appelés liens internes.
- Les liens de référence permettent de connecter un nœud ou partie de nœud à un nœud ou partie de nœud [CON 87]. Ils servent à établir une relation entre deux parties différentes d'un même hyperdocument ou de deux documents distincts. Il sont aussi appelés liens externes.
- Les liens exécutable sont des liens externes qui lancent l'exécution d'un programme informatique pour le traitement de données.

Un lien est caractérisé par ses extrémité ou ancres. L'ancre source est représentée sous forme de bouton. L'activation de ce bouton affiche la fenêtre correspondante à l'ancre destination (Fig. 2.2).

2.3.1.4 Hypergraphe

La représentation interne d'un hypertexte est un réseau de nœuds et de liens modélisé par un graphe orienté, appelé aussi hypergraphe. [SAV 90] définit un hypergraphe comme étant l'ensemble des liens reliant les différentes composantes d'un hyperdocument. L'espace de navigation est appelé hyperespace.

2.3.1.5 Aspect médiatique

L'aspect médiatique (Fig. 2.2) indique la façon dont les informations contenues dans les nœuds vont être présentées à l'utilisateur à travers un média [DES 95]. En général, une fenêtre permet de visualiser le contenu d'un nœud. Un mécanisme de défilement (scrolling) permet de voir la suite des informations d'un nœud dont le contenu entier ne peut être visualisé dans une fenêtre.

La figure 2.2 présente un document hypertexte sous deux aspects différents: sa représentation externe (aspect médiatique) et sa représentation interne (hypergraphe). Le fait de cliquer sur le bouton b (parties encadrées) du nœud A, amène à l'écran l'information contenue dans le nœud B.

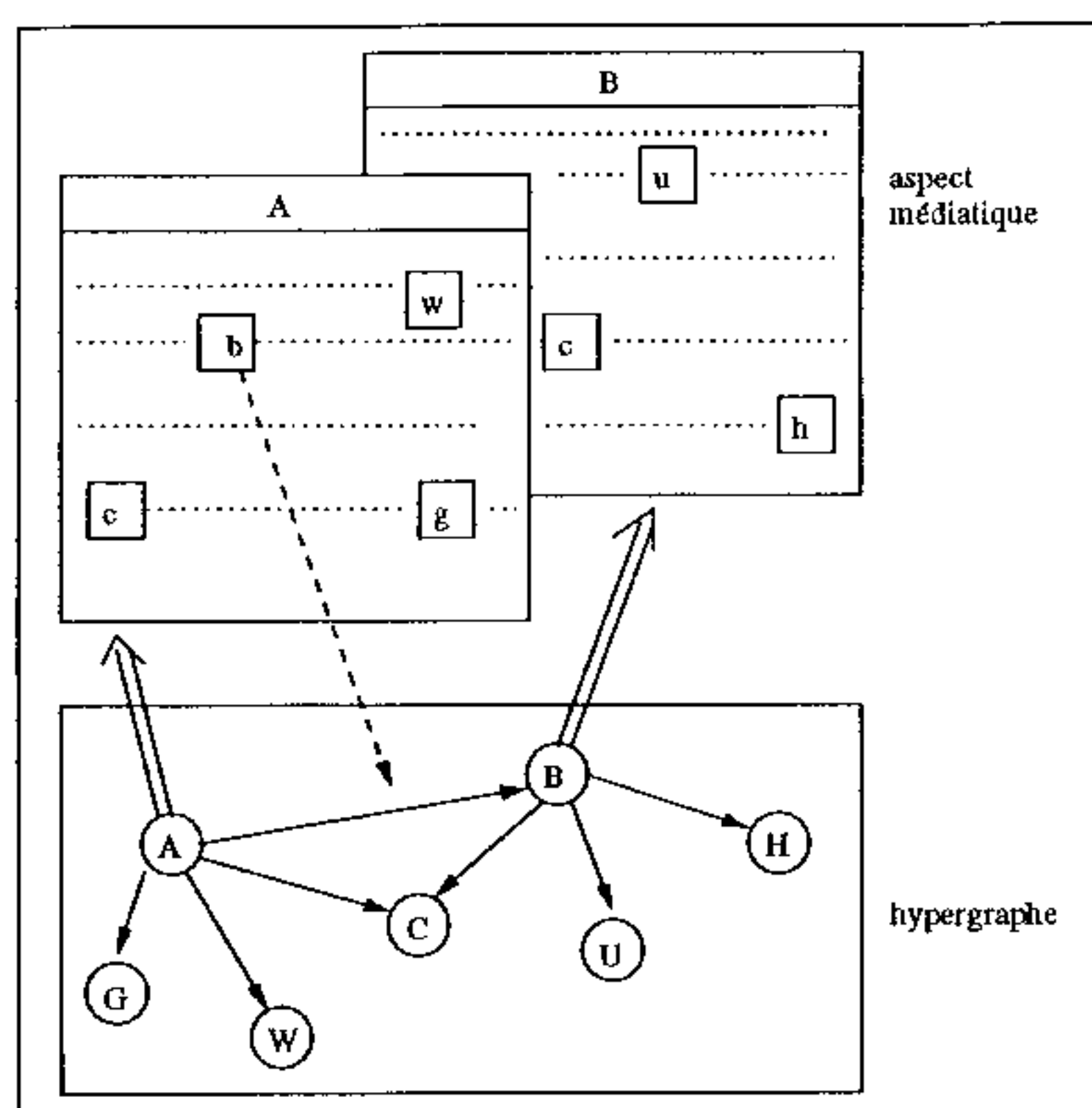


FIG. 2.2 – Exemple de réseau de nœuds et de liens.

2.3.2 Exploration du réseau

Les systèmes hypertextes sont accessibles à tout utilisateur. Cependant on distingue deux grandes classes d'utilisateurs: les lecteurs et les auteurs. Les lecteurs consultent la base d'hyperdocuments créées par les auteurs. Ces auteurs ont la charge de concevoir la base d'hyperdocuments dont les nœuds et les liens sont structurés en réseau.

Un hypertexte propose en général deux types d'exploration des nœuds du réseau: la navigation, et la recherche.

– La navigation

La navigation consiste à parcourir le réseau en utilisant les liens pour atteindre et consulter les nœuds. L'utilisateur peut choisir de parcourir l'hyperdocument de façon classique comme un livre. Pour cela il devra suivre les liens hiérarchiques. Mais il a aussi la possibilité de se laisser guider par son intuition lors du parcours de l'hyperdocument de façon non linéaire en suivant les liens non hiérarchiques ou de référence.

– La recherche

Contrairement à la navigation où l'utilisateur ne sait pas avec précision ce qu'il cherche, la recherche est caractérisée par le fait que l'utilisateur connaît l'information et il désire la localiser dans l'hyperdocument. Pour cela, il formule ses besoins sous forme de requêtes. Le système extrait alors du réseau les nœuds répondant à la question posée. Lorsque la recherche porte sur le contenu des nœuds, on parle de *recherche par le contenu*. Par contre si elle porte sur les nœuds et les liens, on parle de *recherche par*

structure :

exemple : "Rechercher tous les nœuds contenant le mot 'Acacia senegal', ancre source d'un lien référénçant un concept.

2.3.3 Quelques systèmes Hypertextes

De nombreux systèmes hypertextes ont été développés depuis le NLS (On Line System) qui a été crée au début des années 60 par D. Engelbart [FID 88] [JUL 88] [LAR 88] [SAV 88]. C'est le premier système hypertexte qui offrait la possibilité de lier des fichiers de type texte entre eux. Nous pouvons citer entre autres les systèmes suivants qui sont décrits en détail dans [DAN 90] [DES 95] [FOR 94] [MEN 94].

1. INTERMEDIA [UTT 89] a été développé à l'Université de Brown (USA) pour servir de support à l'EAO (aussi bien pour les enseignants que pour les élèves). Il est bâti sur la notion de contexte et de toiles. Son interface utilisateur propose deux mécanismes de consultation :
 - une carte globale présente l'hyperdocument dans sa totalité sous forme de graphe et permet la navigation à l'intérieur de celui-ci, et
 - une carte locale présente une vue partielle de la base centrée sur un seul document en affichant ses liens et ses plus proches voisins
2. NOTECARD [HAL 87] a été développé au centre de recherche de Xeros. Il permet de créer des bases de données très volumineuses. Il est basé sur les réseaux sémantiques en vue d'aider les utilisateurs à mettre en forme leurs idées. Les nœuds des documents appelés fiches sont spécialisés (texte, graphique, image, ...).
3. NEPTUNE [DEL 87] est un système développé par Teckronix pour servir de support à un atelier de génie logiciel (gestion des versions, annotations, ...). Il est construit autour du serveur HAM (Hypertext Abstract Machine), une machine virtuelle permettant de résoudre le problème de portabilité des hypertextes. Il maintient un historique des versions de chaque nœud (pouvant contenir du texte, des images ou du code) et permet un accès rapide à chaque version. L'interface utilisateur basé sur Smalltalk propose plusieurs outils de consultation (graph browser, document browser, version browser, ...).
4. GUIDE [BRO 91] est système hypertexte fonctionnant sous Windows pour PC. Il ne fait pas de différence entre le mode auteur et le mode lecteur. Il distingue quatre type

de boutons :

- le bouton de note : permet l’affichage des messages temporaires,
- le bouton de remplacement : permet le remplacement d’une information par une autre,
- le bouton de référence : permet d’établir un lien entre deux éléments appartenant à un même document ou à des documents différents,
- le bouton de commande : permet de déclencher l’exécution d’une autre application.

5. HYPERCARD [PIN 91] sur Macintosh est un outil de gestion et d’organisation de l’information. Il présente les informations par l’intermédiaire de fiches (card) organisées en pile (stack). Chaque fiche est constituée de champs (zone de texte ou de variables) et de boutons permettant d’établir des liens entre les fiches. A chaque bouton est associé un script écrit en Hypertalk (langage orienté objet). Il propose un historique des nœuds déjà consultés.

6. Le World Wide Web : conçu au CERN (Conseil Européen pour la Recherche Nucléaire), le World Wide Web (ou WWW ou Web) est un réseau de serveurs de documents Web, qui utilise des liens hypertextes pour accéder aux ressources d’Internet. C’est un système hypermédia dont le cadre d’exécution est celui d’une architecture de type client/serveur telle que le réseau mondial Internet [FOR 96].

- Le client appelé aussi navigateur est un logiciel de consultation et de présentation des documents Web s’exécutant sur toutes les plates formes (Mac, station UNIX, PC). Les clients les plus connus sont Mosaic, Netscape et Internet Explorer.
- Le serveur est aussi un logiciel qui est en permanence à l’écoute des requêtes formulées par les clients. Après avoir vérifié la validité des requêtes et du droit d’accès des clients (authentification), il leur transfère les pages Web.

Le dialogue entre le client et le serveur est régi par le protocole HTTP (HyperText Transfer Protocol) dont le but est de gérer le transfert des pages Web structurées en code HTML. Le client a la charge d’exécuter ce code afin d’afficher le document.

HTML (HyperText Markup Language) est le langage utilisé pour décrire la structure des documents hypertextes pour WWW au moyen d’un ensemble de balises; c’est une implémentation relativement simple du langage SGML (Standard Generalisation Markup Language) qui est devenu une norme internationale en 1986 (norme ISO 8879)

pour tenter de résoudre le problème de transport de documents électroniques sur différentes plates-formes [MAR 96].

Un document SGML [MAI 94] est constitué par une suite d'éléments; un élément est une chaîne de caractères encadrée de deux balises alphanumériques, par exemple la balise titre :

`< titre > Base de concept < /titre >`

Les balises peuvent être exprimées :

- la structure globale d'un document : découpage en chapitres, sections, paragraphes, etc.
- la structure fine : typographie, etc.

La structure générique d'un document est exprimée dans une DTD (Définition d'un Type de Document) qui décrit les éléments et les attributs, qui peuvent être utilisés pour décrire les documents d'un type donné. La structure des éléments confère une structure hiérarchique au document et les attributs annotent les éléments ou établissent entre eux des liens référentiels.

Tout comme SGML, HTML est un langage de balisage qui dissocie la structure hiérarchique (titre, paragraphe, ...) d'un document de sa présentation physique. Cette structure logique est directement codée dans le contenu textuel des documents ce qui permet à des applications (moteurs de recherche ou robots) de lancer des recherches intelligentes sur leur contenu.

Les recherches faites autour de SGML ont donné naissance à la norme HyTime (Hypermédia Time-based Structuring Language) qui étend les capacités hypertextuelles de SGML et le langage de requêtes HyQ permettant de naviguer dans un document. HyTime permet :

- de créer des liens standardisés reliant des textes et des objets (images, films, ...),
- de définir de façon dynamique la localisation des éléments liés,
- d'utiliser pleinement les capacités multimédias d'un ordinateur.

Les documents SGML comportant des liens HyTime se convertissent facilement en pages HTML afin d'être accessibles aux navigateurs.

2.3.4 Problèmes des systèmes hypertextes

Avec les systèmes hypertextes, l'utilisateur dispose d'un outil puissant capable de visualiser l'information contenue dans l'ensemble des documents à l'aide d'interfaces de consultation conviviales (multifenêtrages, outils de pointage, ...). Ces informations lui sont présentées graphiquement à travers un média. Il n'a plus besoin de connaître la structuration interne des informations avant d'effectuer une recherche. Il navigue dans l'hyperespace à la recherche d'informations de la même manière qu'opère la pensée, c'est à dire par "association d'idées".

Malgré toutes ces qualités, les systèmes hypertextes présentent quelques insuffisances liées à la conception des hyperdocuments, à la consultation des informations contenues dans les nœuds, et à la maintenance de la base.

2.3.4.1 Problèmes de conception

La conception d'hyperdocument est rendue difficile par l'absence de modèle conceptuel rigoureux de documents [DES 95]. Tout le processus de création d'un document est à la charge de l'auteur. Il doit penser au découpage des informations en granules d'informations. Puis il doit relier ces granules d'information ou nœuds par un réseau de liens intra ou inter documents de façon judicieuse afin de ne pas perdre la sémantique globale et la cohérence des informations. Pour [DES 95] ce problème de conception est lié à "la définition des objets hypertextes (nœuds et liens) pertinents vis à vis de l'application ciblée et de la manière la plus judicieuse de les médiatiser. Ce problème peut donc être résolu, par un choix adéquat de la granularité des informations, des liens qui les associent, et du choix de la médiation.

2.3.4.2 Problèmes de consultation

La force des systèmes hypertextes est de permettre au lecteur de pouvoir naviguer librement sans contrainte dans l'hyperespace par "association d'idées", en suivant les différents liens prévus par l'auteur. Cependant, dès que le réseau d'information devient important, cette grande liberté d'action peut entraîner la désorientation du lecteur, c'est à dire la perte de vue du but de sa recherche.

En effet, par manque d'une vue globale de la base d'informations, il peut accéder à des informations ayant un rapport éloigné de ses objectifs initiaux [DES 95]. Il ne sait plus d'où il vient, ni où il va. Ce problème de désorientation est dû selon [DES 95] :

- au manque de méthode de conception des hyperdocuments, garantissant une bonne

structuration logique dans l'organisation de la base,

- à la prolifération abusive des liens qui posent le problème du choix du lien pertinent,
- et au manque de spécification sémantique des liens vis à vis des informations qu'ils associent.

L'autre mode de consultation est la recherche des informations. dans ce mode on suppose que l'utilisateur a une idée précise des informations qu'il recherche afin de mieux formuler sa requête. Malheureusement ce n'est pas le cas le plus fréquemment rencontré. Outre ce problème, [DES 95] soulève trois autres problèmes qui sont :

- le temps de réponse qui peut être très long si la base d'information est volumineuse,
- la pertinence des informations trouvées pour l'utilisateur,
- et l'exhaustivité des informations trouvées i.e. le nombre d'informations pertinentes parmi toutes celles qui ont été trouvées.

2.3.4.3 Problèmes de maintenance

Dans les systèmes hypertextes, l'auteur est le seul à effectuer la mise à jour de la base d'hyperdocuments sans une aide automatique comme dans les SGBD. Cette maintenance est d'autant plus ardue que la base est volumineuse. En effet, la mise à jour pose le problème de la cohérence de la base :

- cohérence au niveau de la sémantique globale des informations lorsqu'un granule d'information a été modifié, et
- cohérence au niveau des liens qui n'aboutissent pas parce que l'ancre destination a été supprimée.

La mise à jour est rendue encore plus difficile dans les systèmes hypertextes qui incluent les marques des liens à l'intérieur du contenu informatif des documents (cas des documents HTML). Toute modification de la destination d'un lien nécessite la mise à jour conséquente des documents sources.

2.3.5 Quelques solutions

Pour résoudre les problèmes évoqués ci-dessus, les systèmes hypertextes doivent fournir des aides à la conception et à la consultation des hyperdocuments.

– *aide à la conception*

Pour la création de la base d'hyperdocuments, l'auteur a à sa disposition différents outils : en plus des opérations classiques sur le réseau (Création, Modification, Suppression), le système hypertexte peut fournir des types prédéfinis de nœuds et de liens. Il peut également fournir un mécanisme permettant de conserver et de réutiliser d'anciennes versions de nœuds et de réseaux [DAN 90]. Ces versions de nœuds et de réseaux peuvent servir à la vérification de la cohérence de la base d'hyperdocuments après une mise à jour.

– *aide à la navigation*

Pour lutter contre la désorientation, certains systèmes hypertextes proposent une carte de réseau ou des nœuds "browsers" qui permettent d'avoir une vue partielle ou globale du réseau hypertexte. Ils représentent les nœuds hypertextes sous forme d'un graphe dans lequel les nœuds sont représentés par des icônes. L'accès aux informations contenues dans un nœud se fait en sélectionnant l'icône correspondante.

D'autres systèmes utilisent la notion de chemin qui est une liste ordonnée de liens à suivre pour consulter un hyperdocument [DAN 90]. Il permet de créer un parcours personnalisé ou orienté [FOR 93] dans la base d'hyperdocuments, suivant le type de lecteur (expert, novice, ...) afin d'accéder aux informations pertinentes.

Exemple de systèmes :

- Intermedia [UTT 89] offre deux mécanismes de consultation : une carte globale qui présente la base d'hyperdocuments dans sa totalité sous forme de graphe et permet de naviguer à l'intérieur de celui-ci ; et une carte locale qui présente une vue partielle de la base centrée sur un seul document en affichant ses liens et ses proches voisins ayant des similitudes dans le contenu [DAN 90]. La gestion d'un historique des nœuds ou le marquage des pages déjà consultées offre une aide temporelle à l'utilisateur.
- Notecards [IAL 87] permet à l'utilisateur de personnaliser sa consultation par la création d'un nœud "browser". Un nœud browser contient la représentation graphique d'une partie du réseau. Il permet aussi de gérer un historique des nœuds déjà consultés.

- Neptune [DEL 87], utilise la notion de contexte pour offrir au lecteur une vue partielle et privée du réseau. Un contexte est constitué d'un ensemble de nœuds et de liens. Il permet de gérer le problème de parallélisme dans les systèmes hypertextes multi-utilisateurs.

– *aide à la recherche*

Les systèmes hypertextes doivent fournir un langage de requête accessible à n'importe quel utilisateur et permettant les deux sortes de recherche : recherche par le contenu et recherche par structure.

2.4 Systèmes Hypertextes et SGBD

Comme l'a souligné [DAN 90], un hypertexte peut être vu comme une BD débarassée de la lourdeur des SGBD. En fait les systèmes hypertextes et les SGBD sont complémentaires. En effet, ils ont un point commun qui est celui de la gestion de données. Chaque système possède un atout majeur : la navigation pour les systèmes hypertextes et la recherche pour les SGBD.

Les informations d'un hypertexte peuvent être consultées directement par navigation à travers un média. Cependant la recherche d'une information particulière n'est pas aisée. Contrairement à un SGBD dont la puissance réside dans son langage de requêtes qui facilite la recherche de données.

D'où l'intérêt de fédérer ces deux systèmes de gestion de données afin de tirer avantage de leurs atouts respectifs.

Nous pouvons aborder cette coopération selon deux approches :

- La création d'une interface de consultation hypertextuelle d'une BD.
- L'utilisation de langages relationnels dans un hypertexte.

[DES 95] résume cette coopération entre ces deux systèmes comme suit:

– **Recherche initiant la navigation**

L'expression d'une requête permet d'abord de sélectionner une partie des nœuds informatifs répondant au critère de sélection, puis la navigation est utilisée pour consulter ce sous-ensemble restreint d'information. Cette façon de procéder réduit le risque de désorientation grâce au filtrage préalable des informations.

- Navigation initiant la recherche

L'activation d'un lien lors de la navigation provoque le déclenchement de requêtes destinées à rechercher des informations utiles. Cette approche permet de créer des liens ou des documents dynamiques. Le système hypertexte MACWeb [NAN 93] est basé sur ce principe.

- Navigation dans des graphes afin de formuler une requête

Le schéma de données des BD est manipulé comme un hypergraphe. Une requête se formule de façon graphique par association d'idées. Cette requête est affinée au fur et à mesure de la navigation dans l'hypergraphe.

Nous présentons ci-dessous quelques systèmes hypertextes couplés à un SGBD.

Hyperbase

Hyperbase intervient dans un atelier de création de documents hypertextes pour auteur : SEPIA (Structured Elicitation and Processing of Ideas for Authoring) mis en œuvre dans le laboratoire IPSI [SCH 90]. Il possède un moteur hypermédia basé sur le SGBDR Sybase. Son modèle de données est orienté objet. Il reconnaît trois classes d'objets : les nœuds, les liens et les objets composites.

- Les nœuds sont des objets ayant un contenu et un historique.
- Les liens permettent de connecter les objets. On distingue trois types de lien :
 - le lien donnant accès à une information ou à un message, est représenté par un attribut dans un objet nœud,
 - le lien de dépendance entre deux objets est représenté par un objet lien
 - le lien de structuration hiérarchique est représenté par un objet composite.
- Les objets composites sont des ensembles partiellement ordonnés et définis par l'utilisateur par des références à d'autres objets. Ils peuvent être le résultat à une requête, le regroupement d'objets, un chemin dans un réseau hypertexte, un ensemble d'objets dans un espace d'activités, une structure hiérarchique d'un livre.

L'accès à ces objets est assuré par le langage Transact-SQL du SGBDR Sybase.

OHQL (Object Hypertext Query Language)

OHQL [MOR 91] [AND 93] est une interface graphique basée sur les concepts hypertextes pour l'interrogation de base de données orientée objet. Le schéma de la BD est représenté par un hypergraphe dont les nœuds représentent les classes et les liens décrivent les contraintes d'intégrité et de domaine [MEN 94] [FOR 94]. L'expression d'une requête s'effectue en sélectionnant dans l'hypergraphe les nœuds et les liens. Les informations ainsi sélectionnées constituent la requête qui sera affinée de façon progressive en naviguant dans l'hyperespace.

Hyper*SQL

[ORA 89] Hyper*SQL est une interface hypertextuelle entre le SGBD Oracle et le système hypertexte Hypercard qui est un outil de gestion et d'organisation de l'information sur les machines Macintosh d'Apple. Ce couplage Hypercard-Oracle, offre à Oracle un environnement de programmation qui est celui d'Hypertalk, langage hôte d'hypercard. Hypertalk permet d'exécuter toutes les commandes d'Oracle, tandis que Oracle même assure la gestion classique des tâches d'un SGBD. Les cartes d'hypercard sont utilisées pour présenter la structure d'une relation ou son contenu, le résultat d'une requête, ou pour formuler une requête de façon semi-graphique ne nécessitant pas forcément la connaissance de SQL.

QFE (Query Farandole Easily)

QFE est un générateur d'interfaces pour l'interrogation de bases de données à l'aide de contexte [EST 90]. L'interface hypertextuelle est Hypercard et le SGBDR est Farandole. L'interrogation de la base s'effectue au moyen de quatre types de cartes :

- carte du schéma de la base : elle permet de représenter le schéma de la base sous forme de graphe.
- carte de contexte : les contextes sont des sous-ensembles sémantiques des graphes des relations. Le schéma de la base qui est un super contexte peut être décomposé en sous-contextes qui correspondent à des vues partielles de la base.
- carte de requête : les requêtes sont créées à partir de la description des relations d'un contexte donné.
- carte réponse : elle permet de visualiser les résultats d'une requête.

Pour soumettre une requête, l'utilisateur commence par restreindre l'ensemble des informations cherchées en naviguant dans les contextes et sous-contextes. l'ensemble des nœuds et

des lien (relations, attributs, opérateurs,...) sélectionnés constituent la requête. Les réponses sont affichées dans les cartes réponses.

MULTEXT

MULTEXT (Multilingual Text Tools and Corpora) est un projet européen de développement d'un ensemble d'outils logiciels pour l'analyse et la manipulation de textes et de données multilingues [MAI 94] [MUL 96].

Ce projet s'inscrit dans un contexte de base de données textuelles fondée sur le langage de description de document SGML.

L'interrogation des documents est réalisée à l'aide du langage SgmlQL basé sur SQL. Des travaux sont en cours afin d'étendre les fonctionnalités de SgmlQL à la manipulation de documents hypertextes codés selon la norme HyTime, ainsi que le stockage des textes SGML dans des bases de données orientées objets.

Oracle Web System

Oracle Web System [ORA 96] est un environnement de développement et de mise en œuvre d'applications Web basées sur le SGBDR Oracle7. Il fournit un environnement se chargeant du développement et de la mise œuvre de A à Z de systèmes de bases de données fondées sur le web. Il facilite la gestion de volumes de données Web important, texte, graphiques, son et séquences vidéo. Il regroupe les produits suivants :

- Le Power Browser est à lui tout seul, un environnement de développement complet pour le Web, bénéficiant d'un navigateur Internet, d'un interpréteur dynamique et d'un browser SQL. Il inclut un langage de script DBML (DataBase Markup Language) orienté basic qui contient les extensions spécifiques à Oracle7 du standard HTML. Il offre des commandes de consultation et de modification des informations des BD depuis le browser.
- Le Web Server regroupe un ensemble de produits qui offrent un environnement multi-média complet pour l'édition et la mise en œuvre d'applications Web. Il permet l'accès à des procédures Oracle7 enregistrées dans un cadre orienté objet et extensible par l'utilisateur. Il permet également la création dynamique, en temps réels de pages HTML grâce à PL/SQL, l'extension langage de procédure à SQL mise au point par Oracle.
- Le Web Server option fournit les branchements pour rendre les applications Oracle7 existantes accessibles via le Web.

*O*₂ Web

Le système *O*₂ Web [O2W 96] est une interface hypertextuelle pour la manipulation des informations hypermédia d'un serveur Web basé sur le SGBDO *O*₂. Il dispose d'un langage de requête nommé OQL (Object Query Language) qui est une extension objet de SQL. L'information présentée au client est une vue des objets de la base de données.

*O*₂ Web propose les fonctionnalités suivantes :

- Le mode générique permet la création de pages HTML pour la présentation des données de la base suite à une requête.
- La personnalisation globale étend le mode générique en donnant au programmeur, la possibilité d'ajouter des entêtes et pieds de page identiques pour toutes les pages HTML. Dans ce mode, le programmeur peut réagir à certains événements tels que la connexion et la déconnexion au serveur ou la personnalisation des messages d'erreur.
- La personnalisation locale permet au programmeur de contrôler tout le processus de génération des pages HTML. Il peut définir pour chaque objet de la base une présentation personnalisée des pages HTML.
- La mise à jour de la base de données à partir d'un formulaire Web.

*O*₂ Web offre également plusieurs outils d'interfaçage avec d'autres applications (Access, C, C++, programme API, ...).

Intrabuilder

Développé par Borland, Intrabuilder [BOR 96] est un environnement de développement d'applications de gestion de données destinées au World Wide Web ou à un Intranet. L'accès aux données se fait dynamiquement grâce à des fiches visualisables à l'aide de n'importe quel navigateur Web.

Pour la gestion des données, Intrabuilder utilise les ressources du moteur de base de données de Borland (BDE), supportant d'autres bases de données relationnelles telles que Access, dBase, Paradox, Interbase, DB2, Sybase, Oracle, etc.

Il utilise comme langage de programmation une extension de JavaScript (§ 5.3.3.2 page 117) permettant de manipuler les bases de données; JavaScript étant le langage multiplateforme de développement d'applications distribuées pour le Web.

Intrabuilder offre de nombreux outils interactifs appelés experts, d'aide à la création et à la maintenance des applications Web :

- l'expert fiche aide à la création de fiches de saisie de données dans les tables,
- l'expert page d'accueil permet de créer et de gérer dynamiquement des pages HTML,
- l'expert état permet de créer des états récapitulatifs,
- l'expert table propose des modèles de champs et de tables pour les BD les plus courants : dBase, paradox, Access, ...

Tous ces composants d'une application Intrabuilder sont automatiquement sauvegardés sous forme de programmes JavaScript que l'utilisateur peut modifier sans problèmes.

2.5 Conclusion

Après avoir présenté brièvement les notions utilisées dans les BD en insistant sur leurs limites dans le traitement de nouveaux types de données complexes, structurées et volumineuses ou des documents multimédias, nous avons introduit les concepts des hypertextes; nous avons montré leurs insuffisances notamment dans la navigation et la recherche ainsi que dans la conception d'hypertextes. Nous avons ensuite présenté quelques SGBD couplés à des systèmes hypertextes qui tentent d'apporter une solution à ces problèmes en tirant parti de leurs qualités respectives : la recherche pour les SGBD et l'accès aux données de façon naturelle par association d'idées pour les systèmes hypertextes.

On peut aborder cette coopération entre SGBD et système hypertexte selon deux approches :

- la création d'une interface de consultation permettant la navigation hypertextuelle dans le schéma d'une BD; la base d'hyperdocuments est une vue de la BD. Ce type de coopération est réalisé dans les systèmes : Hyperbase, OHQL et QFE.
- l'intégration d'informations d'une BD dans les hypertextes; la BHD peut être une vue de la BD ou en être totalement indépendante. Les systèmes Hyper*SQL, Oracle Web System, O2 Web et Intrabuilder gèrent ce type de coopération.

[FOR 94] a proposé à cet effet un modèle de Gestion d'Information de Type Hypertexte (GITII) fondé sur la coopération d'une BD relationnelle et une base d'hyperdocuments

(BHD), sans modification fondamentale du modèle relationnel; la BHD nécessite l'intégration d'informations de type classique ou de type hypertexte (texte, image,...) dans les hyperdocuments. Dans ce modèle, la BD gère les informations de type hypertexte de façon relationnelle, et les hyperdocuments sont manipulés par la BHD.

Nous remarquons cependant qu'aucune de ces deux approches ne propose une solution pour traiter les informations des BD qui sont constituées d'un texte important ou d'une image de manière hypertextuelle. Elles n'abordent pas le problème de maintenance et de vérification de cohérence de la BHD. En ce sens que la BD ne peut gérer les hyperdocuments (suppression ou ajout d'un nœud ou d'un lien, ...).

Or le modèle de BdC que nous proposons, offre non seulement à l'utilisateur une interface hypertextuelle d'accès à la base de concepts, mais surtout un modèle de structuration et de représentation des concepts d'un domaine; il étend le modèle GITH par la possibilité: (i) de stocker dans une base de données le contenu informatif et la structure logique d'un hyperdocument, (ii) de générer dynamiquement l'hyperdocument par un processus d'hypérisation.

Pour la construction de ce modèle, nous nous inspirons du modèle des BD textuelles fondées sur le langage de description de documents SGML, qui dissocie la structure logique d'un texte de son contenu. De même, dans la BdC les hyperdocuments sont stockés non pas sous forme d'un hypertexte déjà créé, mais plutôt sous forme d'un hypertexte en deux composantes (contenu informatif et structure logique).

Cette approche facilite non seulement la maintenance de la BHD notamment :

- dans la vérification de la cohérence sémantique des hyperdocuments. En effet, contrairement à la mise à jour manuelle et fastidieuse dans les hypertextes classiques (contenu informatif et structure logique au sein du même document), la mise à jour des hyperdocuments se fait dans la BD à travers ces deux composantes. Un processus d'hypérisation répercute automatiquement cette mise à jour sur la BHD.
- dans la vérification de la cohérence au niveau des liens qui n'aboutissent pas parce que le système ne trouve pas le nœud destination. Pour déterminer ces liens, nous recherchons dans un premier temps les liens qui figurent dans le code source des hyperdocuments, ensuite nous les comparons avec ceux qui ont été stockés dans la BdC.

mais aussi, elle permet de créer une vue de la base d'hyperdocuments correspondant au choix de l'utilisateur des informations terminologiques ou de caractères définitoires qu'il maîtrise le mieux et qu'il souhaite manipuler.

Dans le chapitre suivant, le modèle de BdC que nous proposons tente de répondre à toutes ces préoccupations.

Chapitre 3

Le modèle de la base de concepts

3.1 Introduction

Dans une BdC est défini un certain nombre de propriétés et d'opérations de manipulation des concepts. Dans cette partie, nous présentons ces éléments et nous donnons la représentation conceptuelle d'une BdC à l'aide du modèle Entite/Association.

Après avoir présenté les logiques terminologiques qui sont un formalisme de représentation de connaissances, nous décrivons la notion de concept utilisé dans une BdC de façon formelle; nous terminons par la description des éléments du schéma conceptuel de la BdC.

3.2 Logiques terminologiques

Les logiques terminologiques (LT) ou logiques de description de concepts ont été développées dans le domaine de recherche en intelligence artificielle [BOR 95]. Elle permettent de représenter la connaissance d'un point de vue traitement informatique et non cognitif (cas des graphes conceptuels) [OUN 95]. Ces LT sont implémentées dans beaucoup de systèmes dont le système BACK [HOP 93]. Elles utilisent un formalisme clair et des mécanismes bien définis.

3.2.1 Le formalisme

Contrairement aux logiques standards (logique des prédicats du premier ordre) qui manipulent des formules, les LT sont basées sur la notion de termes et d'opérateurs de construction de termes. Elles utilisent la sémantique dénotationnelle i.e. une interprétation ensembliste

des termes.

3.2.1.1 Les termes

Il existe deux sortes de termes : les concepts et les rôles.

- Un concept décrit une collection d'individus semblables d'un domaine à modéliser (analogie avec la notion de classe d'objets dans les modèles à objets). On distingue deux sortes de concepts : les concepts primitifs et les concepts définis.
 - un concept primitif est un concept partiellement déterminé par sa description qui est interprétée comme un ensemble de conditions nécessaires. Ils correspondent à des entités du monde réel qui ne peuvent être décrites de façon exhaustive telles que *Personne*, *Voiture* et *Plante*.
 - un concept défini est un concept entièrement déterminé par sa description qui est interprétée comme un ensemble de conditions nécessaires et suffisantes que doivent satisfaire ses instances. Exemple : *Adan* et *Eve*, instances respectivement des concepts *Homme* et *Femme* (cf. figure 1.3 page 13).
- Un rôle représente une relation binaire entre des concepts. Il est utilisé pour spécifier les propriétés de leurs instances.

Dans la figure 1.4 page 14, "utilise" est un rôle qui relie le concept *Personne* au concept *Voiture*.

3.2.1.2 Les opérateurs

Chaque logique terminologique propose un ensemble d'opérateurs de formation de termes (concept/rôle). Ces opérateurs sont :

- **and** : création d'un terme par la conjonction de concepts
- **or** : création d'un terme par la disjonction de concepts
- **not** : négation d'un terme
- **all** : restriction de rôle (précise le type de valeur des instances d'un rôle)
- **atleast** : restriction du nombre d'instance d'un rôle (valeur minimale)
- **atmost** : restriction du nombre d'instance d'un rôle (valeur maximale)

- `inv` : formation de rôle inverse
- `range` : restriction du type du rôle
- `:<` : introduction de termes primitifs (opérateur de subsumption)
- `:=` : introduction de terme définis

Parmi ces opérateurs, le plus important est l'opérateur de subsumption qui permet d'organiser par classification, les termes d'une terminologie en une hiérarchie de concepts (généralisation/spécialisation).

La subsumption permet aussi d'inférer de nouvelles relations d'héritage entre les concepts. Ces inférences basées sur la subsumption sont appelées raisonnements taxonomiques.

Un concept C_i subsume (ou est plus général que) un concept C_j signifie qu'il est possible d'obtenir la description de C_j à partir de celle de C_i , soit en ajoutant des contraintes, soit en restreignant les contraintes déjà présentes.

Exemple : le concept "Personne" subsume le concept "Homme", i.e. la description de Homme est contenue dans celle de Personne. Un homme est une personne de sexe masculin.

Dans le système BACK, la connaissance est divisée en deux composantes : la connaissance terminologique et la connaissance assertionnelle.

- La connaissance terminologique permet de définir les concepts qui peuvent être utilisés dans les assertions (base de faits). Cette connaissance est contenue dans la TBOX (Terminological Box).

Exemple : Définition des termes dans le système BACK dans la notation BNF (nous rappelons sa syntaxe à l'annexe E page 181).

```

<term_descrip> ::= <prim_concept> | <def_concept>
<prim_concept> ::= <term_name> "<" <term>
<def_concept> ::= <term_name> "==" <term>
<term> ::= <conceptual_type> | <role>
<conceptual_type> ::= <concept> | "number" | "string"
<concept> ::= <concept_name>
               | "anything"

```

```

| "nothing"
| "not" <concept>
| <concept> "and" <concept>
| <concept> "or" <concept>
| <concept> "and" <concept>
| "all" "(" <role> "," <conceptual_type> ")"
| "atleast" "(" "integer" "," <role> ")"
| "atmost" "(" "integer" "," <role> ")"
| "exactly" "(" "integer" "," <role> ")"
<role> ::= <role_name>
| "domain" "(" <concept> ")"
| "range" "(" <conceptual_type> ")"
| <role> "and" <role>
| "not" <role>
| "inv" <role>

```

- La connaissance assertionnelle définit les règles associées à la base de connaissances (base de règles). Ces connaissances sont contenues dans la ABOX (Assertional Box).

L'opérateur $\Rightarrow$ définit une règle SI-ALORS permettant de décrire les contraintes d'intégrité.

En utilisant le modèle BACK on peut définir l'assertion "*Toute personne homme ou femme est un animal intelligent*" de la façon suivante :

- Définition du concept primitif racine :
 - (1) animal :< anything.
- Définition des rôles :
 - (2) qualité :< domain(animal) and range(string).
 - (3) nom :< domain(anything) and range(string).
 - (4) sex := attribut_domain[masculin, féminin].
 - (5) sexe :< domain(anything) and range(sex).
- Définition des concepts primitifs :
 - (6) personne :< animal and all(qualité: 'intelligent')
 - (7) homme :< personne and all(sexe: masculin)
 - (8) femme :< personne and all(sexe: féminin)

- Définition des concepts définis :
 - (9) `Adan := homme and all(nom: 'Adan')`
 - (10) `Eve := femme and all(nom: 'Eve')`

Dans l'exemple ci-dessus, nous avons commencé par définir la racine de l'arbre conceptuel du domaine (ligne 1); cette racine est le concept `animal` qui subsume le concept `personne` (ligne 6) dont l'arbre conceptuel a été présentée à la page 13. Puis nous avons décrit les propriétés des instances des concepts primitifs (ligne 2 à 5). La ligne 6 traduit le fait qu' "une personne est un animal intelligent"; les lignes 9 et 10 indiquent que les concepts `Adan` et `Eve` sont respectivement des instances des concepts `homme` et `femme`, eux mêmes subsumés par le concept `personne` (ligne 7 et 8).

3.2.2 Comparaison entre logique terminologique et base de concepts

L'avantage des LT est que leur formalisme est bien défini; elles permettent non seulement de représenter la connaissance d'un domaine particulier par leurs opérateurs de construction de termes, mais elles infèrent aussi d'autres faits en raisonnant sur cette connaissance grâce à la subsumption. Cependant, elles présentent quelques limites [RUD 96] lorsque ces connaissances deviennent complexes et structurées (relations d'association ou d'agrégation, hypertexte, ...).

Le modèle de BdC que nous proposons ci-dessous, s'inspire des travaux effectués en LT; on y retrouve les mêmes notions de concepts et de rôle (à rapprocher de la notion caractère définitoire). Quant à la subsumption, elle est mise en œuvre par les informations terminologiques concept générique et sous-concept.

Cependant, ces deux modèles de représentation de connaissance divergent sur leur utilisation : les LT privilégient le traitement informatique des concepts (inférence et classification), alors que la BdC insiste plus sur la découverte des relations sémantiques qui existent entre les concepts d'un domaine; ces relations sont modélisées dans une base de données et sont accessibles le plus naturellement possible, par association d'idées comme dans un hypertexte.

3.3 Représentation formelle d'un concept

Dans cette partie, nous donnons une description d'un point de vue formel d'un concept quant à sa structuration hiérarchique, ses relations avec d'autres concepts, ses propriétés, et

les opérations qui peuvent lui être appliquées.

3.3.1 Définition

Soit D un domaine quelconque :
 $D = \{C_1, C_2, \dots, C_n\}$
où les C_i avec $i=1,2, \dots, n$, désignent des concepts.

On distingue deux sortes de concepts : les concepts élémentaires et les concepts complexes.

– **Les concepts élémentaires** : ce sont des concepts dont la compréhension ne fait appel à aucun autre concept; ces concepts sont les feuilles terminales de l’arbre conceptuel du domaine (concepts définis en LT).

Exemple : les concepts Adan est un concept élémentaire du concept Homme (Fig. 1.3 page 13).

– **Les concepts complexes** : ce sont des concepts qui peuvent être décomposés en sous-concepts (concepts primitif en LT). Ces sous-concepts peuvent être élémentaires ou complexes. Un sous-concept est une instantiation d’un concept complexe.

Exemple : le concept Personne à pour concepts fils les concepts Homme et Femme (Fig. 1.3 page 13).

Lorsqu’un concept est complexe, il est représenté par un arbre de concepts (ou arbre conceptuel du domaine) dont les feuilles terminales sont des concepts élémentaires.

Donc le domaine D contient non seulement les concepts C_i mais aussi leurs éventuelles descendances (Fig. 3.1). La racine de tout arbre de concept est le domaine D lui-même.

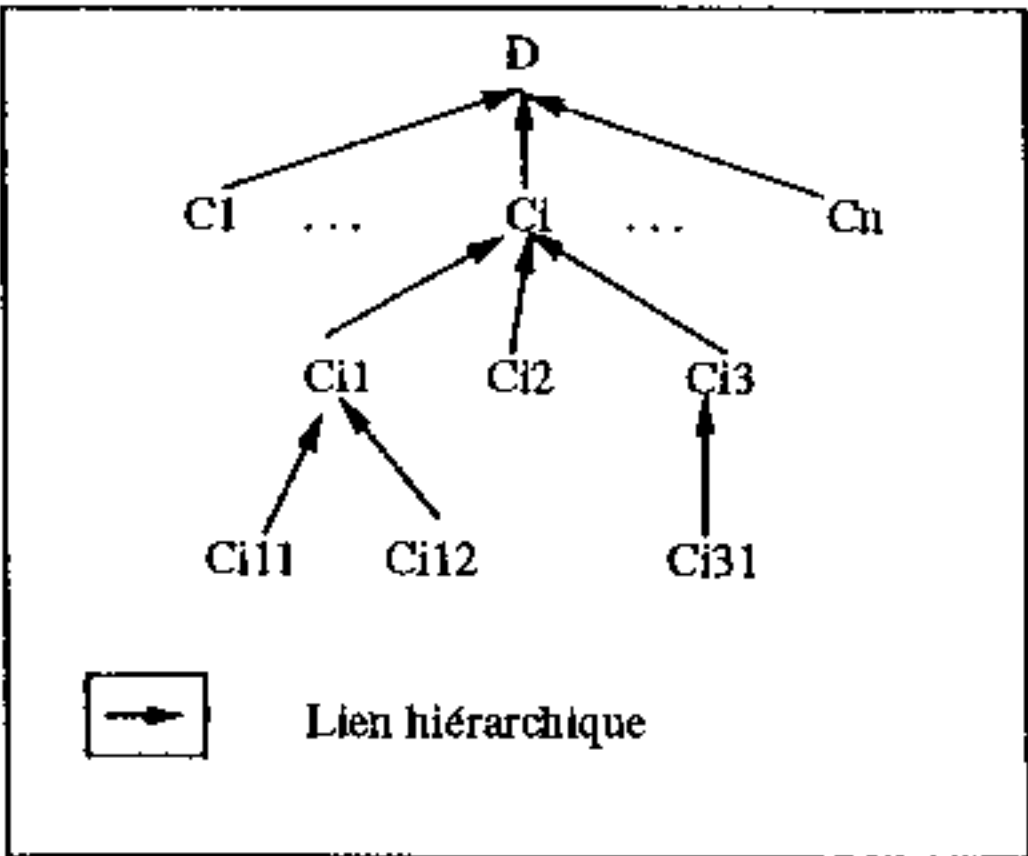


FIG. 3.1 – Arbre de concepts d’un domaine D.

Il peut exister des liens entre concepts sémantiquement voisins. On définit ainsi deux

sortes de liens :

- les liens hiérarchiques: entre un concept C_i et sa descendance modélisés par les informations terminologiques Concept générique et Sous-concept;
- les liens de référence: entre deux concepts C_i et C_j liés par une relation sémantique (Fig. 3.2) modélisés par les informations terminologiques de type relation conceptuelle (telle que Concept lié).

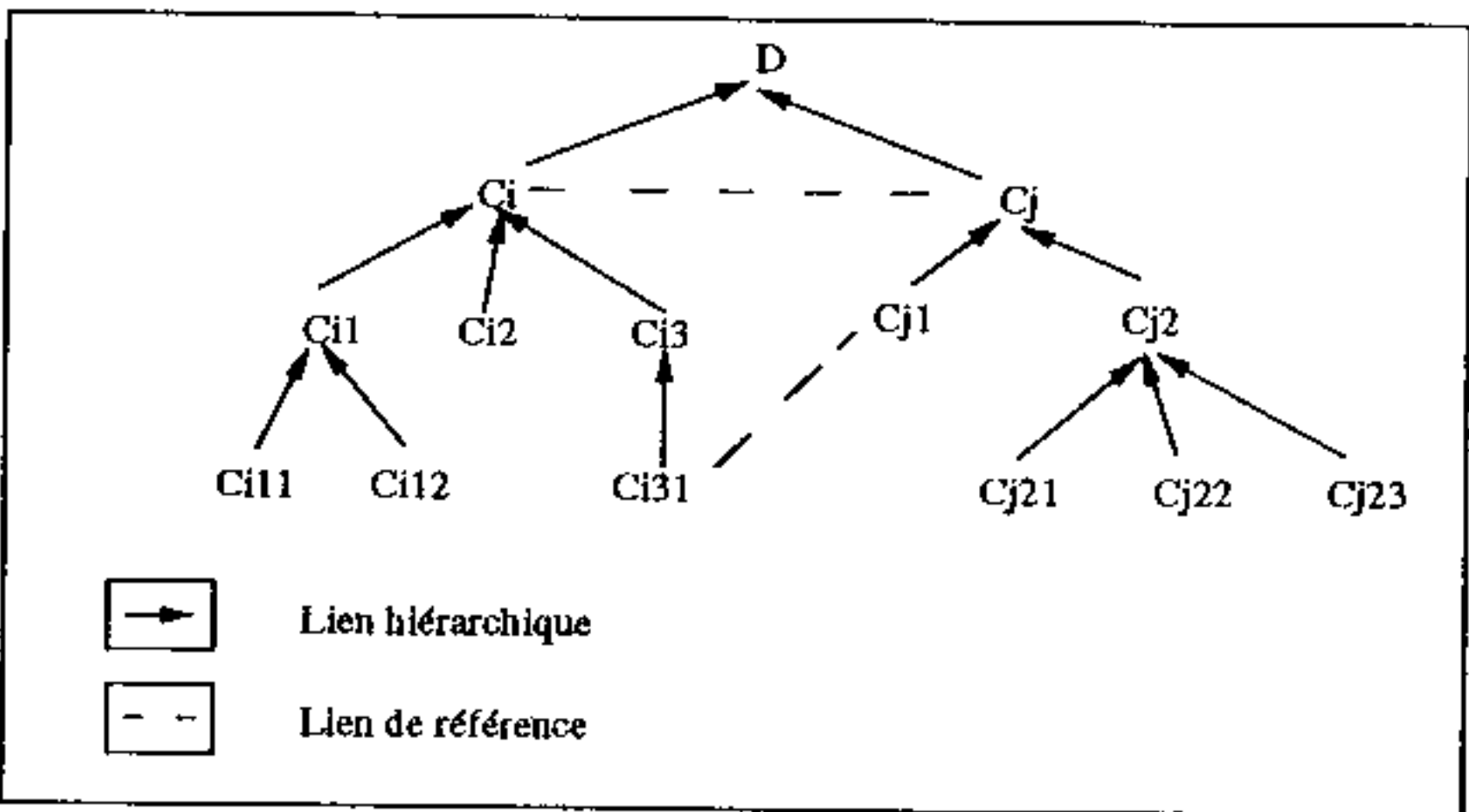


FIG. 3.2 – Relation entre deux concepts liés d'un domaine D .

3.3.2 Propriétés et opérateurs d'un concept

Dans la BdC est défini un ensemble d'opérateurs de manipulation des concepts et de leurs propriétés. Les informations terminologiques et les caractères définitoires constituent les propriétés des concepts; elle sont automatiquement héritées par leurs descendances.

Cependant, contrairement aux LT, ces opérateurs ne permettent pas d'inférer d'autres concepts.

Soient :

- C_i un concept,
- T_i l'ensemble de ses propriétés terminologiques,
- D_i l'ensemble de ses caractères définitoires et
- $\mathcal{O}$ l'ensemble des opérateurs définis sur l'ensemble de concepts.

Le concept C_i est le doublet (T_i, D_i) représenté comme suit :

$$C_i = \langle T_i, D_i \rangle$$

L'ensemble des concepts C est l'union des concepts C_i ainsi définis :

$$C = \cup_{i=1,n} C_i$$

Une BdC est donc représentée par le doublet (C, O) constitué de l'ensemble des concepts C et de l'ensemble O des opérateurs définis sur la BdC :

$$\text{BdC} = \langle C, O \rangle$$

3.3.2.1 Les propriétés

Les informations terminologiques et les caractères définitoires constituent les propriétés des concepts qui sont données ci-dessous :

- Terme
- Langue
- Synonymes
- Homonymes
- Concepts génériques
- Sous-concepts
- Définition
- Image
- Contexte
- Concepts liés
- Caractères définitoires

3.3.2.2 Les opérateurs

Ils permettent de manipuler les concepts et leurs propriétés. On a défini les opérateurs suivants :

- `ajouter_concept(c)` : permet d'ajouter un nouveau concept `c` à la BdC en précisant les valeurs de ses propriétés. Ce concept s'insère automatiquement dans la hiérarchie des concepts.
- `supprimer_concept(c)` : permet de supprimer le concept `c` de la BdC. Cette suppression entraîne la suppression des instances de ce concept ainsi que celles de ses concepts fils. La hiérarchie des concepts est ainsi mise à jour.
- `rechercher_concept(c, critère)` : permet de trouver le concept `c` selon un critère portant sur la valeur de ses propriétés.
- `modifier_concept(c, x, valeur)` : permet de remplacer la valeur de la propriété `x` du concept `c` par une autre valeur.
- `afficher_concept(c)` : permet d'afficher la valeur des propriétés d'un concept.
- `comparer_concept(c1, c2)` : permet de déterminer si deux concepts sont équivalents ou non. Cette équivalence est basée sur la notion de synonymie.

3.4 Le modèle de données de la base de concepts

Comme nous l'avons présenté aux § 1.2.2.1 et 1.2.2.2, Une base de concepts manipule essentiellement deux types d'informations :

- les informations terminologiques qui sont de nature textuelle ou graphique permettent de prendre connaissance de la sémantique des concepts, et
- les caractères définitoires qu'on peut considérer comme les attributs d'une relation ou les propriétés d'un objet, sont utilisés dans la manipulation des concepts (comparaison, recherche, etc.).

3.4.1 Classification des informations terminologiques

On peut classer les informations terminologiques (Terme, Langue, Définition, Contexte, Image, Concept générique, Sous-concept, Concept lié, Synonyme, Homonyme) que nous

avons présentées au § 1.2.2.1 en trois groupes :

- les informations de type attribut,
 - les informations de type hypertexte et
 - les informations de type relation conceptuelle.
- a) Les informations de **type attribut simple (ITA)** : elles représentent les attributs simples de la BdC.
On en distingue deux types : Terme et Langue.
- b) Les informations de **type hypertexte (ITHT)** : ce sont les informations terminologiques qui sont traitées de façon hypertexte parce qu'elles renferment un texte volumineux ou un graphique.
On en distingue trois types : Définition, Contexte et Image.

La définition et le contexte d'un concept sont de nature textuelle; ils peuvent faire référence à des informations terminologiques ou à des caractères définitoires d'autres concepts. Ces informations terminologiques ou ces caractères définitoires sont marqués comme des ancres de liens terminologiques ou ancres de liens définitoires.

Par exemple la définition et le contexte du concept Voiture (cf. § 1.2.2.1 page 11), font référence à des concepts du domaine (par exemple le concept Moteur).

Comme nous l'avons souligné au paragraphe 1.3.2, ces liens sont modélisés et leurs valeurs sont stockées dans la BdC séparément du texte de la définition.

Il est important que la mise à jour des hyperdocuments associés à ces informations terminologiques, et la vérification de la cohérence de la base de concepts puisse se faire de manière simple et facile. Pour atteindre cet objectif, nous avons opté pour une représentation séparée du contenu informatif et des liens de ces informations terminologiques de type hypertexte (cf. § 3.4.4).

- c) Les informations de **type relation conceptuelle (ITRC)** : elles assurent la liaison entre les concepts du domaine liés par une relation sémantique ou d'association. Ce sont les informations terminologiques suivantes : Concepts génériques, Sous-concepts, Concepts liés, Synonyme, et Homonyme .

Exemple : la relation d'association assure qui lie les concepts Personne et Voiture (Fig. 1.4).

3.4.2 Classification des caractères définitoires

Les caractères définitoires (CD) qu'on peut considérer comme les attributs d'une relation ou les propriétés d'un objet sont classés en deux catégories :

- les caractères définitoires simples (CDS) et
- les caractères définitoires complexes (CDC).

Pour illustrer ces différents types de CD nous utiliserons les concepts suivants et leurs caractères définitoires :

Carrosserie : nombre de portes, couleur

Roue : diamètre, pression

Chassis : numéro

Moteur : nombre de Km, puissance

Constructeur : nom, adresse, téléphone

Garage : nom, adresse, téléphone

Voiture : numéro, marque, modèle, Carrosserie, Roue, Moteur, Constructeur, Garage

- a) Les **caractères définitoires simples** : ils peuvent être utilisés comme attributs simples d'une relation.

Exemple : tous les caractères définitoires des concepts Carrosserie, Roue, Chassis et Moteur ainsi que les caractères définitoires numéro, marque et modèle du concept Voiture sont des CDS.

- b) Les **caractères définitoires complexes** : ils sont constitués de plusieurs CDS et/ou de plusieurs CDC. Ce sont des attributs composés.

On distingue deux sortes de CDC :

- les CDC de type concept qui sont des concepts du domaine (CDTC) :
par exemple les concepts Carrosserie, Roue, Chassis et Moteur sont des CDC du concept Voiture ; ce sont également des concepts appartenant au domaine étudié, ils sont appelés (avec leurs instances), caractères définitoires de type concept du concept Voiture (et de ses instances).
- les CDC de type relation qui ne sont pas des concepts pertinents du domaine (CDTR) :
par exemple les concepts Constructeur et Garage sont des CDC du concept Voiture mais ce ne sont pas des concepts appartenant au domaine étudié, ils sont appelés

(avec leurs instances), caractères définitoires de type relation du concept Voiture (et de ses instances).

Les règles de détermination des ces caractères définitoires sont données aux § 3.4.3.

Nous modélisons cette classification des informations terminologiques et des caractères définitoires d'un concept en notation BNF comme suit :

<code><Concept></code>	<code>::= "(" {<IT>} "," {<CD>} ")"</code>
<code><IT></code>	<code>::= <ITA> <ITHT> <ITRC></code>
<code><ITA></code>	<code>::= "Terme" "Langue"</code>
<code><ITHT></code>	<code>::= "Définition" "Contexte" "Image"</code>
<code><ITRC></code>	<code>::= "Concept générique" "Concept lié" "Synonyme" "Homonyme"</code>
<code><CD></code>	<code>::= "CDS" "CDC"</code>
<code><CDC></code>	<code>::= " CDTC" "CDTR"</code>

3.4.3 Instanciation des propriétés des concepts

Nous proposons trois types de règles permettant de déterminer efficacement les valeurs des informations terminologiques et celles des caractères définitoires d'un concept.

3.4.3.1 Instanciation des informations terminologiques

La détermination des valeurs des informations terminologiques est aisée, sauf pour celles relatives à l'information terminologique Concept lié que nous formalisons selon la règle 1 qui est donnée ci-dessous.

Le terme association que nous employons ici a la même sémantique que celui du modèle Entité/Association.

- Règle 1 : détermination de deux concepts liés par une relation d'association :

«Soit A une association entre C₁ et C₂ deux concepts du domaine étudié; si les cardinalités des couples (C₁, A) et (C₂, A) sont différentes de (0, 1) ou (1, 1) alors les deux concepts sont liés. A est le nom de la relation conceptuelle entre C₁ et C₂ l'antonyme de A, est le nom de la relation conceptuelle entre C₂ et C₁».

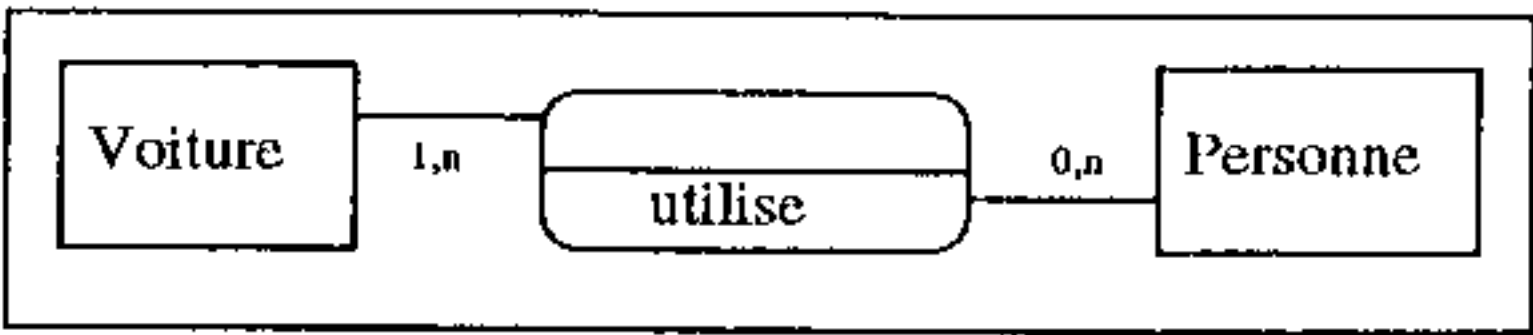


FIG. 3.3 – Association entre deux concepts liés.

Exemple : nous supposons que Personne et Voiture appartiennent au même domaine.

La figure 3.3 décrit deux concepts Personne et Voiture liés par une relation conceptuelle de type association. Une personne peut utiliser plusieurs voitures, et une voiture peut être utilisée par plusieurs personnes. D’où les cardinalités (1,n) et (0,n).

3.4.3.2 Détermination des caractères définitoires

Les caractères définitoires sont les propriétés des objets des concepts. Si la détermination des caractères définitoires simples est aisée, il n’en est pas de même pour les caractères définitoires de type complexe. Les deux règles suivantes aident à leur détermination.

- Règle 2 : détermination des caractères définitoires de type concept :

«Soit A une association entre C_1 et C_2 deux concepts du domaine étudié; si la cardinalité du couple (C_1, A) est égale à $(0, 1)$ ou $(1, 1)$, le concept C_2 est un caractère définitoire de type concept du concept C_1 ».

Exemple : nous supposons que Voiture et Moteur appartiennent au même domaine.

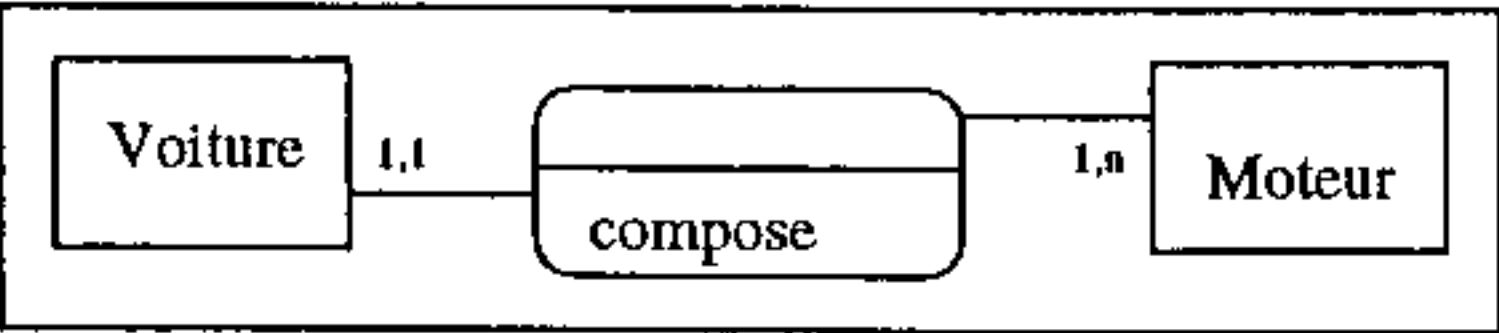


FIG. 3.4 – Caractères définitoires de type concept.

La figure 3.4 décrit le concept Moteur qui est un caractère définitoire de type concept du concept Voiture. Une voiture peut est composée d’un seul type de moteur, et un moteur peut caractériser plusieurs voiture. D’où les cardinalités (1,1) et (1,n).

- Règle 3 : détermination des caractères définitoires type relation :

«Soit A une association entre C_1 et C_2 deux concepts. Seul C_1 est un concept du domaine étudié; quelque soit la cardinalité des deux couples (C_1, A) et (C_2, A) , le concept C_2 est une relation qui est un caractère définitoire de type relation du concept C_1 ».

Exemple : nous supposons que les concepts Constructeur et Garage n'appartiennent pas au domaine.

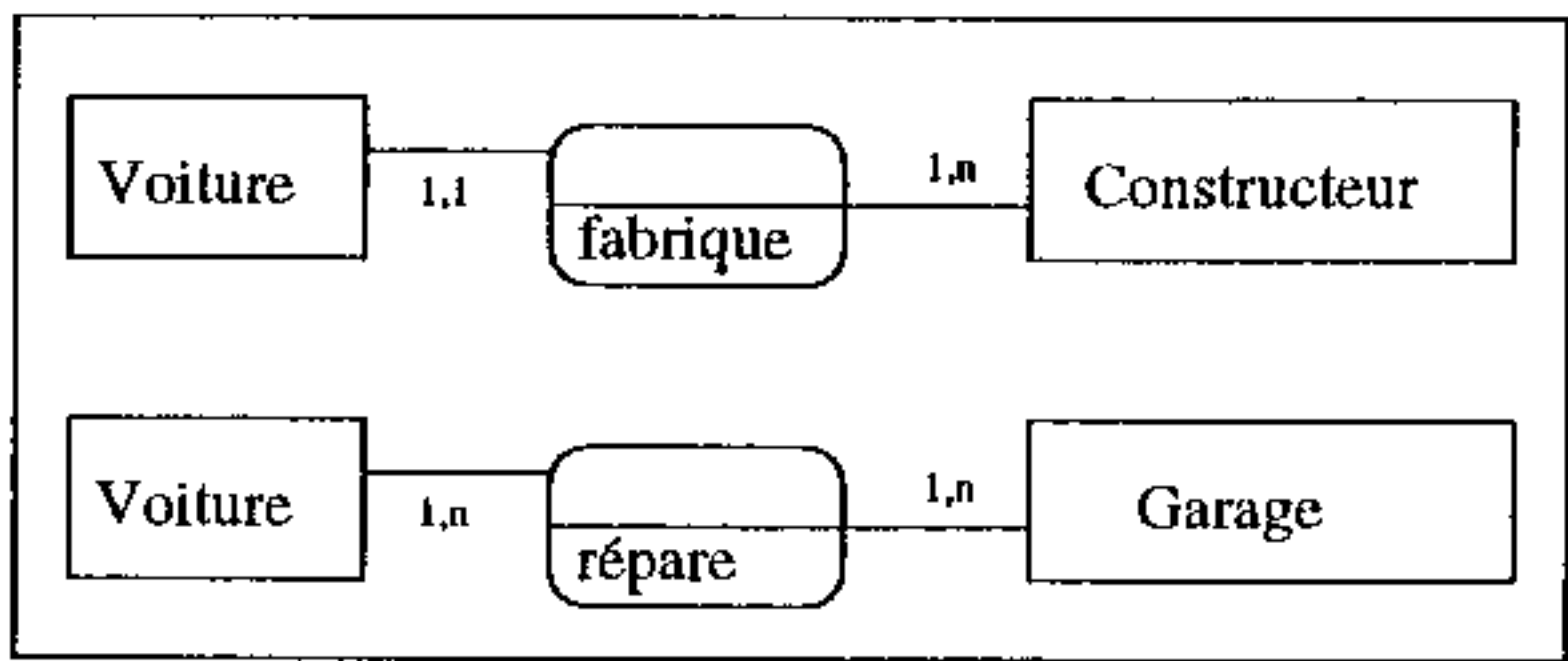


FIG. 3.5 – Caractères définitoires de type relation.

La figure 3.5 décrit les concepts Constructeur et Garage qui n'appartiennent pas au domaine étudié (enregistrement de voitures) et qui sont des caractères définitoires de type relation du concept Voiture.

3.4.4 Le modèle conceptuel de la BdC

Dans cette partie, nous présentons le modèle conceptuel de données de la BdC à l'aide du formalisme du modèle entité/association.

Le schéma conceptuel de la figure ?? montre les différentes relations qui existent entre les éléments d'une base de concepts. Il est basé sur la classification des informations terminologiques et des caractères définitoires que nous venons de présenter.

Il comporte deux parties :

- la partie de droite modélise le contenu informatif de la BdC et la
- partie de gauche, l'aspect logique de la base d'hyperdocuments de la BdC.

Dans la figure ??, les entités sont représentées par des rectangles et les associations par des rectangles aux angles arrondis. Ces rectangles sont divisés en deux parties : la partie du haut représente le nom de l'entité ou de l'association, et la partie du bas leurs attributs.

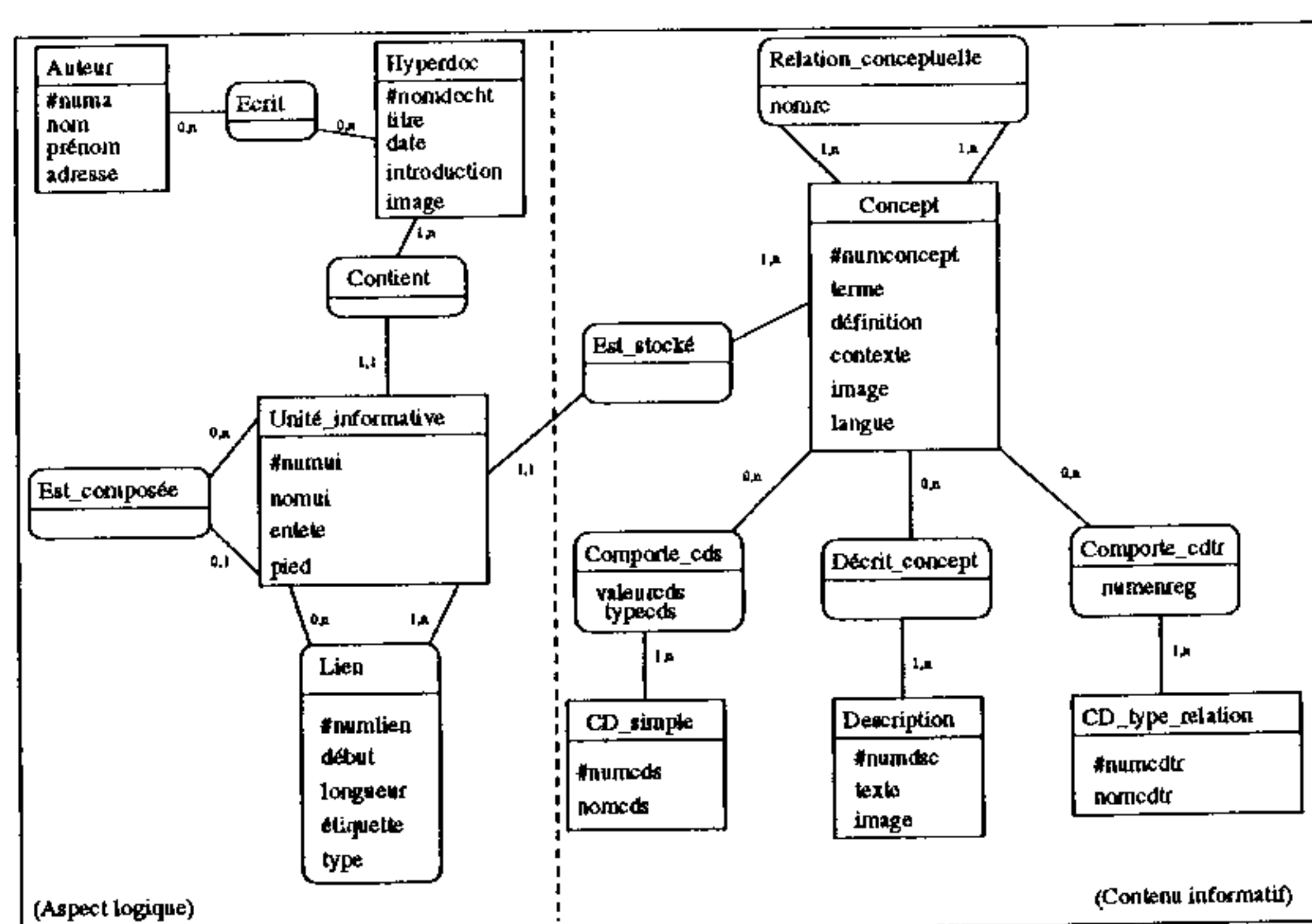


FIG. 3.6 – Schéma conceptuel de donnée de la BdC.

3.4.4.1 Aspect informatif de la BdC

Cet aspect permet de modéliser le contenu informatif de la BdC qui est constitué par les informations terminologiques et les caractères définitoires.

Les informations terminologiques sont modélisées par l'entité Concept, et les caractères définitoires, par les entités CD_Simples et CD_type_relation.

1. L'entité Concepts :

Elle modélise tous les concepts de la BdC; ses attributs sont caractérisés par les informations terminologiques suivantes :

- *numconcept* : de type numérique, clé de l'entité.
- *terme* : de type chaîne de caractères, sa valeur est le nom du concept.
- *définition* : de type chaîne de caractères, sa valeur est le texte qui donne la définition du concept.
- *contexte* : de type chaîne de caractères, sa valeur est le texte qui indique le contexte d'utilisation du concept.
- *image* : de type chaîne de caractères, sa valeur est le nom d'un fichier contenant la représentation graphique de l'image d'un concept.
- *langue* : de type chaîne de caractères, sa valeur est la langue qui est utilisée dans le domaine pour décrire les concepts.

L'entité Concept est liée dans plusieurs relations conceptuelles avec d'autres concepts,

elle comporte plusieurs caractères définitoires simples ou complexes et elle peut être stockée dans plusieurs unités informatives.

2. L'association *Relation_conceptuelle*:

Elle modélise les informations terminologiques de type relation conceptuelle; ses attributs sont:

- *nomrc*: de type chaîne de caractères; sa valeur indique la sémantique de la relation. Il peut avoir comme valeur: les informations terminologiques de type relation conceptuelle (concept lié, homonymes, synonyme, concept générique, sous-concept) et le caractère définitoire de type concept (CDTC).

3. L'entité *CD_simple*:

Elle modélise les caractères définitoires simples, on peut l'assimiler à un dictionnaire de données; ses attributs sont:

- *numcds*: de type numérique, clé de l'entité.
- *nomcds*: de type chaîne de caractères, sa valeur est le nom d'un Caractère définitoire simple.

Un caractère définitoire simple peut appartenir à plusieurs concepts.

4. L'association *Comporte_cds*:

Elle modélise l'ensemble des valeurs des caractères définitoires simples; son attribut:

- *valeurcds*: de type chaîne de caractères, elle permet de déterminer la valeur réelle du caractère définitoire simple selon son type.
- *typecds*: de type chaîne de caractères, sa valeur indique le type (numérique ou texte) d'un caractère définitoire simple; il permet d'effectuer des opérations mathématiques sur les concepts.

5. L'entité *CD_type_relation*:

Elle modélise l'ensemble des caractères définitoires de type relation; ses attributs sont:

- *numcdtr*: de type numérique, clé de l'entité.
- *nomcdtr*: de type chaîne de caractères, sa valeur est le nom d'une relation dont les tuples constituent des caractères définitoires complexes d'un concept.

6. L'association *Comporte_cdtr*:

Elle modélise l'ensemble des valeurs des caractères définitoires de type relation d'un

concept; son attribut est :

- *numenreg*: de type numérique, numéro de l'enregistrement qui est caractère définitoire de type relation d'un concept.

7. L'entité Description:

Elle modélise l'ensemble des informations de description d'un concept; ses attributs sont :

- *numdesc*: de type numérique, clé de l'entité.
- *texte*: de type chaîne de caractères, sa valeur est le texte de description
- *image*: de type chaîne de caractères, sa valeur est le nom d'un fichier image de description.

3.4.4.2 Aspect logique de la base d'hyperdocuments

Cet aspect modélise la structure logique de la base d'hyperdocuments (BHD) qui permet d'exploiter de manière hypertextuelle la BdC.

Un hyperdocument est constitué d'unités informatives qui sont reliées les unes aux autres par deux types de liens : les liens hiérarchiques modélisés par l'entité Unité_informative et les liens de référence modélisés par l'association Lien (Fig. 3.7).

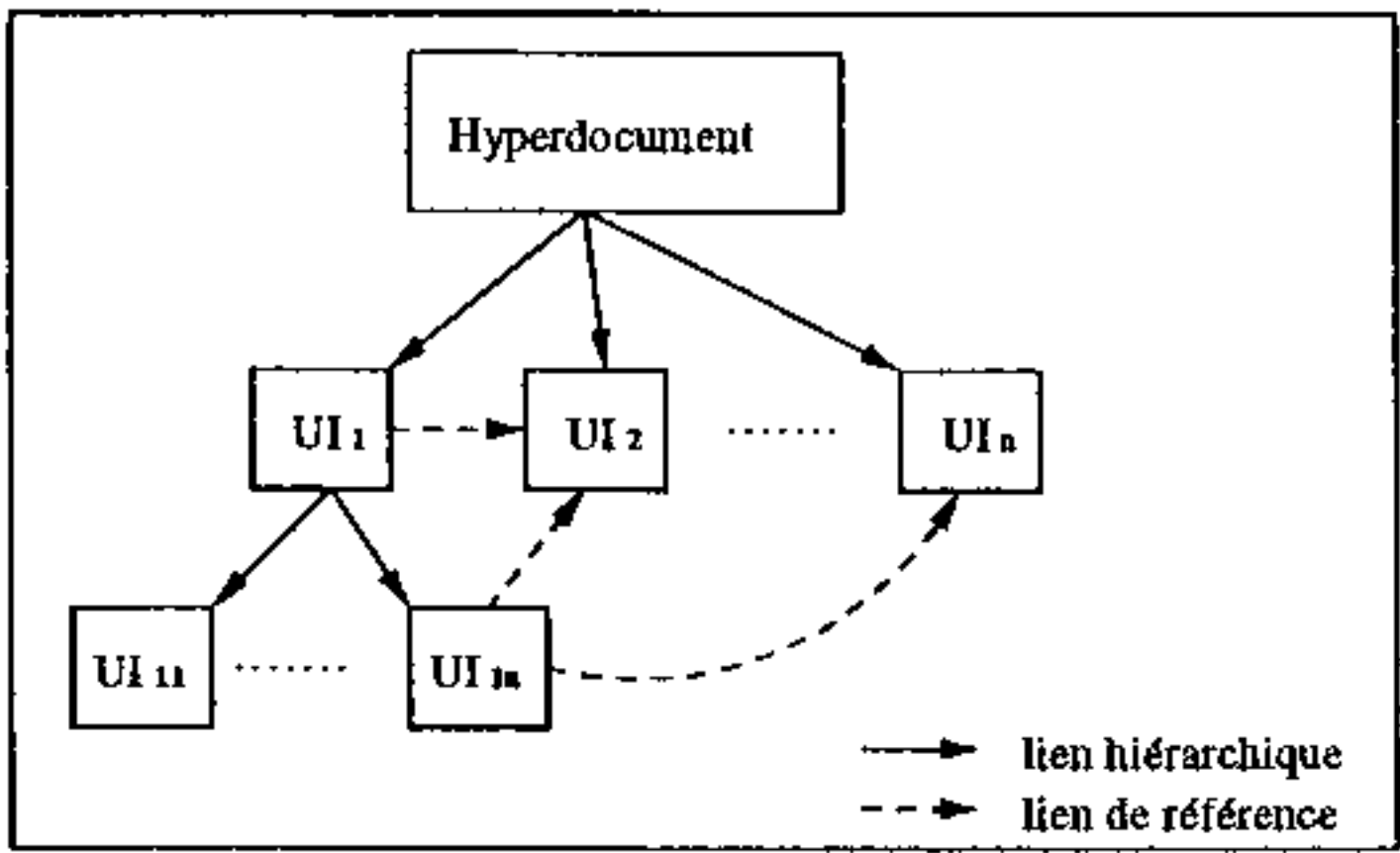


FIG. 3.7 – Structure d'un hyperdocument.

1. L'entité Hyperdoc:

Elle modélise un document hypertexte. Un hyperdocument est caractérisé par les attributs suivants :

- *nomdocht*: de type chaîne de caractères, sa valeur est le nom d'un document, clé de l'entité.
- *titre*: de type chaîne de caractères, sa valeur est le titre du document.

- *date*: de type date, sa valeur est le date de création du document.
- *introduction*: de type chaîne de caractères, sa valeur est un texte qui décrit l'application qui est modélisée.
- *image*: de type chaîne de caractères, sa valeur est le nom d'un fichier image qui illustre l'application qui est modélisée.

Un document contient plusieurs unités informatives et il peut être écrit par plusieurs auteurs.

2. L'entité Auteur:

Elle modélise l'ensemble des auteurs d'un hyperdocument; ses attributs sont :

- *numa*: de type numérique, clé de la relation.
- *nom*: de type chaîne de caractères, sa valeur est le nom de l'auteur d'un document.
- *prénom*: de type chaîne de caractères, sa valeur est le prénom de l'auteur d'un document.
- *adresse*: de type chaîne de caractères, sa valeur est l'adresse de l'auteur du document.

3. L'entité Unité_informative:

Elle permet de modéliser la structure hiérarchique d'un hyperdocument; ses attributs sont :

- *numui*: de type numérique, clé de l'entité.
- *nomui*: de type chaîne de caractères, sa valeur est le nom d'un fichier hypertexte de la base d'hyperdocument.
- *entete*: de type chaîne de caractères, sa valeur est une en-tête de page.
- *pied*: de type chaîne de caractères, sa valeur est un pied de page.

Une unité informative peut être composée d'autres unités informatives (grâce à l'association *Est_composée*). Elle est contenue dans un seul hyperdocument, et possède plusieurs liens de références (grâce à l'association *Lien*) . Elle permet de stocker les informations terminologiques et les caractères définitoires d'un seul concept.

4. L'association Lien :

Elle permet de modéliser les liens de référence; ses attributs sont :

- *numlien*: de type numérique, clé de la relation.
- *début*: de type numérique, sa valeur indique la position du premier caractère qui marque le début d'une ancre.

- *longueur*: de type numérique, sa valeur est le nombre de caractère à prendre en compte en plus de la valeur de Début afin de définir totalement l’ancree.
- *étiquette*: de type chaîne de caractères, sa valeur est une ancre située dans l’unité informative destination.
- *type*: de type chaîne de caractères, sa valeur indique l’information terminologique de type hypertexte contenant l’ancree source du lien (définition, contexte, image), ou le caractère définitoire (CDS, CDTC, CDTR) qui est l’ancree du lien. Si le type désigne image, CDS ou CDTR alors les attributs début et longueur ont une autre interprétation, le tableau ci-dessous résume toutes ces possibilités.

type	début	longueur	interprétation
définition contexte	x	y	position du caractère et longueur de l’ancree
image	x	y	centre d’une zone réactive de rayon fixe
CDS	x	0	x, rang du CDS qui est une ancre
	x	1	x, rang du CDS dont la valeur est une ancre
CDTR	x	0	x, rang d’un attribut du CDTR qui est une ancre
	x	1	x, rang d’un attribut du CDTR, sa valeur est une ancre

FIG. 3.8 – Table d’interprétation des liens.

3.4.5 Le schéma relationnel de la base de concepts

Le schéma relationnel est déduit du modèle Entité/Association [ASC 90] [GAL 87], par application de règles de transformation; toutes les entités deviennent des relations ainsi que les associations : Ecrit, Comporte_cds, Comporte_cdctr, Décrit_concept, Relation_conceptuelle et Lien. Nous obtenons le schéma relationnel suivant :

1. Concept(#numconcept, terme, définition, contexte, image, langue)
2. Relation_conceptuelle(#numconcept, #numconcept2, nomrc)
3. CD_simples(#numcds, nomcds)
4. CD_type_relation(#numcdtr, nomcdtr)
5. Comporte_cds(#numconcept, #numcds, valeurcds, typecds)
6. Comporte_cdtr(#numconcept, #numcdtr, #numenreg)
7. Description(#numdesc, texte, image)
8. Décrit_concept(#numconcept, #numdesc)

9. Hyperdoc(#nomdocht, titre, date, introduction, image)
10. Auteur(#numa, nom, prénom, adresse)
11. Unité_informative(#numui, nomui, entete, pied, numui2, nomdocht, numconcept)
12. Lien(#numlien, début, longueur, étiquette, type, numui, numui2)
13. Ecrit(#numa, #nomdocht)

L'annexe B donne des renseignements complémentaires sur les attributs de ces relations.

La figure 3.9 décrit une instanciation de ces relations avec les concepts du domaine enregistrement de voitures. Les enregistrements des tables Constructeur et Garage sont des caractères définitoires de type relation.

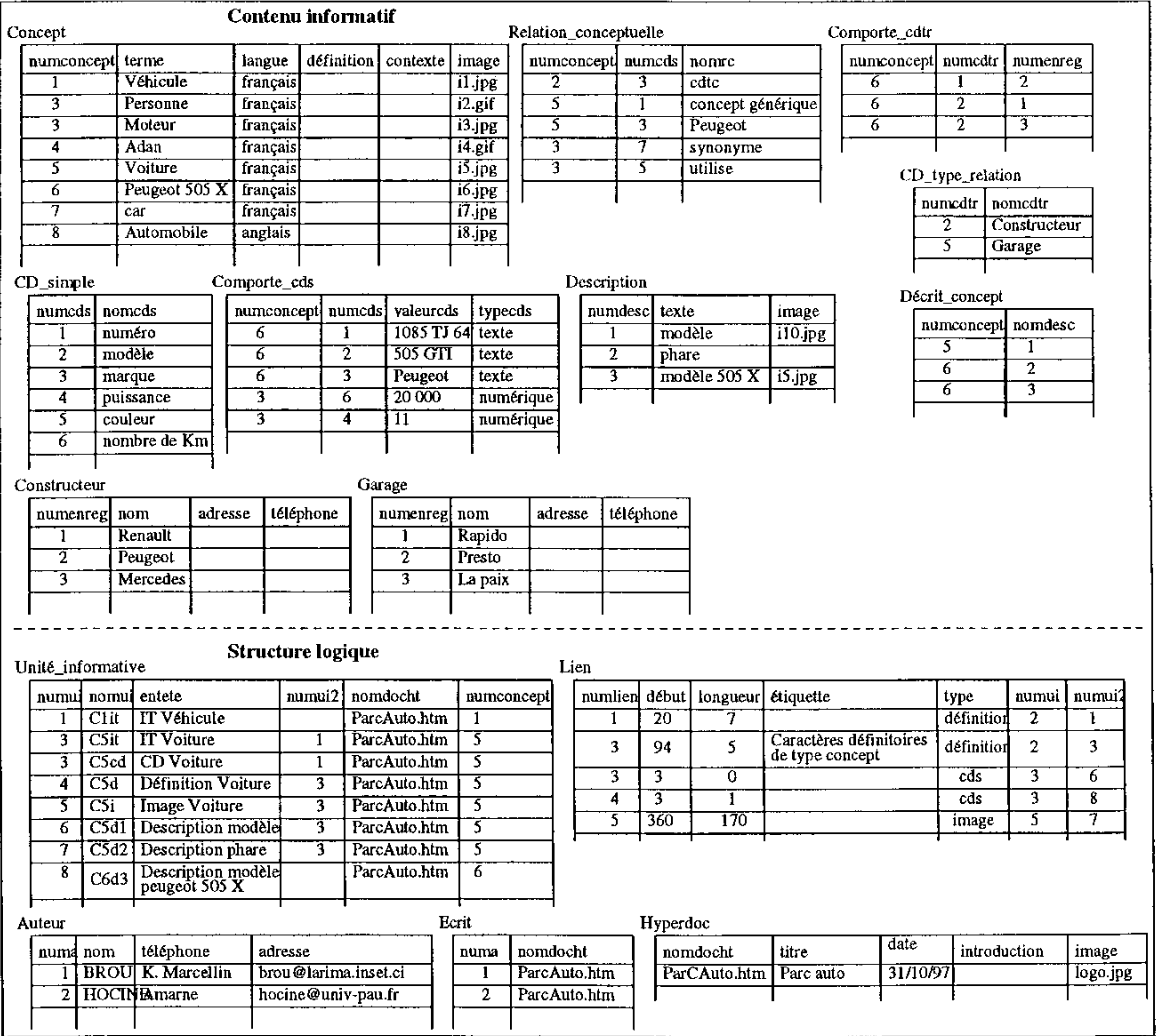


FIG. 3.9 – Instanciation de la BdC pour le domaine enregistrement de voitures

3.5 Conclusion

Nous venons de présenter le modèle de données d'une BdC par la description de son modèle conceptuel dans le formalisme Entité/Association ainsi que le modèle relationnel associé.

Ce modèle conceptuel de donnée est constitué de deux parties qui sont :

- L'aspect informatif permettant de modéliser les informations terminologiques et les caractères définitoires de la BdC.
- L'aspect logique : permettant de modéliser la structure logique de la base d'hyperdocuments; il permet non seulement de représenter les informations terminologiques de type hypertexte, mais aussi d'exploiter de manière hypertextuelle la BdC.

Cette base d'hyperdocuments est obtenue par un processus d'hypérisation du contenu informatif de la BdC (cf. algorithme dans l'annexe A).

Cette approche est intéressante parce que toutes les unités informatives et les liens (hiérarchiques ou référentiels) des hyperdocuments de la BdC sont stockées dans des tables relationnelles; ceci rend aisée les recherches de concepts par l'utilisation de requêtes SQL, et facilite les mises à jour des d'hyperdocuments qui sont effectuées d'abord dans les tables de manière relationnelle, ensuite le processus d'hypérisation les répercute sur la base d'hyperdocuments afin de garantir la cohérence de la BdC.

Dans le chapitre qui suit, nous montrons comment représenter et gérer de manière hypertexte une BdC.

Chapitre 4

Un modèle de représentation et gestion hypertexte des concepts

4.1 Introduction

Nous proposons un modèle hypertexte pour une base de concepts. Il est basé sur le modèle de document hypertexte expert proposé par [FOR 94]. Nous abordons également le problème de représentation, d'accès aux informations ainsi que la maintenance de la base d'hyperdocuments.

4.2 Choix du modèle de données hypertexte

Le modèle de document hypertexte utilisé est basé sur celui qui a été proposé dans [FOR 94]. Nous rappelons ci-dessous le schéma conceptuel et les objets hypertextes de ce modèle à l'aide du formalisme du modèle entité/association étendu.

Comme le montre la figure 4.1, un document Hypertexte Expert (HTE) est un objet complexe basé sur deux principaux concepts : les unités informatives (UI) et les liens hypertextes.

Un document est composé d'UI; une UI qui est soit une UI élémentaire (note, texte, graphique, image), soit une UI composée; et elle peut contenir des liens référentiels qui peuvent référencer un document entier ou une UI.

La composante experte du système est intégrée aux niveaux des différents liens associants

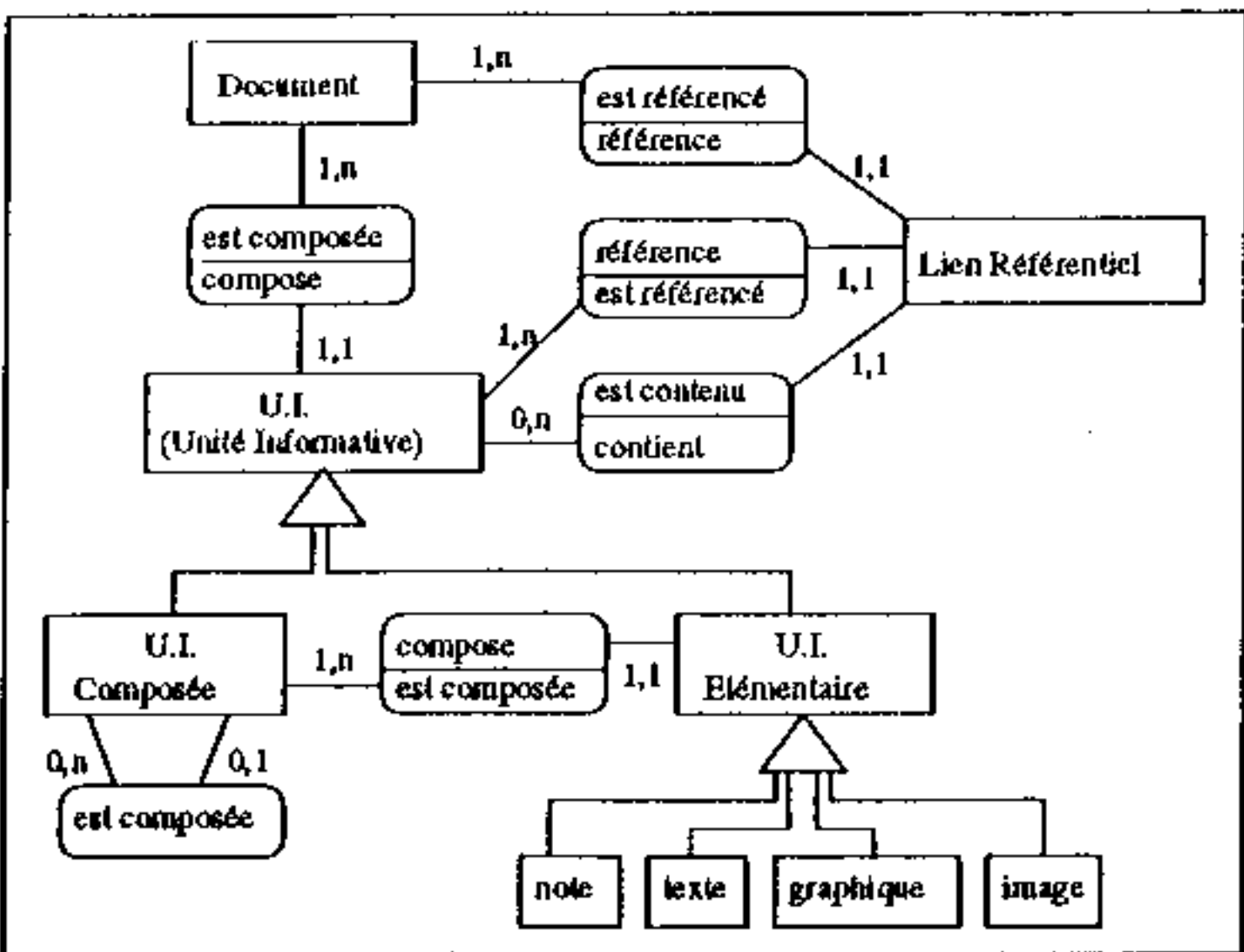


FIG. 4.1 – Schéma conceptuel d'un document hypertexte.

les divers objets documentaires (documents et UI). Elle est décrite à l'aide de prédicats.

4.2.1 Les unités informatives

Une unité informative x est un couple $x : (Ctx, smx)$ [DES 95].

où

- Ctx est une séquence d'informations naturelles ou une séquence d'instructions exécutables, c'est le contenu informatif de l'UI.
- Smx est un ensemble de connaissances traduisant une partie de la sémantique de Ctx et destinées à être exploitées par la composante experte du système, c'est la sémantique de l'UI.

Il existe deux sortes d'UI : les UI élémentaires et le UI composées .

- Une UI élémentaire est une séquence d'informations homogènes c'est à dire ayant le même type, considérées comme atomique (non décomposable). Une UI élémentaire peut être une partie d'une image, d'un graphique, d'un texte, d'un son, d'un programme, etc. C'est la plus petite entité référençable par un lien.
- Une UI comopsée est une UI non-atomique; elle rassemble dans un objet hypertexte unique plusieurs autres UI (pouvant être des UI élémentaires ou des UI comopsée). Elle est donc composée d'informations qui peuvent être hétérogènes.

Un objet documentaire (OD) est une entité de la base de documents référencés par un lien. Il peut être donc soit un document, soit une unité informative d'un document [FOR 94]. Il est

définit par le nom d'un document suivi éventuellement par les noms des UI de ce document séparés par le symbole "/". L'UI qui se trouve à droite de ce symbole est une composante de l'UI qui est à sa gauche.

On peut exprimer en notation BNF un objet documentaire comme suit :

```

<objet_documentaire> ::= "(" <nom_document> {{"<UI>"/"<UI>"} } ")"
<nom_document> ::= <identificateur_document>
<identificateur_document> ::= "chaîne de caractères"
<UI> ::= <identificateur_UI>
<identificateur_UI> ::= "chaîne de caractères"

```

Exemple : l'UI Définition composante de l'UI Information terminologique (UIIT) du document hypertexte voitures.html; on a les objets documentaires suivants :

(voiture.html), (voiture.html, UIIT) et (voiture.html, UIIT/Définition)

4.2.2 Les prédicats

Il existe trois prédicats de manipulation des unités informatives :

1. CIBLER(D,A) : c'est un prédicat binaire permettant d'accéder à une UI par activation d'un lien, et d'exécuter les actions qui lui sont associées.

où:

- D : est l'identificateur de l'UI à atteindre
- A : est la liste des actions à effectuer qui mettent à jour le contexte), La liste A peut être vide (aucune action à effectuer).

2. ACTIF(S) : c'est un prédicat unaire qui teste si l'ancre d'un lien, localisée dans une UI est active. Lorsqu'une ancre est active, elle est visualisée pour l'utilisateur dans le document dont l'UI fait partie. Lorsqu'elle est inactive, elle n'est pas visible.

où:

- S : désigne l'emplacement ou l'élément de l'UI qui est le point d'ancrage.

3. CONTEXTE(C) : c'est un prédicat unaire qui teste si le contexte vérifie certaines conditions portant sur une UI.

où:

- C : est la liste des conditions à vérifier par le contexte.
par exemple un clic sur le bouton Suivant (page 82) affiche les UI du concept qui est situé immédiatement à droite d'un autre concept, au même niveau dans l'arbre de concept; le contexte ici est le concept courant.

4.2.3 Les liens hypertextes

Les rapports sémantiques existant entre deux UI sont traduites par un lien qui les relie. Un lien possède une UI source (UIS) et une UI destination (UID). On distingue deux sortes de liens : les liens de composition et les lien de référence.

Les liens de composition (ou liens hiérarchiques) sont des liens qui associent une UI composée aux différentes UI qui la composent.

Les liens de référence sont des liens qui ne sont pas des liens de composition. On distingue quatre sortes de liens de référence :

- Le lien permanent à destination fixe :
c'est le lien de référence classique que nous trouvons dans tous les systèmes hypertextes. Il est toujours activable à partir de l'UIS ou son ancre s_i est active; l'UID est toujours la même quelque soit le contexte d'activation. Il peut être schématisé comme suit (Fig. 4.2) :

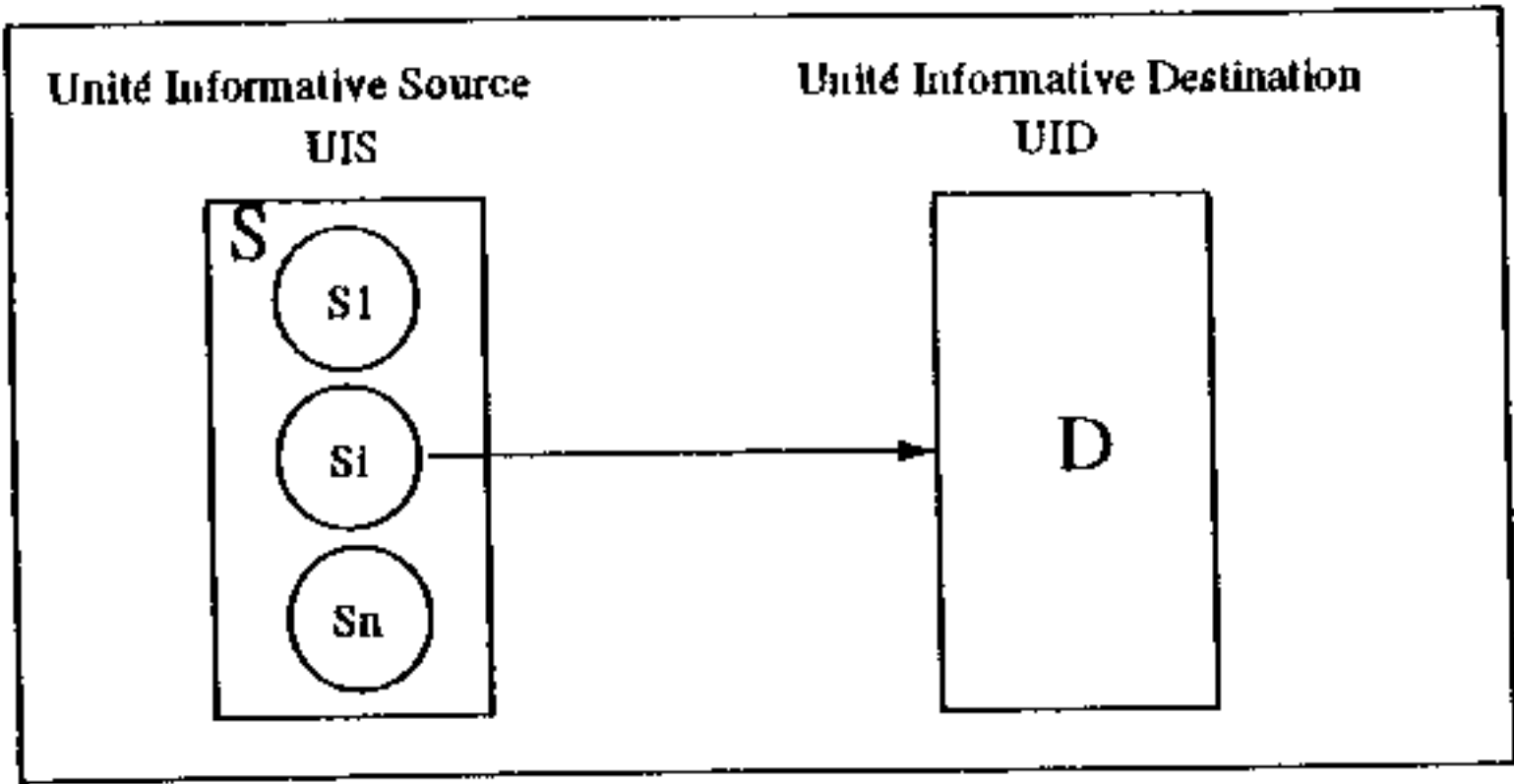


FIG. 4.2 – Lien permanent à destination fixe.

Le prédicat $ACTIF(s_i)$ est toujours vérifié pour ce type de lien. En utilisant le formalisme des règles de production, nous pouvons le représenter comme suit :

SI $ACTIF(s_i)$
ALORS $CIBLER(D)$

- Le lien permanent à destination contextuelle:
ce type de lien est toujours activable à partir de l'UIS où l'ancre est active. C'est le contexte agissant sur la cible qui détermine l'UID référencée en fonction de conditions (notées Cd_i) portant sur ces liens (on parlera alors de liens contextuels). L'existence et par conséquent l'activation de ces liens dépendent du contexte. Ce type de lien peut être schématisé comme suit (Fig. 4.3):

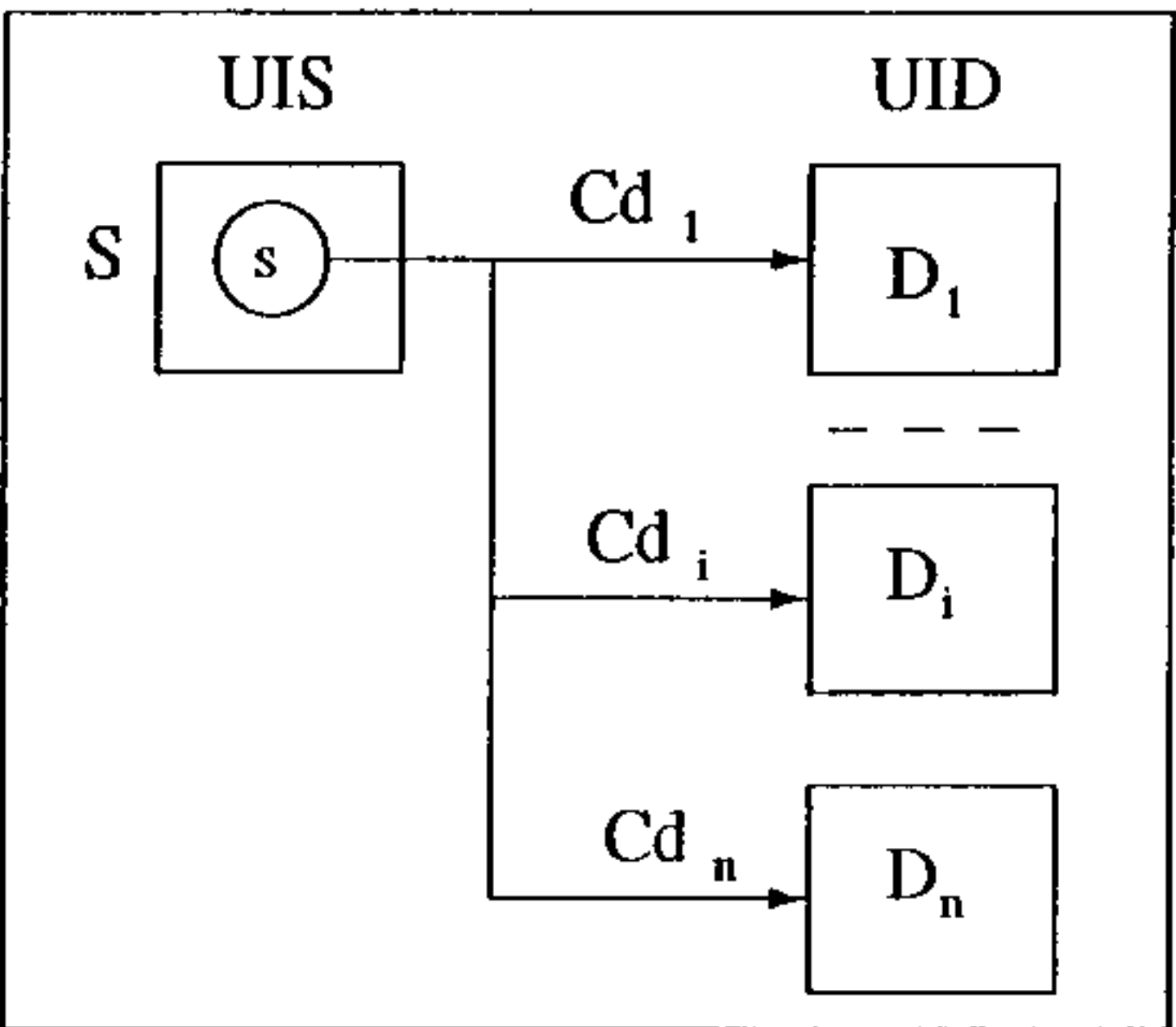


FIG. 4.3 – *Lien permanent à destination contextuelle.*

Cd_i : désigne une liste de conditions à vérifier par le contexte. Si ces conditions sont vérifiées alors la cible D_i est atteinte.

En utilisant le formalisme des règles de production, nous pouvons le représenter comme suit :

SI ACTIF(s)
ET CONTEXTE(Cd_i)
ALORS CIBLER(D_i , A_i)

Le prédicat ACTIF(s) est toujours vérifié. Le choix de l'UID à atteindre est donc fonction du contexte tenant compte des conditions portant sur ces liens.

- Le lien contextuel à destination fixe:
ce lien n'est activable que si le contexte en cours a permis la génération de l'ancre de ce lien (le bouton est affiché); c'est la notion de bouton variable. Il peut être schématisé comme suit (Fig. 4.4):

C_s désigne la liste des conditions que doit vérifier le contexte pour que S soit l'ancre du lien vers l'UID.

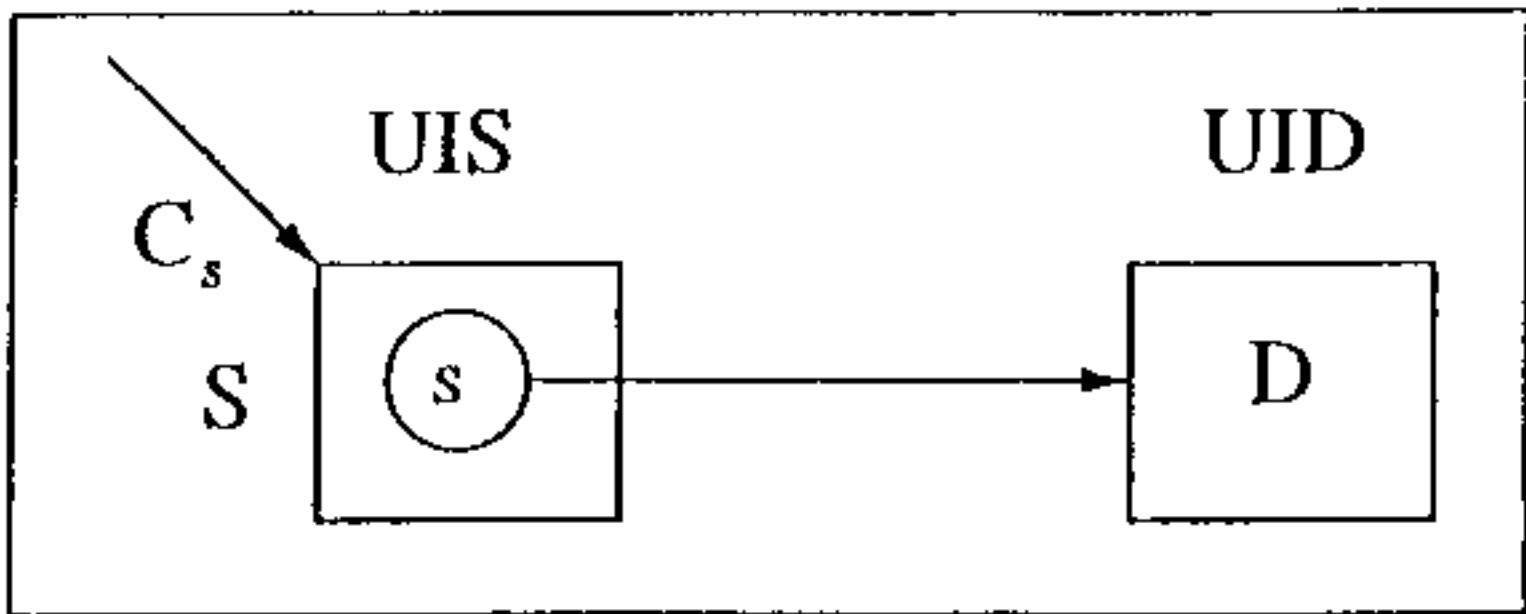


FIG. 4.4 – *Lien contextuel à destination fixe.*

En utilisant le formalisme des règles de production, nous pouvons le représenter comme suit :

SI	CONTEXTE(C_s)	Le contexte vérifie les conditions C_s
ALORS	ACTIF(s)	
	CIBLER(D,A)	
SINON	ACTIF(s)=FAUX	L'ancree s n'est pas actif; le bouton est transparent pour l'utilisateur

- Le lien contextuel à destination contextuelle:
Ce lien est la généralisation des précédents liens. Il peut être schématisé comme suit (Fig. 4.5):

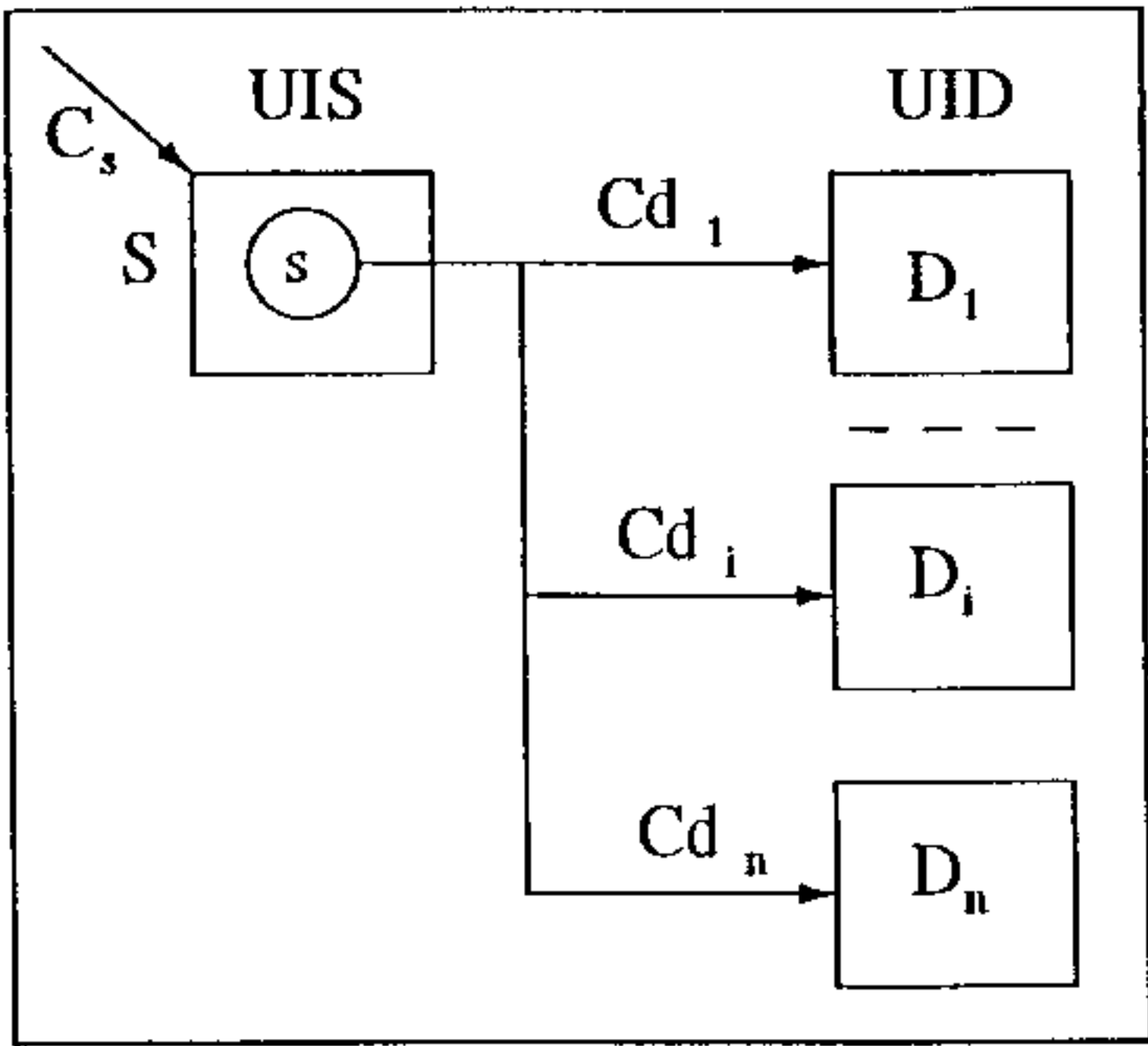


FIG. 4.5 – *Lien contextuel à destination contextuelle.*

En utilisant le formalisme des règles de production, nous pouvons le représenter comme suit :

SI	CONTEXTE(C_s)	Le contexte vérifie les conditions C_s
ALORS	ACTIF(s)	

	CIBLER(D_i, A_i)	
SINON	ACTIF(s)=FAUX	L'ancre s n'est pas actif; le bouton est transparent à l'utilisateur
SI	ACTIF(s_1)	
	CONTEXTE(Cd_1)	Le contexte vérifie les conditions Cd_1
ALORS	CIBLER(D_1, A_1)	
... ..		
SI	ACTIF(s_n)	
	CONTEXTE(Cd_n)	Le contexte vérifie les conditions Cd_n
ALORS	CIBLER(D_n, A_n)	

A partir du modèle de document hypertexte expert que nous venons de décrire, nous allons présenter le modèle hypertexte adopté pour une base de concepts.

4.3 Un modèle hypertexte pour une base de concepts

Nous proposons un modèle hypertexte pour une base de concepts. Il s'appuie sur les unités informatives et les liens du modèle hypertexte expert présenté au § 4.2.

Dans le schéma de la figure 4.6, seuls apparaissent clairement les liens hiérarchiques à travers les décompositions successives des concepts complexes en sous-concepts. Les liens de référence sont implicites au travers des informations terminologiques de type relation conceptuelle (telle que concept lié) et de type hypertexte (telle que définition); la figure 4.8 page 80 illustre bien ces liens de référence.

4.3.1 Les unités informatives

On distingue six types d'unités informatives qui correspondent aux informations manipulées dans la BdC (Fig. 4.6).

- 1. **Unité informative pour les informations terminologiques (UIIT) :**
c'est une UI complexe qui renferme toutes les informations terminologiques d'un concept (Fig. 4.7). Ces informations sont représentées par un couple de la forme :

IT : valeur

où

- IT désigne une information terminologique, et

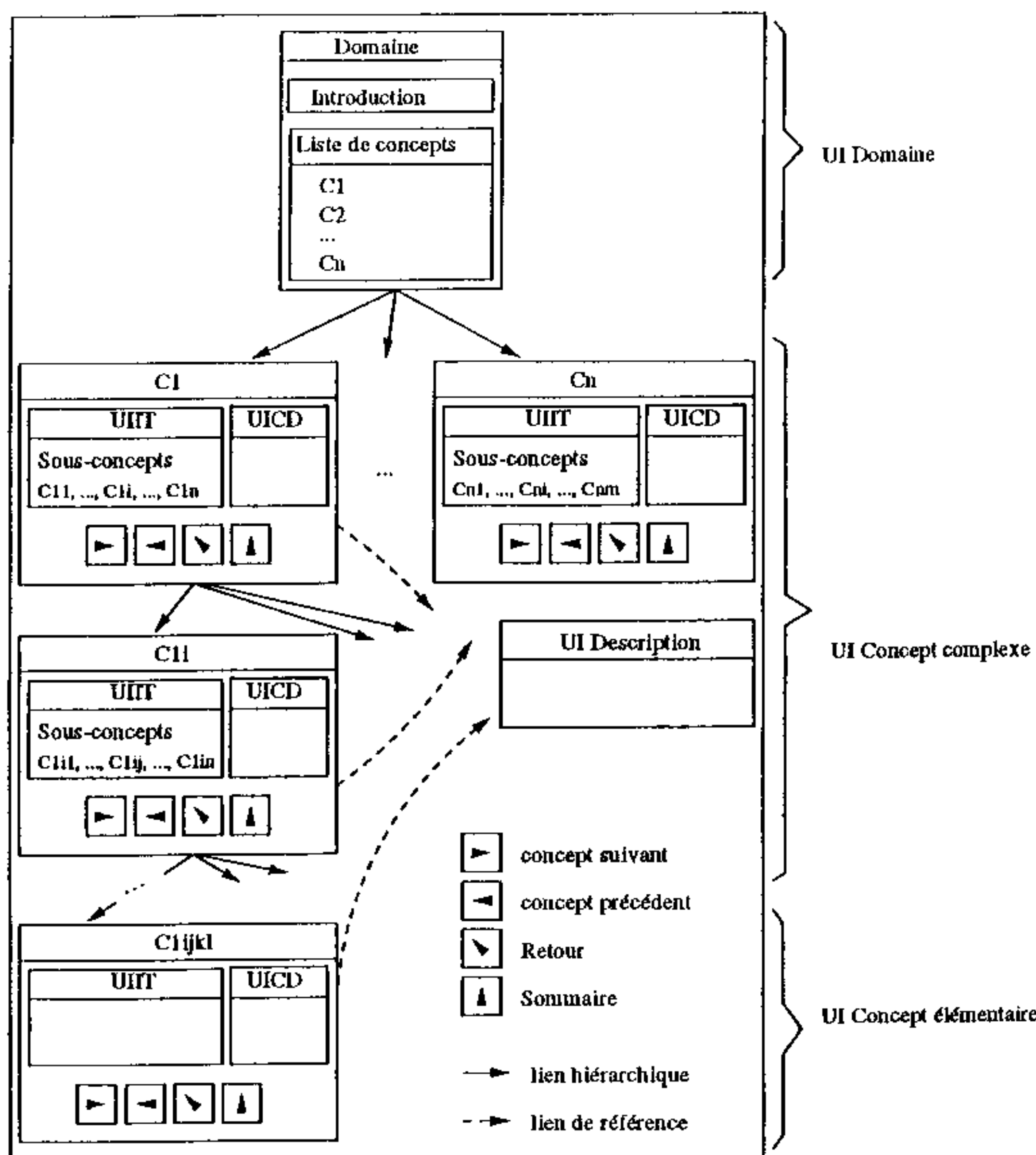


FIG. 4.6 – Structuration des unités informatives de la BdC.

- valeur, sa valeur stockée dans la BdC; elle peut être simple : valeur des informations terminologiques de type attribut (ITA) ou complexe : valeur des informations terminologiques de type hypertexte (ITHT) ou de type relation conceptuelle (ITRC). La valeur d'une ITHT peut être un hypertexte ou une image; celle d'une ITRC est le terme d'un concept qui est l'ancre d'un lien de référence.

Ce type d'unité informative contient des liens vers tous les autres unités informative (cf. figure 4.8 page 80).

2. Unité informative pour les caractères définitoires (UICD):

c'est une UI composée qui contient toutes les informations permettant de décrire les caractères définitoires d'un concept (Fig. 4.7). Ces informations sont représentées selon le type du caractère définitoire :

- Représentation des caractères définitoires simples (CDS):
ils sont représentés comme suit :

où CDS désigne un caractère définitoire simple et valeur représente sa valeur; cette valeur est de type simple.

- Représentation des caractères définitoires complexes : ils sont de la forme

valeur

où valeur de type complexe désigne la valeur d'un caractère définitoire de type complexe.

Pour les caractères définitoires de type concept, cette valeur est le terme d'un concept qui est l'ancre d'un lien de référence.

Pour les caractères définitoires de type relation, cette valeur est une table relationnelle. (cf. figure 4.8 page 80)

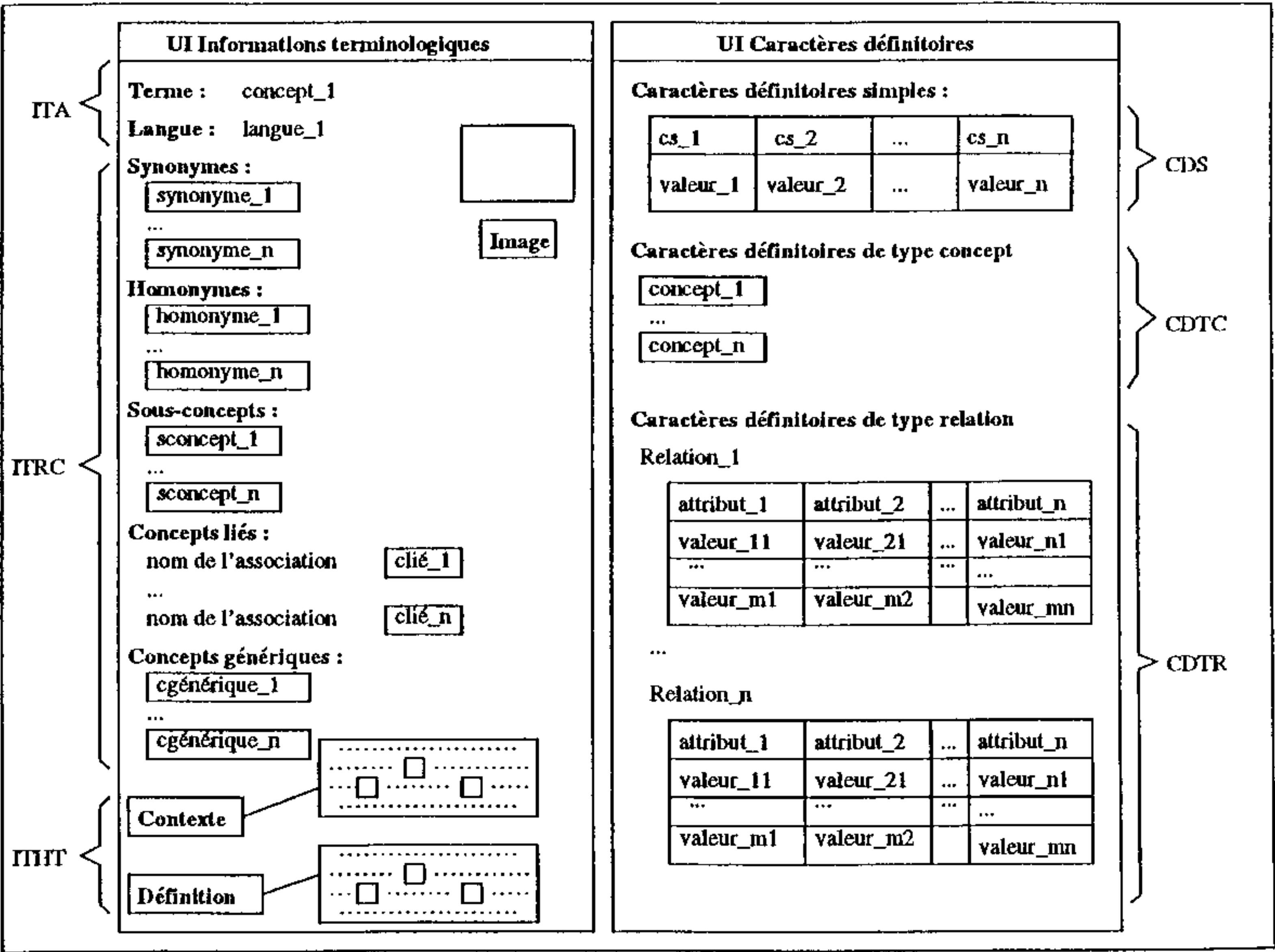


FIG. 4.7 – Exemples d’UIIT et d’UICD.

La figure 4.7 présente toutes les composantes des UI information terminologique (ITA, ITRC, ITHT) et des UI caractère définitoire (CDS, CDTC, CDTR).

3. Unité informative pour les concepts complexes (UI Concept complexe) : Cette UI est constituée des unités informatives UIIT et UICD. Chaque concept complexe est représenté par une UI Concept complexe qui est composée par les UI de ses sous-concepts (cf. figure 6.8 page 136).

4. Unité informative pour les concepts élémentaires (UI Concept élémentaire) :

Une UI Concept élémentaire est identique à une UI Concept complexe, sauf que le concept qu'il représente est un concept élémentaire. Le champ "sous-concept" n'est donc pas alimenté (cf. figure 6.10 page 138).

5. Unité informative de description (UI Description) :

C'est une UI complexe constituée d'UI élémentaires de type texte et/ou graphique permettant de décrire les informations qui ne sont pas du domaine (cf. figure 6.13 page 141)

6. Unité informative du domaine (UI Domaine) :

C'est une UI composée constituée d'une UI élémentaire contenant un texte de présentation du domaine, et d'une liste de concepts génériques les plus abstraits du domaine, ancres de lien qui pointent vers des UI Concept complexe (cf. figure 6.7 page 135).

4.3.2 Les liens hypertextes

Il existe deux types de lien : les liens hiérarchiques et les liens de référence.

4.3.2.1 Les liens hiérarchique

Les liens hiérarchiques structurent une UI composées avec ses UI composantes; ils sont implicites au travers de la description de l'UI composée, et ne sont pas décrit en tant qu'entité individuelle.

Dans la BdC, ces liens hiérarchiques sont décrits par les arbres conceptuels du domaine (Fig. 4.6); concrètement, ils sont modélisés au travers de la relation Unité_informative; nous rappelons sa déclaration en intension.

Unité_informative(numui, nomui, entete, pied, numui2, nomdocht, numconcept)
avec

- *numui* : de type numérique, clé de la relation.
- *nomui* : de type chaîne de caractères, sa valeur est le nom d'un fichier hypertexte de la base d'hyperdocuments.
- *entete* : de type chaîne de caractères, sa valeur est une entête de page.
- *pied* : de type chaîne de caractères, sa valeur est un pied de page.

- *numui2*: de type numérique, numéro de l'unité informative dont l'UI de numéro *numui* est une composante.
- *nomdocht*: de type chaîne de caractères, sa valeur est le nom d'un document hypertexte dont l'UI est composante.
- *numconcept*: de type numérique, numéro du concept dont les informations sont stockées dans l'UI.

Une unité informative est contenue dans un seul hyperdocument et elle permet de stocker les informations terminologiques et les caractères définitoires d'un seul concept. Elle peut être la composante d'une autres unité informative; cette structuration hiérarchiques entre unités informatives est assurée l'attribut *numui2*.

La table *Unité_informative* de la figure 3.9 page 66 indique que l'UI C5it a pour composantes les UI C5it et C5cd et que l'UI C5it a pour composantes les UI C5d, C5i, C5d1 et C5d2.

4.3.2.2 Les liens de référence

Les liens de référence associent deux UI différentes appartenant à un même hyperdocument ou à deux hyperdocument différents; ils sont explicites et sont représentés par des entités à part entière. Ils sont modélisés dans la BdC par la relation *Lien*; nous rappelons sa déclaration en intension.

Lien(*numlien*, *début*, *longueur*, *étiquette*, *type*, *numui*, *numui2*)

avec

- *numlien*: de type numérique, clé de la relation.
- *début*: de type numérique, sa valeur indique la position du premier caractère qui marque le début d'une ancre.
- *longueur*: de type numérique, sa valeur est le nombre de caractère à prendre en compte en plus de la valeur de *Début* afin de définir totalement l'ancre.
- *étiquette*: de type chaîne de caractères, sa valeur est une ancre située dans l'unité informative destination.
- *type*: de type chaîne de caractères, sa valeur indique l'information terminologique de type hypertexte (définition, contexte, image, ...) ou le caractère définitoire (CDS, CDTC, CDTR) contenant l'ancre source du lien.

- *numui*: de type numérique, numéro de l'unité informative qui contient l'ancree source (déterminée par la valeur des attributs début et longueur).
- *numui2*: de type numérique, numéro de l'unité informative qui contient l'ancree destination (déterminée par la valeur de l'attribut étiquette).

Nous distinguons quatre types de lien de référence (FIG. 4.8) : les liens référençant un concept, les lien référençant un hypertexte, et les liens ancrés dans un hypertexte (et pouvant référencer n'importe quel objet de la BdC) et les liens référençant une UI de description. Tous ces liens sont des liens permanents à destination fixe.

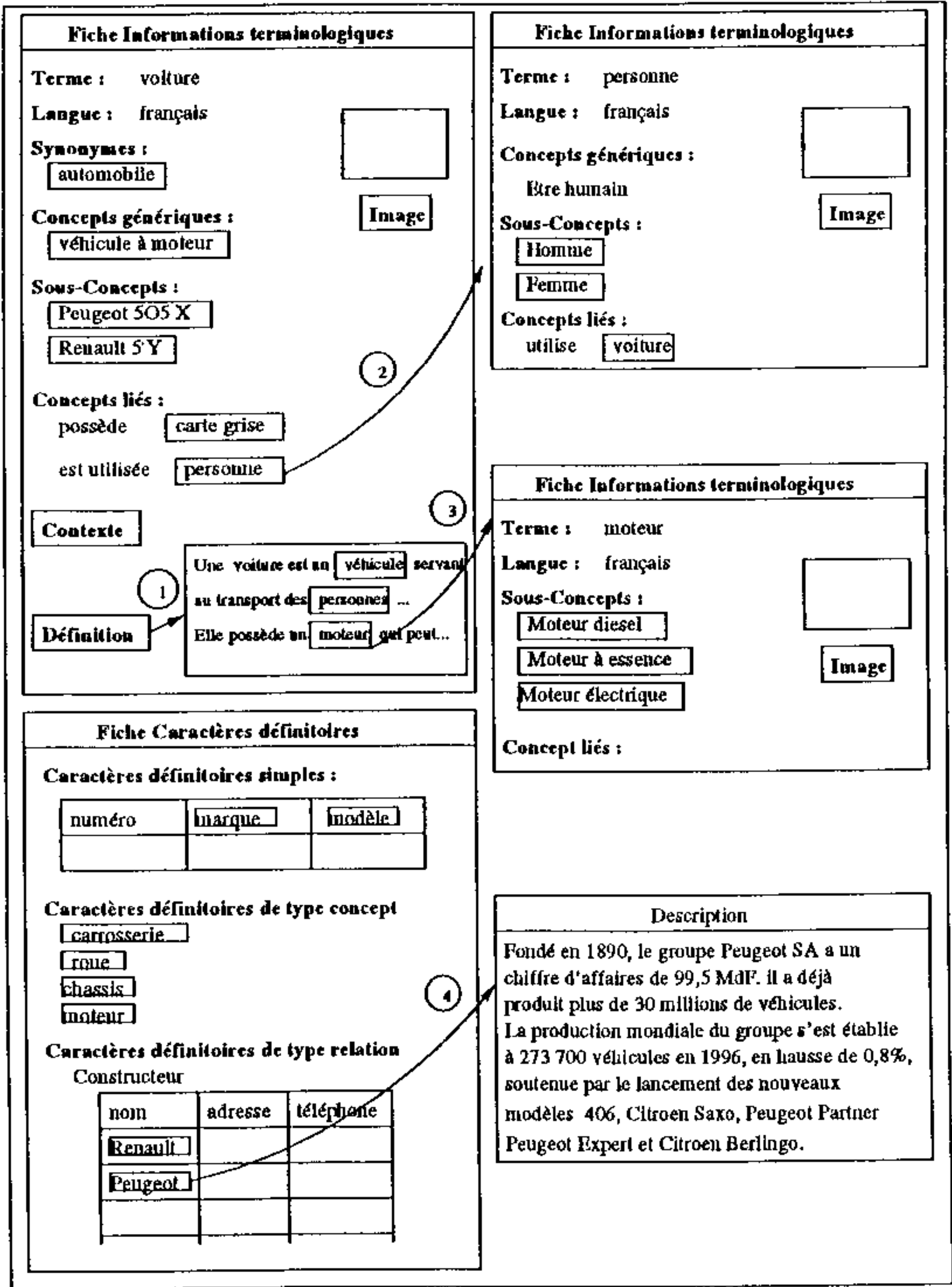


FIG. 4.8 – Les liens de référence.

- Les liens référençant un concept : ces liens permettent de représenter la relation sémantique qui existe entre les concepts, et modélisée par les informations terminologiques de type relation conceptuelle (synonyme, homonyme et concept lié) et par les

caractères définitoires de type concept. Ces liens, ancrés dans une UIIT ou dans une UICD ont pour destination une UIIT (Fig. 4.8, lien numéro 2).

- **Les liens référençant un hypertexte**: ces liens permettent d'accéder à la représentation hypertextuelle d'une information terminologique de type hypertexte : Définition, contexte et Image (Fig. 4.8, lien numéro 1).
- **Les liens de référence ancrés dans un hypertexte**: ces liens sont ancrés dans la représentation hypertextuelle d'une information terminologique de type hypertexte. Ils peuvent avoir comme destination n'importe quel objet de la BdC : UIIT, UIC, ... (Fig. 4.8, lien numéro 3).
- **Les liens de description**: ces liens sont ancrés dans la représentation hypertextuelle d'une information terminologique de type hypertexte ou dans un caractère définitoire simple ou un caractère définitoire de type relation (Fig. 4.8, lien numéro 4)

Les deux premiers liens sont implicites dans la BdC, sauf les deux derniers qui sont modélisés par la relation Lien.

La table Lien de la figure 3.9 page 66 présente une instantiation de ces différents types de lien.

4.4 Exploitation hypertexte de la BdC

L'exploitation hypertexte de la BdC que nous proposons est caractérisée par deux aspects :

- le premier porte sur la manipulation hypertextuelle de la BdC fondée sur la notion de fiche terminologique,
- le second est relatif au processus d'hypérisation des informations terminologiques de type hypertexte.

Dans les paragraphes qui suivent, nous montrons comment gérer une BdC de manière hypertexte selon le modèle choisi au § 4.4.2. D'abord en présentant les moyens d'accès aux informations (navigation, requête, notion de point de vue), puis en décrivant le processus d'hypérisation, et enfin en analysant les mécanismes de maintenance et de vérification de la cohérence de la BdC.

4.4.1 Manipulation de la BdC

La consultation des concepts est effectuée grâce à une interface hypertextuelle basée sur la notion de fiche terminologique;

Une fiche représente un concept à l'aide d'un ensemble de champs textuels contenant les valeurs propres au concept. Ces champs contiennent les informations terminologiques ainsi que les caractères définitoires d'un concept. A chaque concept est associé deux fiches :

- une fiche pour les informations terminologiques modélisée par une UIIT, et
- une fiche pour les caractères définitoires modélisée par une UICD.

Terme :
Langue :
Synonymes :
Concepts génériques :
Sous-concepts :
Concepts liés :
Contexte
Définition

voiture
français
automobile
véhicule à moteur
Peugeot 505 X
Renault 5 Y
possède : carte grise
est utilisée : personne
Une voiture est un véhicule servant au transport des personnes ...
Elle possède un moteur qui peut...

Image

Caractères définitoires simples :

numéro

marque

modèle

Caractères définitoires de type concept

carrosserie
roue
chassis
moteur

Caractères définitoires de type relation

Constructeur

nom

adresse

téléphone

Renault

Peugeot

Description

Suivant

Précédent

Retour

Sommaire

FIG. 4.9 – Interface de consultation de la BdC

La figure 4.9 présente un exemple d’instanciation de ces fiches pour le concept Voiture. Ces deux fiches sont visualisées simultanément sur la même page écran; le passage d’un concept à un autre se fait par activation de lien de référence décrits précédemment, et de liens navigationnels que nous décrirons plus loin.

L’activation de l’ancre Définition (idem pour Contexte et Image) permet de visualiser l’hypertexte correspondant.

Le déclenchement d'un lien référençant un concept amène à l'écran les fiches terminologiques de ce concept. Par exemple l'activation de l'ancre personne entraîne l'affichage des fiches terminologique du concept Personne, en remplacement de celles du concept Voiture.

La zone Description permet l'affichage du contenu des unités informatives de description.

L'accès aux informations de la base de concepts s'effectue soit par navigation grâce à l'interface hypertextuelle que nous venons de présenter, soit par une requête suivie de navigation.

4.4.1.1 Accès à la BdC par navigation

Dans une BdC, il est important que l'utilisateur puisse passer d'un concept à son (ou ses) concept générique et vice versa, ou passer d'un concept à son plus proche voisin ayant le même concept générique. La navigation dans la base d'hyperdocuments (BHD) est assurée par trois types de liens navigationnels : liens verticaux, liens horizontaux et liens diagonaux. A chaque type de lien est associé un lien inverse (Fig. 4.9).

- Les liens verticaux permettent de parcourir l'arbre de concepts dans un sens descendant ou dans un sens ascendant :
 - le parcours dans le sens descendant s'effectue par l'intermédiaire de l'information terminologique Sous-concept;
 - le parcours dans le sens ascendant est assuré par l'information terminologique Concept générique.
- Les liens horizontaux permettent de passer d'un concept à son plus proche voisin ayant le même concept générique que lui. Il sont matérialisés par des liens dont les ancres sont Suivant et Précédent.
- Les liens diagonaux permettent d'accéder à n'importe quel objet documentaire; ils sont modélisés par les liens de référence.

Un lien de référence inverse est représenté par un lien d'ancres Retour permettant de revenir à l'unité informative source.

- Le lien d'ancre Sommaire existe dans toutes les unités informatives (sauf l'UI du domaine); il permet de retourner rapidement à l'UI du domaine.

Les liens de navigation d'ancre Suivant, Précédent et Retour sont des liens permanents à destination contextuelle. En effet, l'objet documentaire destination de ces liens n'est pas

Un exemple de code de ces liens est donné en JavaScript au § 5.3.3.3 page 118.

4.4.1.2 Accès à la BdC par requête

Les requêtes sont utilisées lorsque l'utilisateur a une idée précise d'un concept et qu'il souhaite le localiser facilement. Dans la BdC, ces requêtes sont de type relationnel. Après l'exécution de la requête par le SGBGR, la réponse est transmise au système hypertexte pour la présentation à l'utilisateur par le biais des objets documentaires correspondant à la réponse fournie; nous utilisons pour cela les prédicats définis sur le modèle de document hypertexte expert que nous avons présenté au § 4.2.2.

Nous modélisons l'expression d'une requête en notation BNF de la façon suivante :

```

<Requête>      ::=  "Localiser("<x> "," <concept> ")" |
                    "Localiser("<x> "," <restriction> "," <concept> ")" |
                    "Localiser("<x> "," <chaîne> "," <concept> ")" |
                    "Localiser("<x> "," <attribut_sel> "," <restriction> ")" |
                    "Localiser("<x> "," <attribut_sel> "," <restriction> ","
                    <concept> ")"
<x>             ::=  chaîne de caractères, élément d'une vue
<Concept>      ::=  "terme d'un concept" | "tout".
<Restriction>  ::=  ["("<nom_attribut> <opérateur> <expression> [")"] [ET | OU]
                    ["("] <nom_attribut> <opérateur> <expression> [")"]
<attribut_sel> ::=  <nom_attribut> ["," <nom_attribut>]
<nom_attribut> ::=  "nom d'un attribut"
<opérateur>    ::=  "un opérateur de comparaison"
<expression>   ::=  <nom_attribut> | <valeur>
<valeur>       ::=  "valeur d'information de type classique"

```

Dans cette définition, le symbole terminal "tout" désigne tous les concepts.

Nous présentons ci-dessous des exemples de requêtes portant sur la recherche de concepts en se basant sur leurs informations terminologiques et leurs caractères définitoires.

Le type de requête est défini par le type d'information terminologique (ITA, ITHI ou ITRC) ou sur le type caractère définitoire (CDS, CDTC ou CDTR). La manipulation des

attributs de jointure est transparente à l'utilisateur.

1. Requête portant sur les informations terminologiques

- sur une information terminologique de type relation conceptuelle :

"Quels sont les concepts liés par la relation conceptuelle R_c avec le concept C_i ?"

Cette requête s'exprime comme suit :

$\Rightarrow$ *Localiser(Concept lié, nomrc = R_c , C_i)*

Elle sera traduite en SQL de la façon suivante :

```
SELECT numconcept, numconcept2 INTO L1
FROM   Relation_conceptuelle
WHERE  nomrc = "Rc";
```

```
SELECT numconcept INTO L2
FROM   Concept
WHERE  (numconcept IN L1) AND (terme = " $C_i$ ") ;
```

```
SELECT nomdocht INTO L3
FROM   Unit_informative
WHERE  numconcept IN L2 ;
```

L2 est la liste des numéros des concepts qui sont liés au concept C_i par la relation "Rc" et

L3 la liste des objets documentaires qui décrivent ces concepts.

Pour localiser ces OD dans la base d'hyperdocuments nous utilisons le prédicat CIBLER qui permet d'atteindre un OD comme suit :

CIBLER(OD_i , A_i)

où

- OD_i désigne un objet documentaire de L3
- A_i désigne le contexte de OD_i .

Exemple d'instanciation de cette requête :

"Quels sont les concepts liés dans la Relation conceptuelle utilise avec le concept Peugeot 505X?"

Cette requête se traduit par :

$\Rightarrow$ Localiser(Concept lié, nomrc = "utilise", "Peugeot 505X")

La question réelle de l'utilisateur étant :

"Quelles sont les personnes qui utilisent la voiture Peugeot 505X?".

- sur une information terminologique de type attribut :

"Quels sont les synonymes du concepts C_i , dans la langue de spécialité L?"

$\Rightarrow$ Localiser(Synonyme, langue = L, C_i)

Exemple d'instanciation de cette requête :

"Quels sont les synonymes en anglais de voiture?"

$\Rightarrow$ Localiser(Synonyme, langue = "anglais", "voiture")

- sur une information terminologique de type hypertexte :

"Quels sont les concepts dont la définition contient le terme T?"

$\Rightarrow$ Localiser(Définition, T, tout)

Exemple d'instanciation de cette requête :

"Quels sont les concepts dont la définition contient le terme véhicule?"

$\Rightarrow$ Localiser(Définition, "véhicule", tout)

2. Requête portant sur les caractères définitoires

- sur un caractère définitoire simple :

"Quels sont les concepts dont le caractère définitoire simple nomcs est supérieur à x?"

$\Rightarrow$ Localiser(CD_simple, numconcept, (nomcds > x) ET (typecds = "numérique")

Exemple d'instanciation de cette requête :

"Quels sont les voitures dont le moteur a une puissance supérieur à 11 chevaux?"

$\Rightarrow$ Localiser(CD_simple, numconcept, (puissance > 10) ET (typecds = "numérique")

- sur un caractère définitoire de type concept :

"Quels sont les concepts qui sont caractères définitoire du concept C_i ?"

$\Rightarrow \text{Localiser}(\text{Relation_conceptuelle}, \text{nomrc} = \text{"CDTC"}, C_i)$

Exemple d'instanciation de cette requête: *"Quels sont les concepts qui sont caractères définitoire de voiture ?"*

$\Rightarrow \text{Localiser}(\text{Relation_conceptuelle}, \text{nomrc} = \text{"CDTC"}, \text{"voiture"})$

- sur un caractère définitoire de type relation :

"Quels sont les tuples de la relation de numéro numcdtr qui sont caractères définitoires du concept C_i ?"

$\Rightarrow \text{Localiser}(\text{CD_type_relation}, \text{numcdtr}, C_i)$

Exemple d'instanciation de cette requête:

"Quels sont les garages où est réparée la Peugeot 505X?"

$\Rightarrow \text{Localiser}(\text{CD_type_relation}, \text{numcdtr}, \text{"Peugeot 505X"})$

L'intérêt de ce modèle de requête est qu'il porte uniquement sur les éléments du modèle conceptuel que nous avons défini pour une BdC; de plus la formulation d'une requête ne nécessite pas la connaissance des attributs de jointure parfois très complexes à exprimer.

Afin de tenir compte des différentes préoccupations des utilisateurs potentiels d'une BdC (terminologie, base de données, système de recherche d'informations documentaires, ...), nous proposons la notion de point de vue qui permet de personnaliser la base d'hyperdocuments.

4.4.1.3 Notion de point de vue

Compte tenue de la diversité des utilisateurs d'une base de concepts, qui n'ont pas forcément les mêmes préoccupations, nous proposons la notion de point de vue [ALA 95] [FAL 96] qui permet d'avoir une vue partielle de la BdC correspondant au domaine d'activité ou au centre d'intérêt d'un utilisateur. Par exemple l'information terminologique Image peut ne pas intéresser un utilisateur, alors qu'il intéresse un autre.

Un point de vue est modélisé par la liste des informations terminologiques et les caractères définitoires qui correspondent aux préoccupations d'un utilisateur.

Nous représentons en notation BNF un point de vue comme suit :

$$\begin{aligned} \langle vue \rangle &::= "(\langle x \rangle \{[, \langle x \rangle]\})" \\ \langle x \rangle &::= "tout" \mid "terme" \mid "synonyme" \mid "homonyme" \mid "concept \text{ lié}" \mid \\ &\quad "image" \mid "définition" \mid "contexte" \mid "caractère définitoire" \mid \\ &\quad "caractère définitoire simple" \mid "caractère définitoire de type concept" \mid \\ &\quad "caractère définitoire de type relation" \end{aligned}$$

Nous présentons ci-dessous trois vues qui correspondent à trois préoccupations différentes :

1. $vue_1(tout)$
2. $vue_2(terme)$
3. $vue_3(terme, \text{concept générique}, \text{définition}, \text{image})$

La vue_1 donne un accès complet à toutes les informations de la BdC alors que les vue_2 et vue_3 en donnent un accès limité. La vue_2 est la plus réduite des vues, elle donne un accès qui se résume à la consultation des termes désignant les concepts. Elle est le dénominateur commun de toutes les vues.

La génération d'une base d'hyperdocuments personnalisée dépend du point de vue de chaque utilisateur de la BdC.

Cette notion de point de vue est à rapprocher des notions de dérivation de contexte [DEL 87] et de document conceptuel [NAN 89] [NAN 93] [RIC 89], mais dans un environnement mono-utilisateur (Fig. 4.10).

On peut aussi le comparer à la notion de graphe de contexte [MEN 94] qui représente le sous-ensemble des informations (classe d'objets et de liens) qui sont le plus fréquemment manipulé par une classe d'utilisateurs.

Dans la figure 4.10, le choix d'une vue entraîne la dérivation d'une partie de la BHD globale. Les BHD personnalisées sont constituées d'objets documentaires OD_{ij} qui sont des instances des objets documentaires OD_i de la BHD globale.

La figure 4.11 présente deux instances d'un objet documentaire du concept voiture pour

les points de vues :

vue1(terme, langue, synonyme, concept lié, définition) et

vue2(terme, langue, concept générique, définition, image).

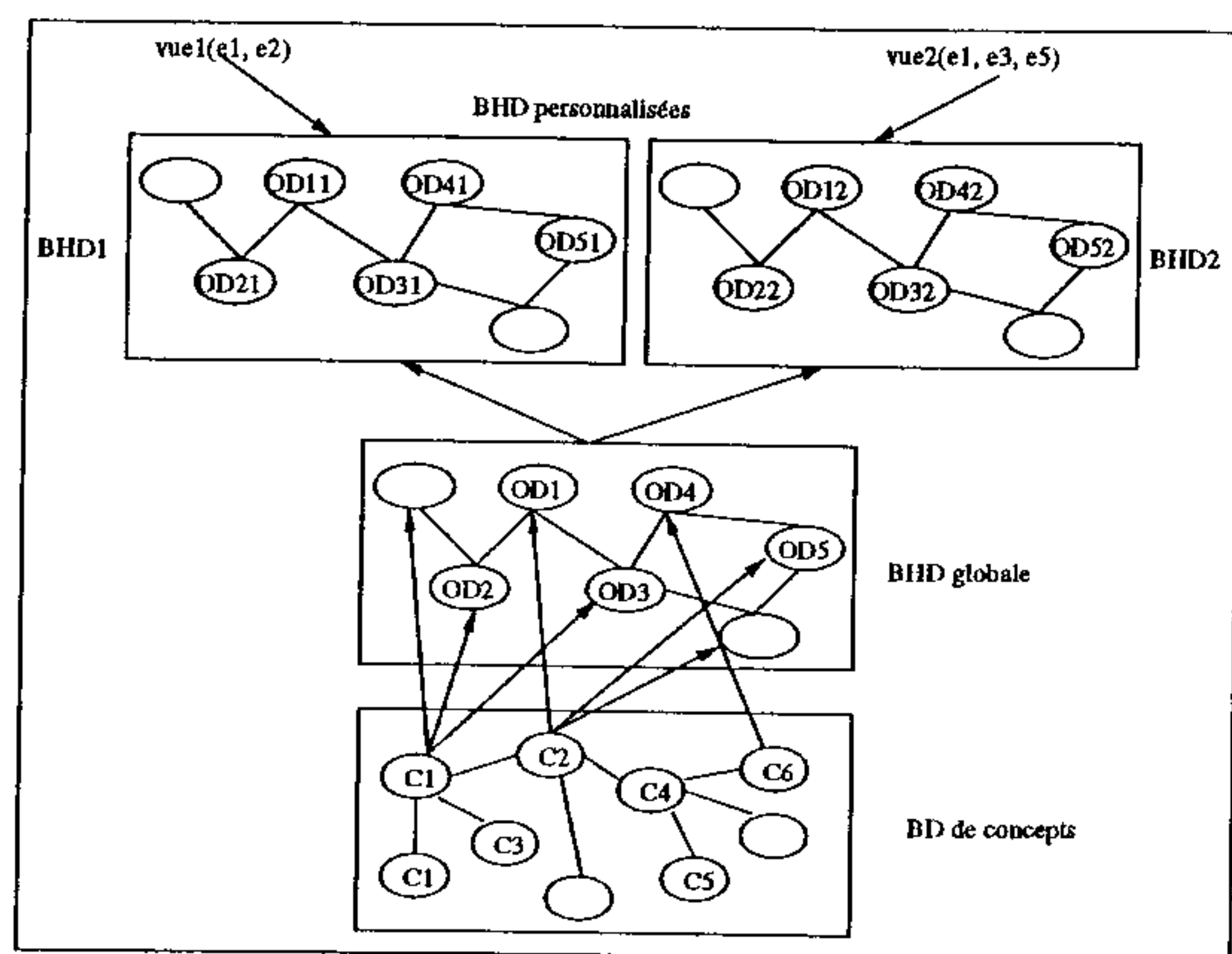


FIG. 4.10 – Notion de point de vue.

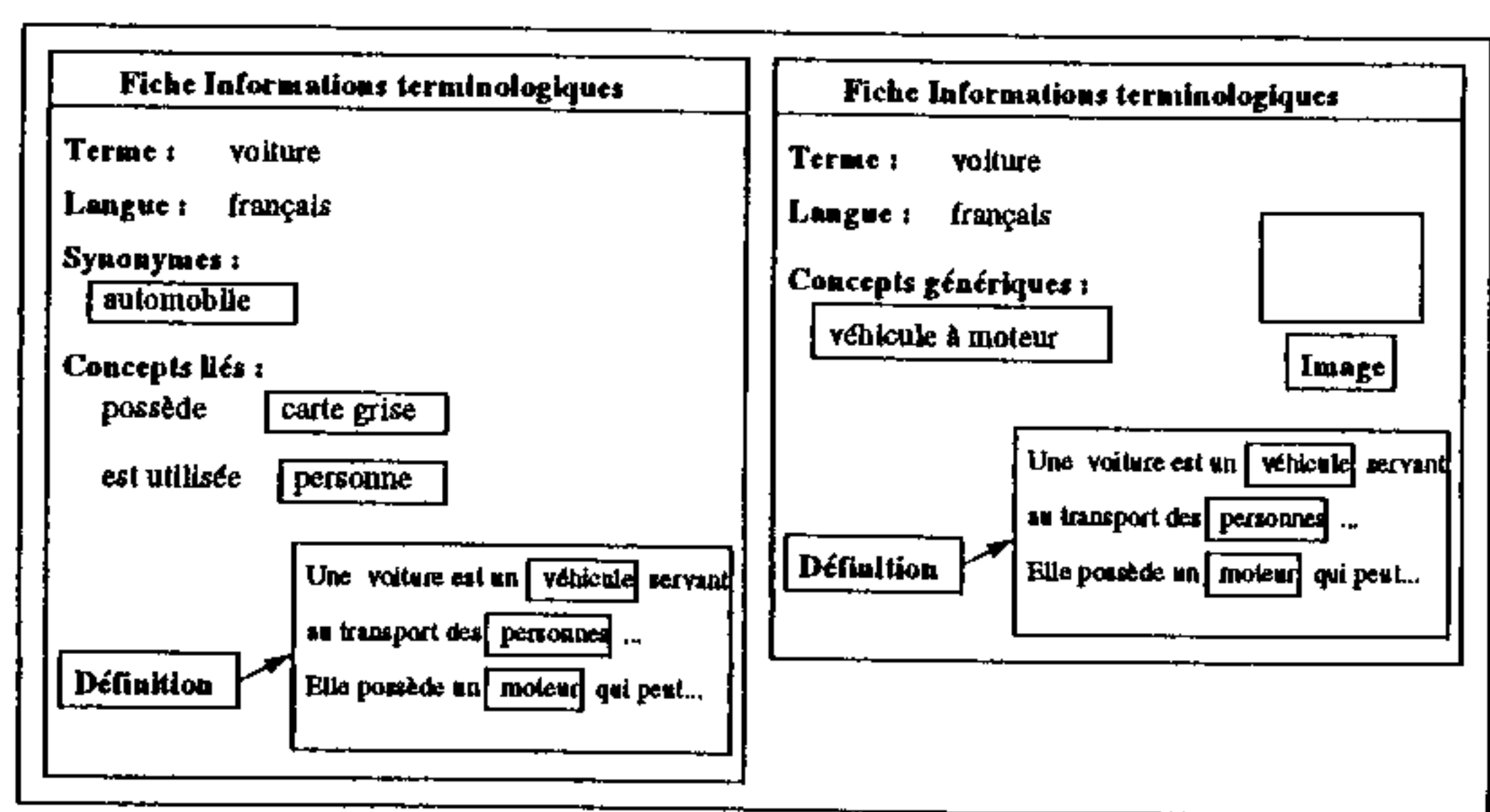


FIG. 4.11 – Instances d'un même objet documentaire.

Nous pouvons écrire les relations suivantes :

$OD_{ij} \subseteq OD_i$ et $BHD_i \subseteq BHD$

A chaque point de vue est associé une base d'hyperdocuments personnalisée qui est une instance de la base d'hyperdocuments globale. L'utilisateur a le choix entre deux possibilités

de consultation :

- Il peut accéder uniquement au sous domaine de la base d’hyperdocuments qui constitue son centre d’intérêt matérialisé par son point de vue; dans ce cas, tous les liens qui ne l’intéressent pas sont désactivés.
- Il peut également accéder aux autres documents de la base qui ne font pas partis de son domaine. Dans ce cas, les liens qui correspondent à son point de vue se distinguent des autres par un marquage spécial.

Ces liens peuvent être considérés comme des liens contextuels à destination fixe (cf. 4.2.3 page 72).

Nous pensons que la notion de point de vue peut être une solution au problème de désorientation auquel se trouve confrontés les utilisateurs lors de la consultation des hypertextes. En effet, le choix d’un point de vue permet de dériver une vue partielle de la base d’hyperdocuments qui contient les informations qui sont mieux maîtrisées par l’utilisateur; ceci limite la surcharge cognitive, source de désorientation [CON 87] [HAL 87].

4.4.2 Le processus d’hypérisation

L’hypérisation est le travail qui permet de transcrire un document textuel linéaire en réseau de nœuds hypertextes [FRI 88]. Pour nous, l’hypérisation permettra de transformer la BdC en une base d’hyperdocuments (BHD). Deux aspects sont à considérer dans ce processus d’hypérisation : la génération de l’interface hypertextuelle de consultation des concepts de la BdC et la génération des hyperdocuments correspondant aux informations terminologiques de type hypertexte dont les mécanismes sont décrits ci-dessous.

Il existe de nombreux outils permettant de créer des interfaces hypertextuelles de consultation de bases de données (DOHQ, O2 Web, Intrabuilder, ...) qui peuvent être utilisés pour la création de l’interface de la BdC.

Cependant, aucun d’eux n’est adapté à la génération d’hypertextes correspondant aux informations terminologiques de type hypertexte, dont le contenu textuel est stocké séparément de ses liens.

Nous présentons ci-dessous un algorithme d’hypérisation de l’information terminologique

de type hypertexte Définition; l'hypertexte cible est un document HTML. Le contenu informatif textuel est fourni par l'attribut définition de la relation Concept, et les liens par la relation Lien.

Ce algorithme a été utilisé pour l'hypérisation des informations terminologiques Définition et Contexte de la maquette de l'application Medtrad portant sur les plantes médicinales (cf. 6.12 page 140).

```

ALGORITHME hypérise(i)
DEBUT
  Ouvrir_bd(ALL)
  texte ← SELECT définition
          FROM Concept
          WHERE numconcept = i;
  Copier_fichier(définition.html, texte)
  Ouvrir_fichier(définition.html, "w")
  tablelien ← SELECT début, longueur, étiquette, numui2
              FROM Lien
              WHERE (type = définition) AND numui IN
                  (SELECT numui
                   FROM Unité_informative
                   WHERE numconcept = i) ;
  TANTQUE EOP(définition.html) différent de VRAI
  | POUR t = 1, Nblien
  | | début = tablelien[t, 1]
  | | longueur = tablelien[t,2]
  | | Aller_a(définition.html, début)
  | | ancresource = Créer_ancre(définition.html, début, longueur)
  | | Insérer_lien(définition.html, ancresource, étiquette, tablelien[t, 4])
  | | Ajouter_fichier("Unité informative", "nomu"i, "définition.htm")
  | FINP
  FINT
  Fermer_bd(ALL)
  Fermer_fichier(définition.html)
FIN

```

Soit C un concept de numéro i.

La procédure Créer_ancre permet de créer une ancre à partir de la position de début et de la longueur de l'ancre d'un lien.

La procédure Insérer_lien permet de créer un lien HTML.

La procédure Ajouter_fichier ajoute un fichier hypertexte dans la table Unité_informative.

Le processus d'hypérisation peut être abordé selon deux approches :

- La première approche consiste à transformer toute la BdC en une base de documents hypertextes avant une session d'utilisation; les accès à la BdC sont alors interdits. Au cours des sessions ultérieures, la cohérence de BdC est vérifiée par rapport à son état précédent. En cas d'anomalie, la BHD est régénérée totalement ou en partie.

- La seconde approche est une hypérisation dynamique, i.e. la base d'hyperdocuments est construite au fur et à mesure de la navigation dans l'hyperespace. Ainsi sa structure logique n'existe qu'au moment de la consultation des hyperdocuments. Lors du déclenchement d'un lien (lien à destination contextuelle), deux cas de figure peuvent se présenter :
 - soit l'unité informative destination a déjà été générée, la consultation est donc immédiate,
 - soit elle n'a pas été générée, alors il y aura consultation de la BdC pour la génération de l'unité informative destination.

Un autre problème se pose, celui du choix du système hypertexte. Les deux possibilités suivantes ont été envisagées :

- Utiliser un Système hypertexte expert (Hyper BD amélioré) [SAI 94]
Ce système procède par marquage des liens. Le contenu informatif des hyperdocuments est indépendante de l'aspect logique modélisé par les liens. Il induit une souplesse d'utilisation. De plus, il offre la possibilité d'évoluer dans un environnement expert (Prolog) afin de programmer les boutons ou de définir des variables dynamiques.
- Utiliser le Web, système hypermédia
Dans ce système, les documents hypertextes sont créés grâce au langage HTML. HTML insère dans un document des balises de formatage qui indiquent aux navigateur (i.e. les programmes capables de lire et d'interpréter les documents HTML) la façon dont le document doit être présenté ainsi que les liens à établir entre plusieurs documents. Il est ainsi possible de créer des documents lisibles sur n'importe quelles plates-formes dans une architecture de type client/serveur tel que Internet. Il existe également des langages (tels que Java, JavaScript, VBScript) permettant d'adjoindre du dynamisme aux documents Web.

L'inconvénient du à l'utilisation du langage HTML est qu'il ne fait pas la différence entre le contenu informatif et la structure logique (structuration des nœuds et des liens) des hyperdocuments; du fait que les liens soient directement insérés dans le contenu informatif ne facilitent pas la maintenance des hyperdocuments.

Par exemple la suppression d'une partie d'un hyperdocument référencée par d'autres hyperdocuments, génère des incohérences dans la BdC. Afin de garantir la cohérence de la base d'hyperdocuments, il faudrait parcourir un à un le code source de ces hyperdocuments et supprimer leurs liens qui pointent vers la partie à supprimer de l'hyperdocument.

L'approche que nous avons adoptée :

Nous préférons la deuxième approche basée sur l'utilisation du langage HTML qui s'est imposé comme norme mondiale (grâce à l'essor du réseau Internet) pour la description de la structure et du contenu des documents hypertextes pour le Web.

Cependant, comme dans la BdC le contenu informatif des informations terminologiques de type hypertexte est indépendante de leur aspect logique (§3.4), les documents HTML seront obtenus par un processus d'hypérisation (cf. annexe A).

Pour des raisons d'efficacité, nous avons opté pour une hypérisation dynamique. Mais cette hypérisation n'est pas systématique; elle n'est effectuée à l'activation d'un lien référant un hypertexte que si l'hypertexte n'a pas encore été généré.

La base d'hyperdocuments ainsi générée ne sera mise à jour que si la BdC à été modifiée.

4.4.3 Maintenance de la BdC

Dans cette partie, nous présentons un algorithme de mise à jour de la base d'hyperdocuments (BHD), ainsi que les mécanismes de vérification de la cohérence de cette base.

La figure 4.12 modélise la structure logique d'un document hypertexte. A chaque concept générique est associé un document constitué de plusieurs unités informatives reliées les unes aux autres par des liens de composition. Une unité informative permet de représenter un concept fils; elle peut être composée d'autres UI. La base d'hyperdocuments est constituée par l'ensemble des documents ainsi définis.

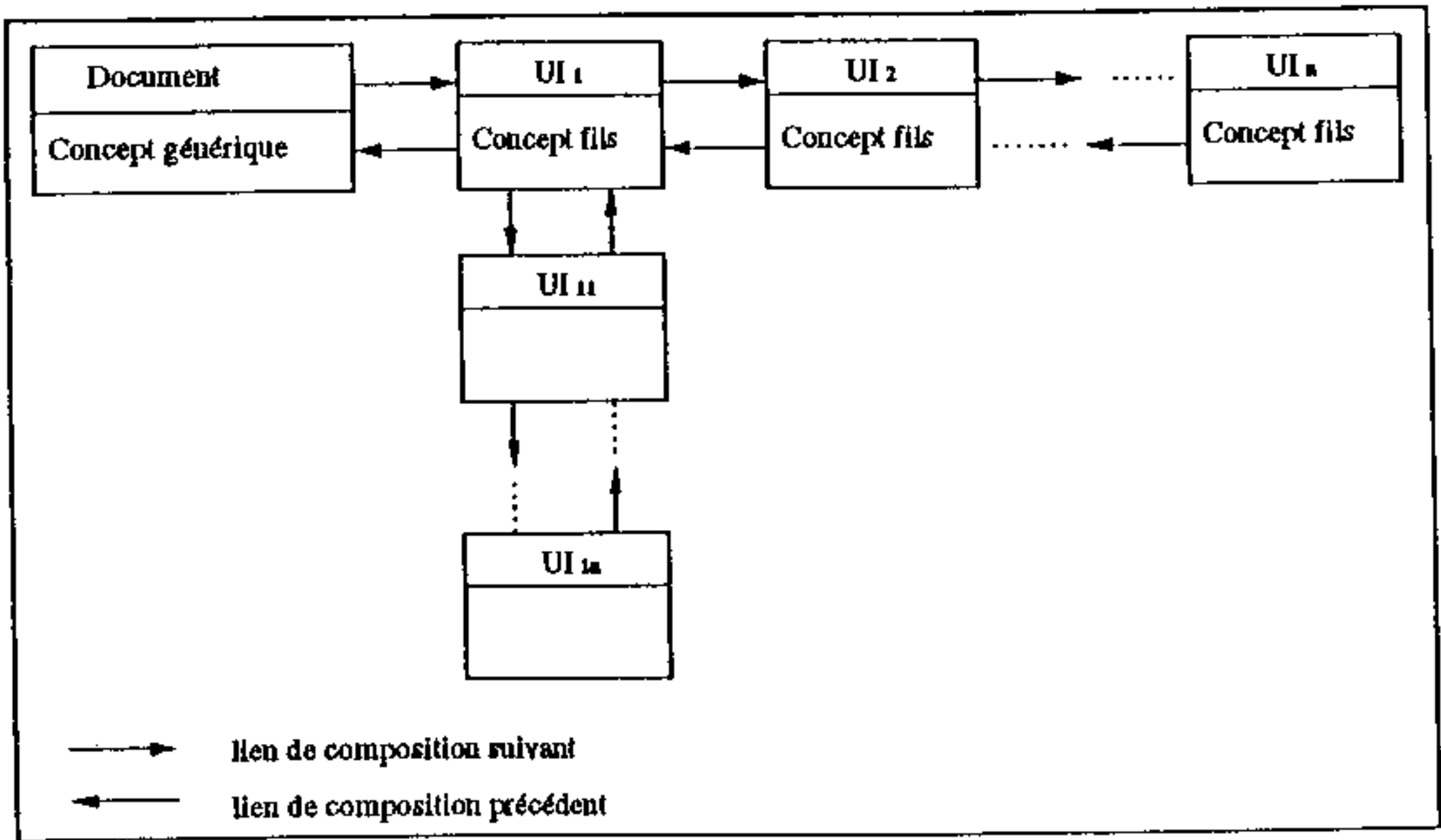


FIG. 4.12 – Structure logique d'un document.

4.4.3.1 Mise à jour de la base d'hyperdocuments

Nous pouvons envisager deux cas de mise à jour : la mise à jour due à une modification des données de la BdC, et la mise à jour due à un changement de point de vue.

Toute mise à jour des données de la BdC entraîne automatiquement la mise à jour totale ou partielle de la BHD globale et des BHD dérivées. Le changement d'un point de vue consiste à ajouter ou à supprimer un élément du point de vue. Cette modification entraîne aussi la mise à jour totale ou partielle de la BHD dérivée.

La mise à jour totale consiste à régénérer totalement une base d'hyperdocuments; elle est utilisée lorsque les objets documentaires à mettre à jour dépasse un certain seuil.

La mise à jour partielle consiste à mettre à jour un nombre faible d'objets documentaires, d'où un problème de cohérence de la base d'hyperdocuments par rapport à la BdC qu'il faudra gérer.

L'intérêt d'une mise à jour partielle est que la modification de la BdC peut avoir une incidence sur un nombre relativement faible d'objets documentaires.

Par exemple, la suppression ou l'ajout d'un synonyme à un concept implique la mise à jour d'un seul objet documentaire. Les deux types de mise à jour résolveraient ce problème; cependant, une mise à jour partielle est moins coûteuse en temps machine.

Nous proposons dans les §suivants des algorithmes de mise à jour de la base d'hyperdocuments.

Soient :

- LOD la liste d'objets documentaires à mettre à jour lorsque la BdC (ou un point de vue) a été modifiée.
- NOD la taille de LOD
- LODBdC la liste des OD de toute la BdC
- NODT la taille de LODBdC
- Seuil critique le nombre d'objet documentaire à ne pas dépasser pour une mise à jour partielle; il dépend de la puissance de traitement de la machine sur laquelle l'utilisateur travaille.

Si les informations terminologiques ou les caractères définitoires de la BdC qui ont été mis à jour figurent dans un point de vue, la base d'hyperdocuments dérivée de ce point de

ALGORITHME MAJBHD

```
DEBUT
  Réception de la demande de MAJ de la base d'HD
  Déterminer la nature de la MAJ (ajout, suppression, ...)
  Déterminer NOD et LOD
  SI NOD > Seuil critique ALORS
    Effectuer la mise à jour totale (MAJT)
  SINON
    Effectuer la mise à jour partielle (MAJP)
  FINSI
FIN
```

ALGORITHME MAJT

```
DEBUT
  i ← 0
  TANTQUE i <= NODT
    Régénérer LOdBdC[i]
    i ← i + 1
  FINT
FIN
```

ALGORITHME MAJP

```
DEBUT
  i ← 0
  TANTQUE i <= NOD
    Régénérer LOD[i]
    i ← i + 1
  FINT
  Vérifier la cohérence de la base
FIN
```

vue est aussi mise à jour.

4.4.3.2 Vérification de la cohérence

On distingue deux sortes de cohérence dans un hyperdocument : la cohérence au niveau des liens qui n'aboutissent pas et la cohérence au niveau de sa structure logique.

a) Cohérence au niveau des liens

Le problème des liens qui n'aboutissent pas parce que le système ne trouve pas l'OD destination a trois explications possibles : soit l'OD destination à été supprimé, soit il a été déplacé, soit il n'a jamais existé.

Pour déterminer ces OD, nous recherchons dans un premier temps les liens qui figurent dans le code source des hyperdocuments, ensuite nous les comparons avec ceux qui ont été stockés dans la BdC.

Prenons par exemple le cas d'une unité informative UI_j de numéro j qui contient des liens qui n'aboutissent pas; les liens théoriques qui partent de l'unité informative UI_j sont déterminés par la requête suivante :

Soit `lien_théorique` la relation qui modélise les liens de l'unité informative UI_j , générée par la requête ci-dessous :

```
SELECT numui2, étiquette, début, longueur INTO Lien_théorique
FROM   Lien
WHERE  numui = j ;
```

Cette requête engendre la relation :

`Lien_théorique(numui2, étiquette, début, longueur)`.

`numui2` étant le numéro de l'objet documentaire destination d'un lien. Les liens effectifs qui partent de l' UI_j peuvent être obtenus par l'analyse du code source de l'hyperdocument qui lui est associé. En effet, le langage HTML utilisé pour décrire la structure des documents hypertextes pour le Web, délimite les liens par des balises dont la syntaxe se présente de la façon suivante :

- lien vers un document complet : `<A HREF="UID"> ancre </A>`
- lien vers une partie d'un document : `<A HREF="UID#étiquette"> ancre </A>`
- définition du point d'ancrage dans l'UID : `<A NAME="étiquette">`

où :

- UID : désigne l'UI destination du lien,
- étiquette : désigne le point d'ancrage dans l'UID,
- ancre : désigne la zone cliquable dans l'UI source.

L'analyse de ces chaînes de caractères qui modélisent un lien en HTML permet de déterminer les paramètres nécessaires à la reconnaissance des liens effectifs contenus dans un document HTML.

Soit la relation `Lien_effectif(UID, étiquette, ancre)` dont les attributs sont ces paramètres.

La comparaison entre ces liens (modélisés par la relation `lien_effectif`) et les liens théoriques (modélisés par la relation `Lien_théorique`) permet de déterminer les liens qui n'aboutissent pas, ainsi que la cause de ce problème qui peut être l'un des deux cas suivants :

- nom de l'UID et/ou de l'étiquette mal orthographié

Dans l'exemple ci-dessous, nous présentons le codes source HTML se rapportant à la définition du concept voiture après hypérisation :

```
<DT> <H1> Définition: </H1> <DD>
Une voiture est un <A HREF="f1.html"> véhicule </A> servant au
transport des <A HREF="f2.html"> personnes</A> et des choses. Elle
possède un <A HREF="f3.html#Caractères définitoires de type concept">
moteur </A> qui peut être diesel ou à essence, ...
</DD> </DT>
```

Le troisième lien `<A HREF="f3.html#Caractères définitoires de type concept"> moteur</A>` à pour ancre source "moteur" et comme ancre destination "Caractères définitoires de type concept" localisé dans le document f3.html; son code HTML se présente comme suit :

```
<DT> <H1> Caractères: </H1> <DD>
<A NAME="Caractères définitoires de type concept">
Caractères définitoires de type concept :
<A HREF = "f4.html">Carrosserie </A>
<A HREF = "f5.html">Roue </A>
<A HREF = "f6.html">Chassis </A>
<A HREF = "f7.html">Moteur </A> ...
</DD> </DT>
```

La figure 4.13 schématise les hypertextes associées à ces deux documents.

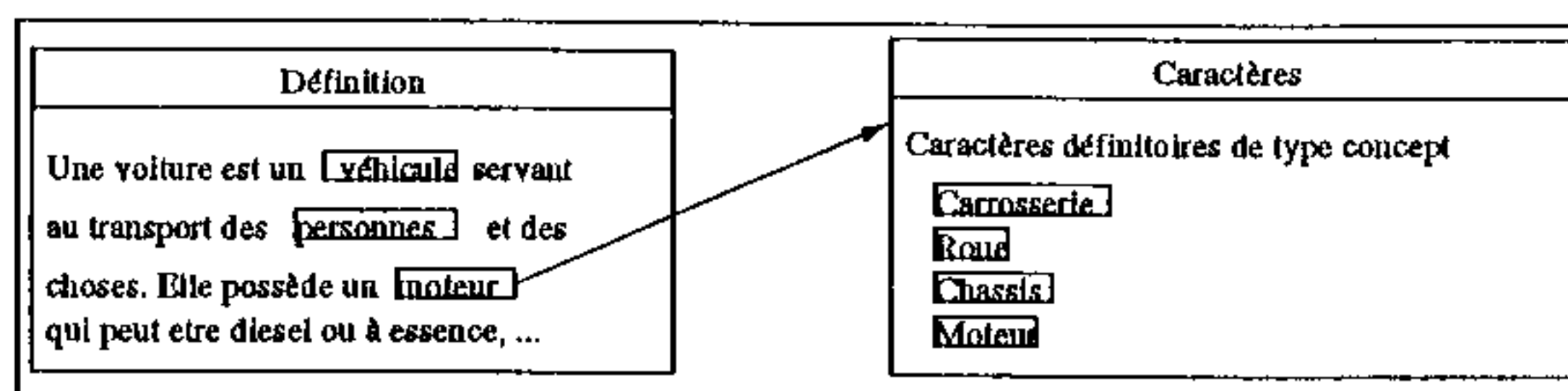


FIG. 4.13 – Lien vers une partie d'un document.

Dans la figure 4.14, il y a une relation de correspondance entre les attributs de ces deux relations :

- numi2 est le numéro de l'UID (3 est le numéro de f3.html),
- le couple d'attribut (début, longueur) permet de déterminer l'ancre source d'un lien, par exemple l'ancre moteur commence au 94 ième caractère et se termine au 99 ième caractère ($99 = 94 + 5$):

- l’attribut étiquette a la même signification dans les deux relations.

La comparaison entre les valeurs de ces attributs permet de déterminer les liens qui n’aboutissent pas.

Lien_théorique				Lien_effectif		
numui2	étiquette	début	longueur	UID	étiquette	ancre
1		20	7	f3.html	Caractères de type concept	moteur
2		54	8			
3	Caractères de type concept	94	5			

FIG. 4.14 – Modélisation des liens en HTML.

b) Cohérence au niveau de la structure logique d’un hyperdocument

Le problème de la cohérence au niveau de la structure logique d’un hyperdocument est très délicate à gérer. Il est dû à la mise à jour du contenu informatif des OD.

Nous présentons ci-dessous deux cas de cohérence sémantique. Le premier cas est une cohérence au niveau des liens de composition suite à la suppression d’une unité informative; le deuxième cas est une cohérence au niveau du contenu informatif après la suppression d’une de ses parties.

Dans les parties qui suivent, nous analysons quelques cas de figures qui illustrent le problème de cohérence sémantique de la base d’hyperdocuments.

1. Supression d’un document

La suppression d’un document doit garantir la cohérence de la base d’hyperdocuments. Cette suppression est réalisée comme suit :

- Rechercher toutes les UI de ce document
- Supprimer toutes ces UI
- Supprimer les auteurs de ce document
- Supprimer le document

2. Suppression d’une unité informative

La suppression d’une UI doit garantir la cohérence du document par rapport aux liens de composition et la cohérence par rapport aux liens de référence.

- *Cohérence par rapport aux liens de composition*

Les trois cas suivants peuvent être envisagés :

- Suppression de la première UI d’un document (UI_1)

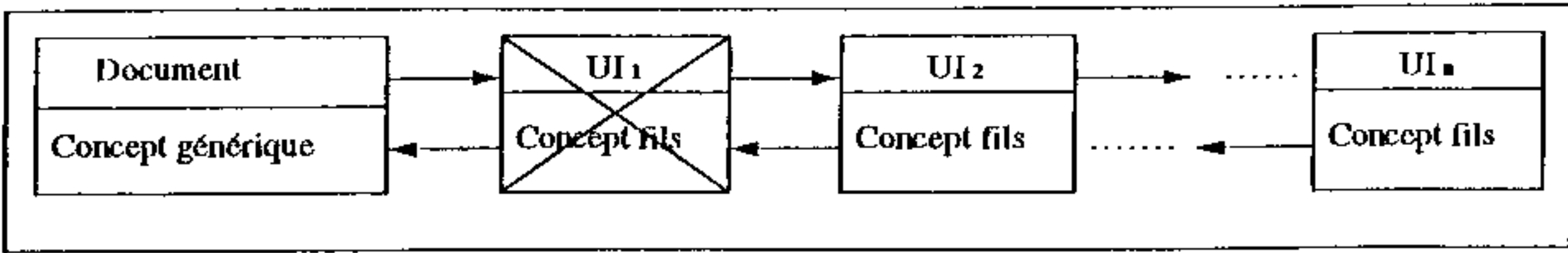


FIG. 4.15 – *Suppression de la première UI d’un document.*

Après la suppression de l’ UI_1 , le lien de composition suivant du document pointe sur l’ UI_2 et le lien de composition précédent de l’ UI_2 point sur le document.

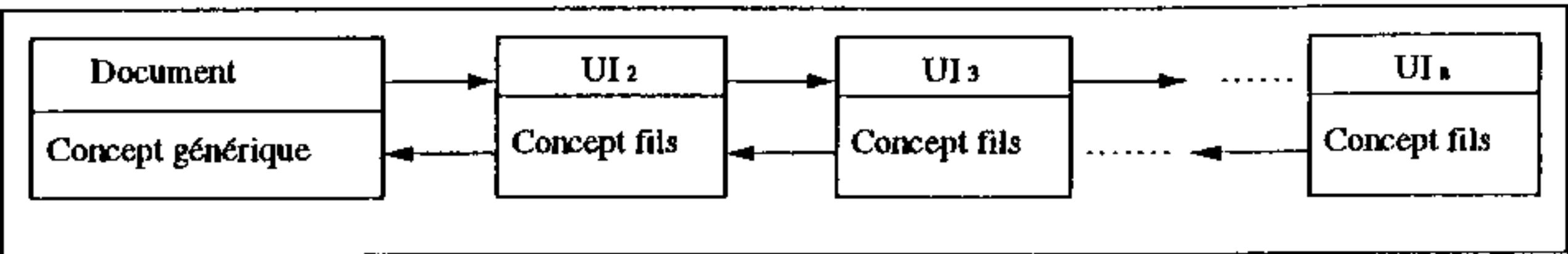


FIG. 4.16 – *Après la suppression de la première UI d’un document.*

- Suppression d’une UI d’indice j insérée entre deux UI d’indices $j-1$ et $j+1$
Après la suppression de l’ UI_j , le lien de composition suivant de l’ UI_{j-1} pointe sur l’ UI_{j+1} et le lien de composition précédent de l’ UI_{j+1} point sur l’ UI_{j-1} .

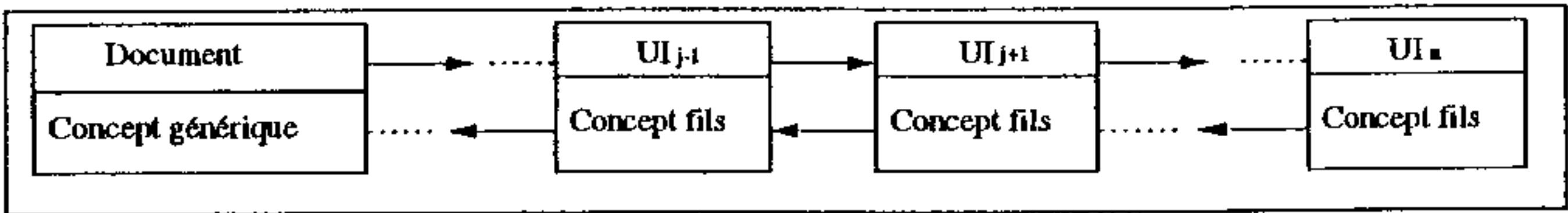


FIG. 4.17 – *Après la suppression d’une UI d’indice différent de 1 et n d’un document.*

- Supression de la dernière UI
Après la supression de l’ UI_n , l’ UI_{n-1} devient la dernière UI du document.
- *Cohérence par rapport aux liens de référence:*

La suppression d’une UI entraine la suppression de tous les lien qui partent de cette UI ou qui arrivent dans cette UI. Soient UI_j l’UI à supprimer et UI_k une UI qui contient des liens dont la destination est UI_j .

Cette suppression est réalisée comme suit :

- Rechercher tous les liens de UI_j et ceux de UI_k qui arrivent à UI_j
- Supprimer ces liens
- Mettre à jour l’unité informative UI_k
- Supprimer l’unité informative UI_j

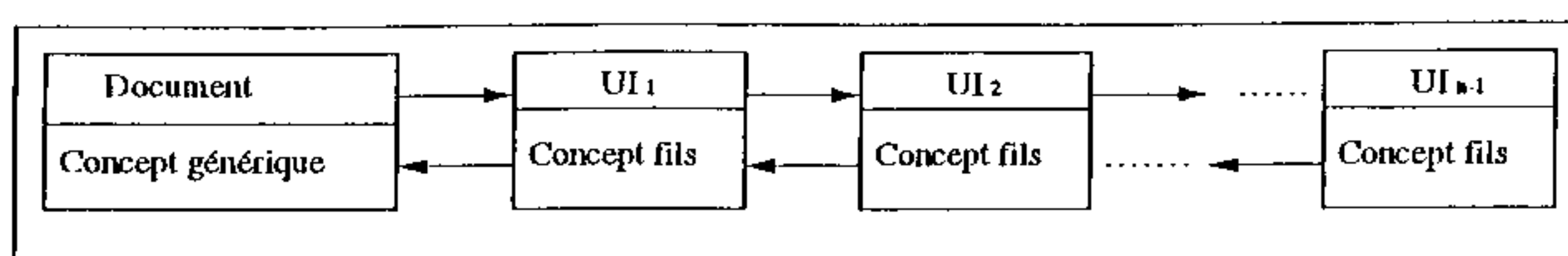


FIG. 4.18 – Après suppression de la dernière UI d'un document.

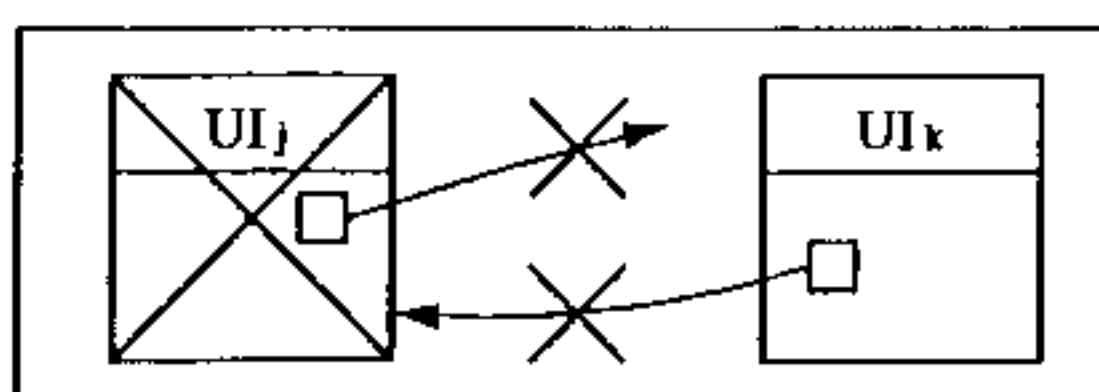


FIG. 4.19 – Suppression d'une UI qui contient ou qui est la destination d'un lien de référence.

3. Suppression d'une partie d'une unité informative

Soient x et y respectivement la position du premier et du dernier caractère de la zone à supprimer (Fig. 4.20). On peut envisager deux cas de figure :

- La partie à supprimer ne contient aucune ancre (source ou destination) d'un lien; dans ce cas la suppression de cette partie n'aura aucune incidence sur la cohérence de la base.
- La partie à supprimer contient des ancres (source ou destination) d'un lien interne ou externe

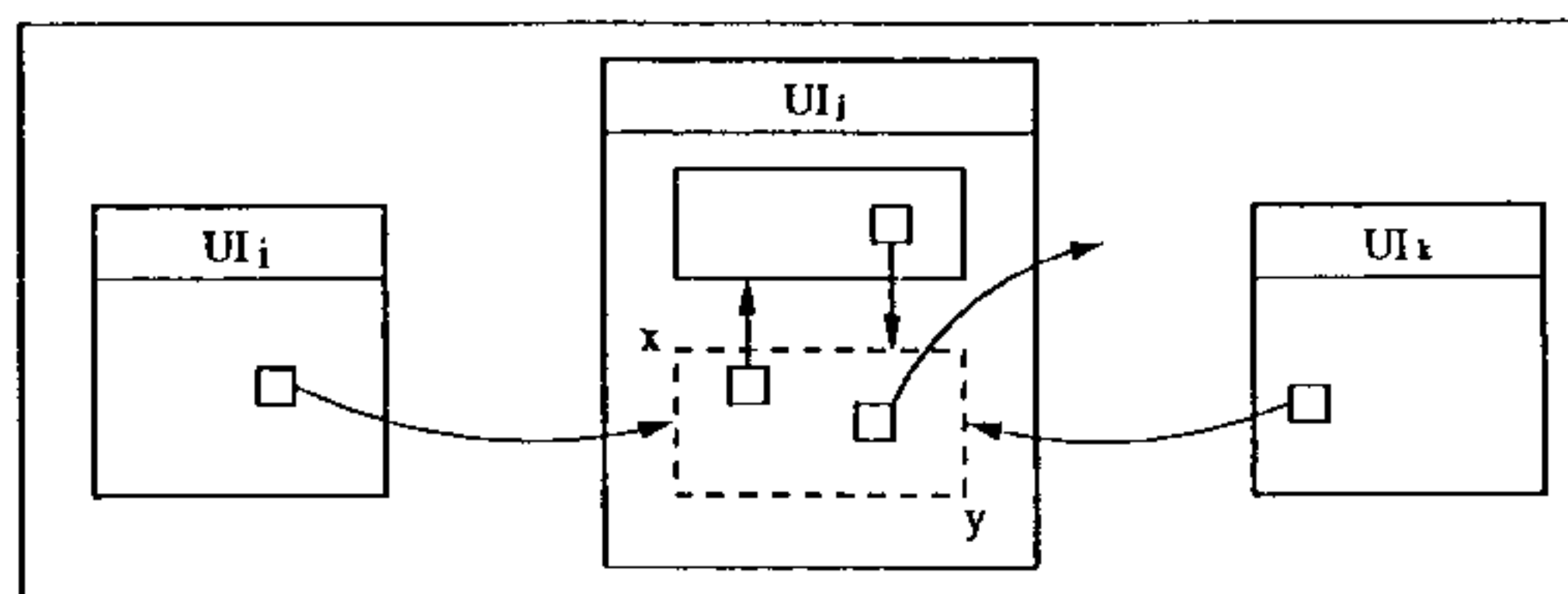


FIG. 4.20 – Suppression d'une partie d'une UI.

Cette suppression est réalisée comme suit :

- Rechercher toutes les UI ayant des liens dont l'ancre (source ou destination) est située dans la partie à supprimer
- Supprimer ces liens
- Mettre à jour ces UI
- Supprimer la partie de l'UI délimitée par x et y

4. Suppression d'une UI composante d'une autre UI

Par exemple la suppression de l' UI_{1j} (Fig. 4.12).

Le principe est le même si on remplace l'unité informative UI_1 par le document et l'unité informative à supprimer UI_{1j} par UI_1 .

4.5 Conclusion

Dans ce chapitre, nous avons proposé un modèle de gestion hypertexte des concepts d'une BdC. Ce modèle est basé sur celui de document hypertexte expert proposé par [FOR 94].

Pour la conception des hyperdocuments, nous avons proposé la séparation de l'aspect informatif de l'aspect logique des documents, l'hypertexte résultant est obtenu par un processus d'hypérisation. Cette approche présente comme avantages : la maintenance facile et la vérification de la cohérence (cohérence sémantique et cohérence au niveau des liens qui n'aboutissent pas) de la base d'hyperdocuments.

La consultation des concepts peut être réalisée soit par navigation à l'aide d'une interface hypertextuelle basée sur la notion de fiche terminologique, soit par une requête, suivie de navigation.

Nous avons également proposé la notion de point de vue qui permet d'avoir une vue personnalisée de la base d'hyperdocuments correspondant au domaine d'activité, ou au centre d'intérêt bien maîtrisé par un utilisateur.

Nous pensons que ces deux notions : point de vue et accès à la BdC à l'aide de requêtes, permettent de limiter le nombre d'objets documentaires lors d'une consultation; elles peuvent être une solution au problème de désorientation auquel se trouvent confrontés les utilisateurs lors de la consultation des hyperdocuments.

Chapitre 5

L'implémentation de la BdC

5.1 Introduction

Dans ce chapitre, nous présentons le schéma de fonctionnement général de la BdC. La liaison entre les systèmes de gestion de données : le SGBD et le système hypertexte (SIIT) qui gèrent la BdC, est assurée par un système de gestion et d'exploitation hypertexte d'une base de concepts (que nous notons CoDB-Web).

Après avoir présenté les fonctionnalités du système CoDB-Web, nous décrivons l'environnement de développement de la BdC; cet environnement est celui du système hypermédia WWW, moyen d'accès aux ressources d'Internet [FOR 96]. Nous terminerons par la proposition d'une méthodologie de conception d'une BdC.

5.2 Architecture générale de la BdC

La figure 5.1 présente les principales composantes de la BdC i.e. la BD qui stocke les concepts, la BHD qui permet l'exploitation hypertextuelle de la BdC, et le système CoDB-Web (Concept DataBase) qui sert d'interface entre le SGBD et le Système hypertexte.

5.2.1 Présentation fonctionnelle

Les deux premières composantes ayant été décrites dans les chapitres précédents (§ 3.4.4 pour l'aspect base de données et § 4.3 pour l'aspect hypertexte), nous ne décrivons que le système CoDB-Web dans cette partie.

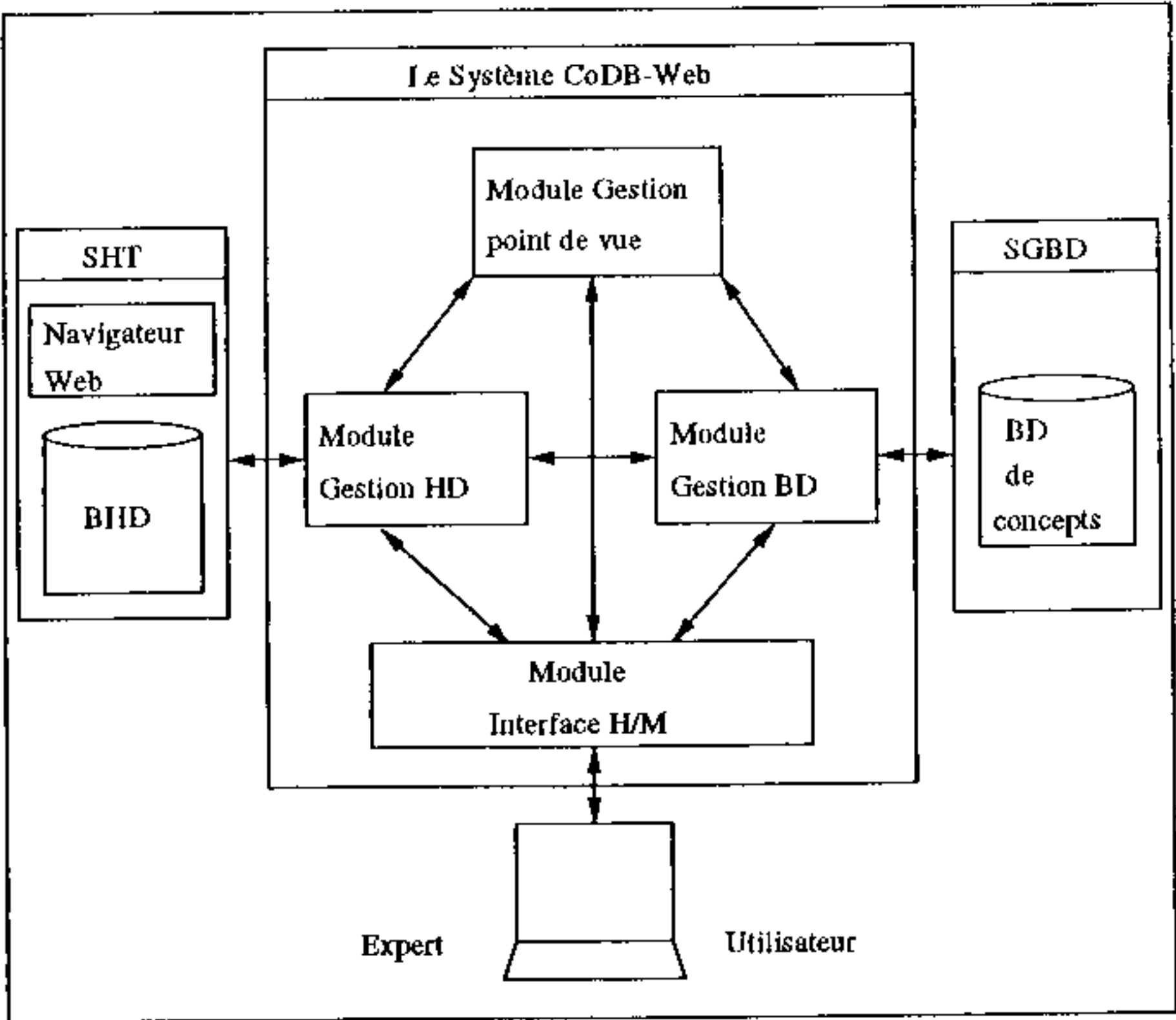


FIG. 5.1 – Schéma de fonctionnement général de la BdC .

Le système CoDB-Web est l'interface de communication entre le SGBD qui gère la base de données de concept et le système hypertexte qui gère la base d'hyperdocuments; il est constitué de quatre modules interconnectés qui sont :

1. Le module interface Homme/Machine est chargé de la gestion du dialogue entre l'utilisateur (ou l'expert) et la BdC, en vue d'exécuter des actions ou d'afficher des messages (message d'erreurs par exemple). Les actions de l'utilisateur s'expriment en terme de demande de consulation des concepts de la BdC soit par navigation, soit par des requêtes suivies de navigation.

Les actions de l'expert sont des demandes de création (génération de tables, saisie de concepts, etc.) et de maintenance de la BdC (Ajout ou suppression de concepts, vérification de la cohérence de la BdC, etc.).

2. Le module gestion des points de vue est chargé de la gestion des points de vue. Il a la charge de faciliter la saisie et la mise à jour des éléments d'un point de vue (ajout, suppression d'un élément d'un point de vue).
3. Le module gestion de la BD de concepts est chargé de la gestion et de l'exploitation de la BD de concepts. Il a la charge de générer les tables et les vues relationnelles, de faciliter la saisie des concepts, et de maintenir la BD (ajout, suppression, modification de concepts). C'est le seul module qui communique directement avec les concepts stockés dans la BD; il est par conséquent en interaction avec les autres modules.
4. Le module gestion de la BHD assure la gestion et l'exploitation de la base d'hyperdocuments. Il a la charge de transformer les données de la BdC en documents hypertextes

(processus d'hypérisation), de générer les BHD personnalisées et de maintenir la BHD (ajout, suppression et modification d'hyperdocuments).

La figure 5.2 présente les différentes fonctions de ces modules; les arcs symbolisent la communication entre ces modules.

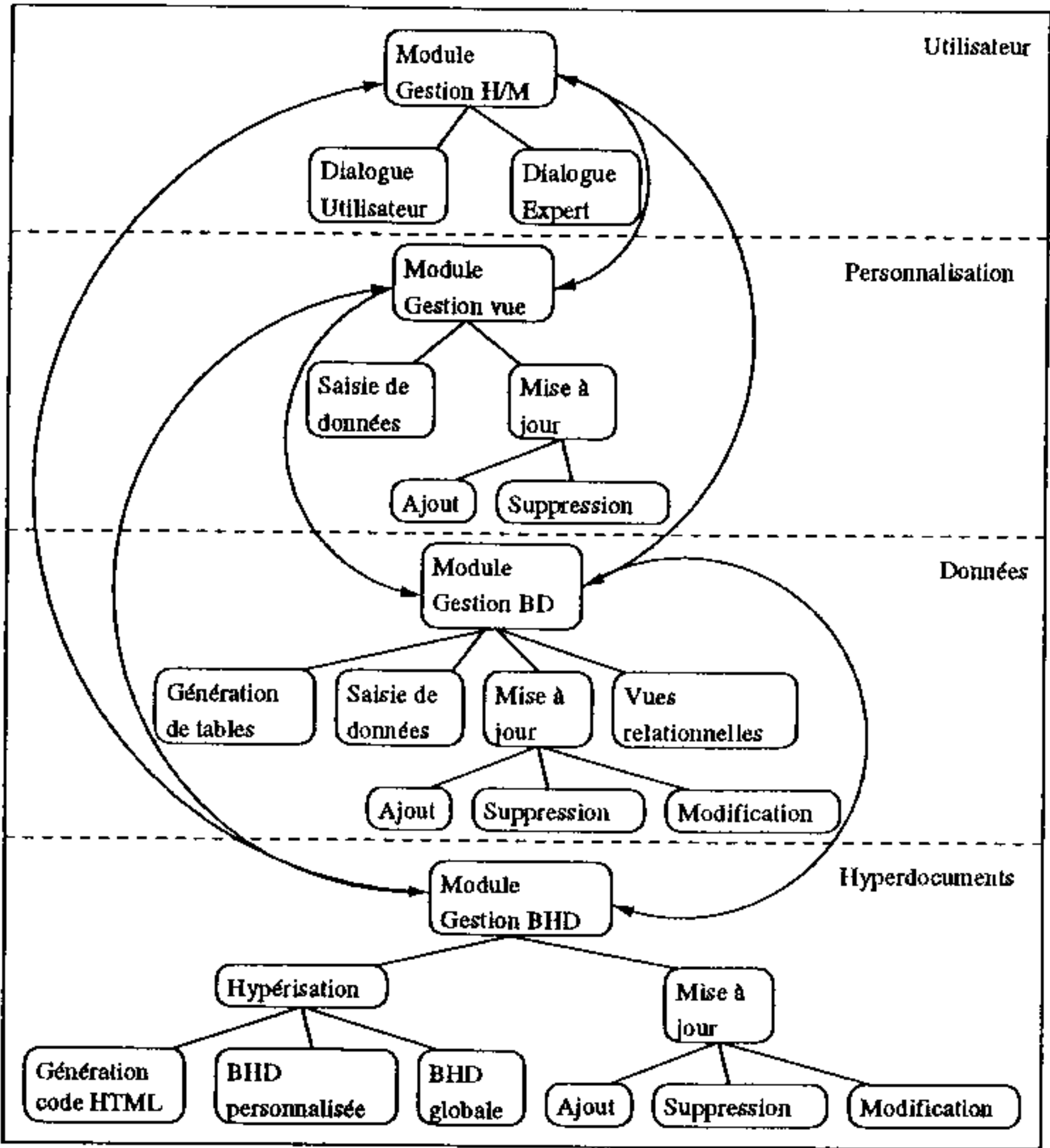


FIG. 5.2 – Architecture logicielle de CoDB-Web.

5.2.2 Circulation des informations entre CoDB-Web et son environnement

La figure 5.3 présente l'environnement de CoDB-Web constitué par le SGBD et le système hypertexte.

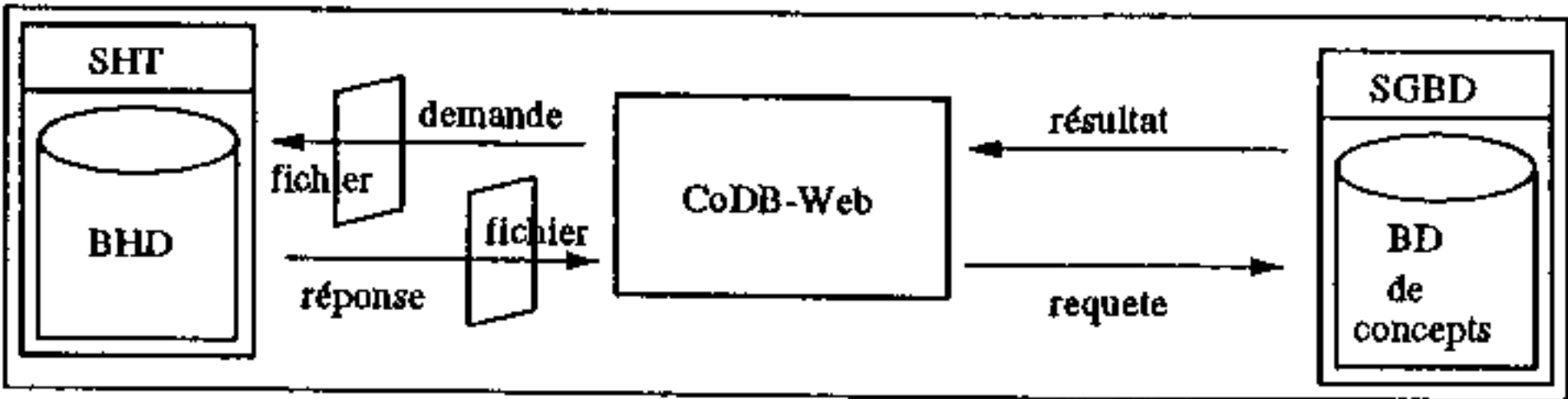


FIG. 5.3 – Circulation des informations entre CoDB-Web et son environnement.

- La communication entre CoDB-Web et le SGBD s'exprime en terme de requête de l'un et de résultat de l'autre.

Le système CoDB-Web s'adresse au SGBD par l'intermédiaire du module Gestion BD, soit pour la gestion des informations de la BdC, soit pour une interrogation.

Concrètement cette communication s'effectue grâce à des procédures prédéfinies d'accès à une BD de JavaScript (cf. § 5.3.3) qui est le langage de développement de CoDB-Web.

- La communication entre CoDB-Web et le SHT s'exprime en terme de demande de l'un et de réponse de l'autre.

Le système CoDB-Web s'adresse au SHT par l'intermédiaire du module Gestion BHD, soit pour la gestion de la BHD, soit pour l'accès à un hyperdocument donné.

Cette communication est réalisée au moyen deux fichiers ASCII de transmission : un fichier pour les demandes de CoDB-Web, et un autre pour les réponses du SHT. Nous expliciterons cette approche au § 5.3.1 par le concept de "cookies HTTP".

5.3 L'environnement de développement de la BdC

L'environnement de développement de la BdC est celui du Word Wide Web (WWW) et de son langage de conception de documents hypertextes HTML. Son cadre d'exécution est celui d'une architecture de type client/serveur utilisant le réseau informatique mondial Internet. Lorsque le réseau est local, on parle plutôt d'Intranet [FOR 96].

5.3.1 Le Système hypertexte WWW

Le système hypertexte utilisé pour l'exploitation hypertextuelle de la BdC est WWW; il représente un moyen facile d'accès aux ressources d'Internet sans que l'utilisateur ait besoin de savoir comment s'opèrent leurs transfert et leurs stockages.

Internet est un ensemble de réseaux informatiques interconnectés, répartis dans le monde entier et permettant aux utilisateurs de communiquer. C'est donc le cadre idéal pour rendre la connaissance stockée dans la base de concepts accessibles à un grand nombre de postes clients, indépendamment de la plate forme qui les supporte.

Le schéma de la figure 5.4 présente les différents éléments impliqués dans le fonctionnement de ce réseau mondial qu'est Internet : le client, le serveur et les protocoles utilisés.

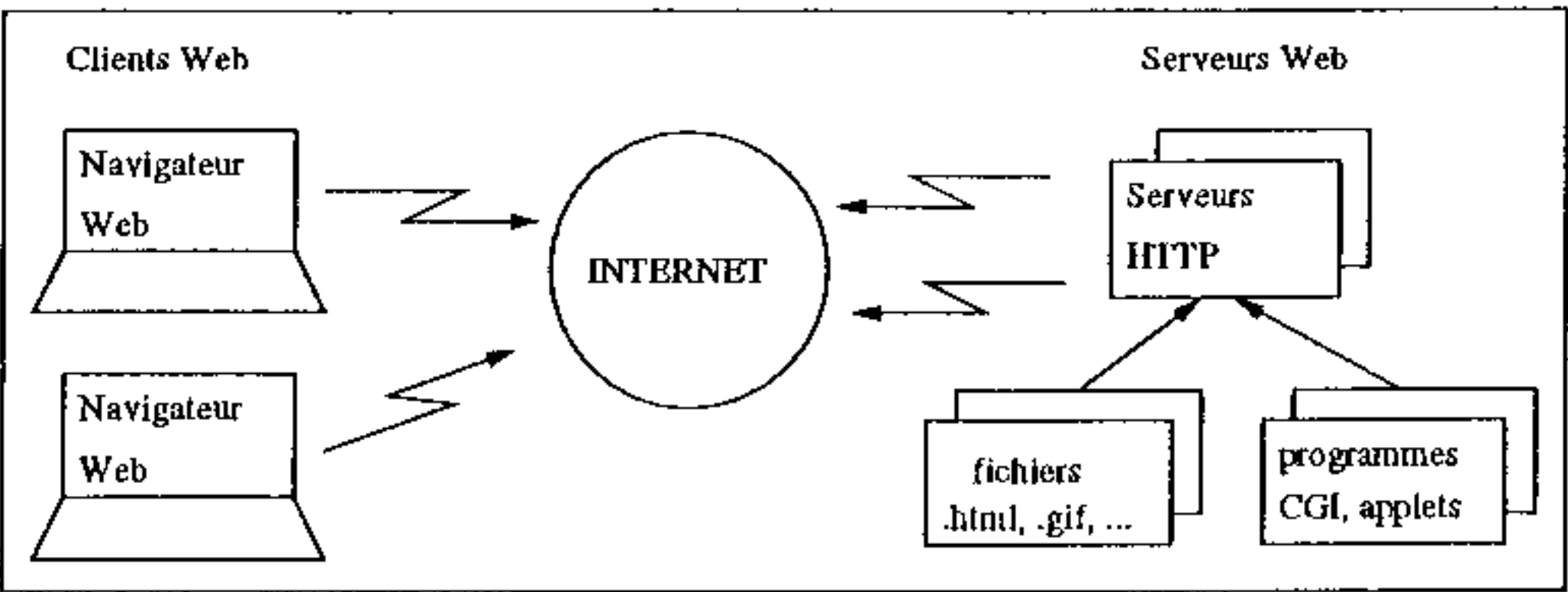


FIG. 5.4 – Architecture client/serveur.

- Le client appelé aussi browser ou navigateur est un logiciel de consultation et de présentation des documents Web s'exécutant sur toutes les plates formes (Mac, station UNIX, PC).

Le navigateur envoie des requêtes au serveur qui lui répond par le transfert de pages Web. Ce dialogue peut être régi par de nombreux protocoles d'échange de données (FTP, Gopher, HTTP, ...); le plus connu est HTTP (HyperText Transfer Protocol) dont le but est de gérer le transfert des blocs de données ou pages Web structurés en code HTML, entre le client et le serveur. Le client ayant la charge d'exécuter ce code afin d'afficher le document.

Les clients les plus connus sont Mosaic, Netscape et Internet Explorer.

- Le serveur est aussi un logiciel qui est en permanence à l'écoute des requêtes formulées par les clients. Après avoir vérifié la validité des requêtes et du droit d'accès des clients (authentification), il leur transfère les pages Web contenant du texte et/ou des images. La plupart des pages Web sont statiques, i.e. elles restent figées dans un état immuable. Des programmes se déroulant sur le serveur permettent de les rendre interactives et dynamiques. Ces programmes sont appelés programmes (ou scripts) CGI (Common Gateway Interface).

CGI est une interface standard qui gère le dialogue entre un client et un serveur HTTP. Les informations en provenance des navigateurs Web sont envoyées au serveur où elles sont traitées par les programmes CGI; les résultats élaborés par le Serveur sont envoyés au client dans des pages HTML en temps réel i.e. dans des pages HTML générées dynamiquement.

L'inconvénient des programmes ou scripts CGI est qu'ils dépendent des systèmes d'exploitation et des serveurs HTTP (ils peuvent être écrits en C, C++, Fortran, Pearl,...).

Afin de diminuer la charge du serveur, une partie de ces traitements peut être exécutée directement par le client. Dans ce cas, la réponse à un client est un programme. Ces

programmes sont écrits dans un langage exécutable par le navigateur quelque soit la plate forme matérielle sur laquelle il est utilisé. Le langage le plus connu et le plus utilisé est le langage Java; les programmes envoyés au navigateur en même temps que les page HTML sont appelés applets java; ce sont de mini-applications intégrées dans les pages HTML afin de les rendre dynamiques et interactives.

- Le concept de cookie : HTTP est un protocole de dialogue client/serveur orienté transfert de pages, or l'exécution d'une application nécessite un enchaînement de pages [ANO 96]. Les navigateurs traitent le passage d'une page à une autre comme deux applications séparées, ce qui ne permet pas de conserver le contexte (variables d'environnement, variables des applets, ...) de la première page; on dit que la communication entre client et serveur est non connectée.

D'où le concept de cookie qui est un fichier de texte ASCII, utilisé pour conserver un contexte entre deux sessions de travail d'un client avec un même serveur.

Le serveur Web peut stocker des informations dans les cookies et envoyer des requêtes au navigateur afin qu'il lui renvoie tout ou partie du contenu du cookie. De même les outils de programmation peuvent également consulter et mettre à jour ces fichiers.

C'est ce principe que nous avons retenu pour assurer la communication entre le système CoDB-Web et le Système hypertexte.

5.3.2 Le langage HTML

HTML [ANO 96] est un langage simple permettant de décrire la structure des documents hypertextes pour WWW au moyen d'un ensemble de balises; c'est une implémentation relativement simple du langage SGML (Standard Generalisation Markup Language) crée pour tenter de résoudre le problème de transport de documents sur différentes plates-formes.

5.3.2.1 Les concepts de base

Une *page* Web ou *document HTML* est composée d'un certain nombre de fichiers HTML contenant du texte et des images ou des programmes, localisés sur un périphérique de stockage accessible par le serveur. L'endroit exacte où est située une page est déterminé par son adresse appelée URL (Uniform Resource Locator). Elle est composée : du protocole d'échange (HTTP, FTP, ...), de l'adresse Internet du serveur qui diffuse la page et du chemin d'accès qui conduit au document.

Un *fichier HTML* est divisé en deux parties : un entête qui contient des informations sur le document, et un corps qui contient le texte même du document.

Chaque élément de structure du document est encadrée par une paire de *balises* : une balise d'ouverture et une balise de fermeture délimitées par les signes < (inférieur à) et > (supérieur à)

Chaque balise a un identificateur ou nom et éventuellement quelques attributs, sa syntaxe se présente de la façon suivante :

```
<Identificateur Attribut=valeur> : balise d'ouverture  
... texte affecté par la balise  
</Identificateur> : balise de fermeture
```

Exemple la balise de définition d'un lien :

```
<A HREF="ficheIT.html"> voiture </A>
```

Dans ce exemple, A est la balise (balise de lien) et "ficheIT.html" est la valeur de son attribut HREF (Hypertext REference). voiture est l'ancre du lien dont l'activation affichera sa fiche terminologique contenu dans le document destination ficheIT.html.

Il existe également des balises qui n'ont pas de balise de fermeture : par exemple la balise de rupture de ligne
 et la balise <HR> pour tracer un trait horizontal.

La structure d'un document HTML se présente comme suit :

```
<HTML>
  <!-- Ceci est un commentaire -->
  <HEAD>
    <TITLE>
      Emplacement du titre
    </TITLE>
  </HEAD>
  <BODY>
    Corps du document
    <ADRESS>
      Adresse électronique de l'auteur
    </ADRESS>
  </BODY>
</HTML>
```

où

- `<HTML>` : est le premier élément qui permet d'encadrer tout le fichier HTML (obligatoire)
- `<HEAD>` : délimite la zone d'entête
- `<TITLE>` : définit un titre
- `<BODY>` : encadre le corps du document
- `<ADRESS>` : bloc d'information sur l'auteur du document
- `<!-- Ceci est un commentaire -->` : est un texte de commentaire

Le point d'entrée dans un document Web est appelé *page de garde* ou page d'accueil; elle contient en général un message de bienvenue, des renseignements sur l'application modélisée par le document, un sommaire constitué de liens de référence vers les autres pages du document, et l'adresse électronique de l'auteur du document.

En HTML, les liens peuvent servir à relier deux éléments d'une même page ou deux pages de deux documents différents localisés sur le même site, ou sur deux sites différents. La syntaxe de ces liens se présente de la façon suivante :

- lien vers un document complet : `<A HREF="URLD"> ancre </A>`
- lien vers une partie d'un document : `<A HREF="URLD#étiquette"> ancre </A>`
- définition du point d'ancrage dans l'OD destination : `<A NAME="étiquette">`

où :

- URLD : désigne l'adresse (URL) de l'objet documentaire destination du lien,
- ancre : désigne la zone active, cliquable dans l'objet documentaire source,
- étiquette : appelée aussi ancre d'arrivée, désigne la zone inactive qui est le point d'ancrage dans l'objet documentaire destination.

La distinction entre les liens à l'intérieur d'un même système (liens intra-système) et les liens entre deux systèmes différents (liens inter-systèmes) est effectuée à travers leur URL.

Lorsque les liens sont intra-système, on indique seulement le chemin d'accès au document dans l'URL destination.

Exemple la figure 5.5 présente le code de la page d'accueil de la BdC Medtrad qui est une base de concepts sur les plantes médicinales.

```
<HTML>
<HEAD>
  <TITLE> Le Système Medtrad </TITLE>
</HEAD>
<BODY>
  <CENTER>
    <FONT size=6 color=#FF0000>
      <HR>
      <H1> Le Système Medtrad
      <BR>
      Médecine Traditionnelle africaine à base de plantes médicinales
    </H1>
    </FONT>
  </CENTER>
  <EM> Bienvenue dans le Système Medtrad ! </EM>
  <HR>
  <BLOCKQUOTE>
    Dans les pays en voie de développement, et plus particulièrement en Afrique,
    la médecine traditionnelle ...
  <BR> <BR>
    L'objectif du système Medtrad est de capitaliser toute cette connaissance
    ancestrale dans une <A HREF="bdc.html">base de concepts</A>
  </BLOCKQUOTE>
  <DL>
    <DT>
      <H1> Liste de concepts </H1>
    <DD>
      <H2><A HREF="plante.html">Plante</A></H2>
      <H2><A HREF="maladie.html">Maladie</A></H2>
      <H2><A HREF="localité.html">Localité</A></H2>
      <H2><A HREF="consultation.html">Consultation</A></H2>
    </DD>
  </DT>
</DL>
  <HR>
  <ADRESS> K. M. BROU, INP, Yamoussoukro &lt;brou@larima.inset.ci>&gt;</ADRESS>
</BODY>
</HTML>
```

FIG. 5.5 – Code HTML de la page d'accueil de Medtrad .

Ce code HTML exécuté par le navigateur Netscape affiche la page de garde de la figure 5.6; les mots ou groupes de mots soulignés sont des ancres de liens vers d'autres pages du document. Le sommaire est constitué par la liste des concepts génériques.

Le Système MEDTRAD Médecine Traditionnelle Africaine à base de plantes médicinales Bienvenue dans le Système Medtrad !
Dans les pays en voie de développement et plus particulièrement en Afrique, la médecine traditionnelle ... L'objectif du système Medtrad est de capitaliser toute cette connaissance ancestrale dans une <u>base de concepts</u>. Liste de concepts Plante Maladie Localité Consultation
K. M. BROU, INP Yamoussoukro <brou@larima.inset.ci>

FIG. 5.6 – Page d'accueil de Medtrad.

5.3.2.2 Implantation du modèle de documents hypertextes de la BdC

Après cette présentation informelle du langage HTML (langage de création de documents hypertextes pour WWW), nous allons montrer comment est implanté le modèle de documents hypertextes de la BdC en décrivant ses objets documentaires et ses liens.

La BHD est constituée d'objets documentaires reliés entre eux par des liens.

– Les objets documentaires :

La BHD est constituée de quatre sortes d'objets documentaires (un objet documentaire est définit par le nom d'un document suivi des unités informatives qui le composent) :

- un OD pour la présentation du domaine (ODD),
- un OD pour la description des concepts élémentaires (ODCE),
- un OD pour la description des concepts complexes (ODCC),
- un OD pour la description des concepts n'appartenant pas au domaine (ODDE).

L'objet documentaire du domaine est représenté par une page d'accueil dont chaque entrée du sommaire (constituée de concepts), fait référence à un objet documentaire décrivant un concept complexe (cf. 5.6).

Les autres objets documentaires sont les composantes de la page de garde, i.e. des fichiers HTML.

Nous présentons ci-dessous la structuration hiérarchique de la BHD dans le formalise BNF :

$$\begin{aligned} \langle BHD \rangle &::= \langle Page \text{ d'accueil} \rangle \{ ", " \langle fichier ODCC \rangle \} \\ \langle fichier ODCC \rangle &::= \{ "fichier ODCE" [", " \langle fichier ODCC \rangle] \} \end{aligned}$$

Un fichier *ODCC_i* décrivant un concept complexe peut être composé d'autres fichiers ODCC et/ou d'autres fichiers ODCE décrivant les concepts élémentaires.

– Les liens :

En HTML, la structuration hiérarchique des documents est à la charge de l'auteur. Une page est reliée à ses fichiers HTML qui la composent par des liens hiérarchiques explicites. Ces liens hiérarchiques sont modélisés dans la BHD par les informations terminologiques Concepts génériques et Sous-concepts. Nous représentons ces liens hiérarchiques en HTML de la manière suivante :

```
<A HREF="ODdestination.html">Concept</A>
```

où

- ODdestination.html désigne un fichier ODCC ou un fichier ODCE.
- Concept désigne un concept générique (ou concept complexe) localisé dans un OD du domaine ou dans un ODCC.

Nous avons présenté au § 4.4.2 un algorithme qui permet de créer les liens de référence lors du processus d'hypérisation. Ces liens sont modélisés par la relation Lien décrite précédemment que nous rappelons ci-dessous :

Lien(#numlien, début, longueur, étiquette, type, numui, numui2)

Nous représentons en HTML ces liens de référence de la manière suivante :

```
<A HREF="ODdestination.html#ancree destination"> ancree source</A>
```

où

- ODdestination.html désigne un fichier ODCC ou un fichier ODCE dont le numéro est numui2,
- ancree destination est déterminée par l'attribut étiquette, et
- ancree source est déterminée par les attributs début et longueur.

– Les liens navigationnels :

La navigation dans la BHD est assurée par des liens verticaux, horizontaux et diagonaux. A chaque type de lien est associé un lien inverse.

- Les liens verticaux permettent de parcourir l'arbre de concepts dans un sens descendant ou dans un sens ascendant; ils ont comme ancree source un concept complexe

qui figure dans la liste des sous-concepts (sens descendant) ou dans la liste de concepts génériques (sens ascendant) d'un concept.

Leur implémentation est identique à celle des liens hiérarchiques.

- Les liens horizontaux permettent de passer d'un concept à son plus proche voisin ayant le même concept générique que lui.

```
<A HREF="ODdestination.html"> <IMG SRC="Suivant.gif"> </A>
```

```
<A HREF="ODdestination.html"> <IMG SRC="Précédent.gif"> </A>
```

où

Suivant.gif et Précédent.gif sont des fichiers images dont l'activation permet d'accéder à l'OD de deux concepts voisins ayant le même concept générique.

- Les liens diagonaux permettent d'accéder à n'importe quel objet documentaire; ils sont modélisés par les liens de référence. Un lien de référence inverse est représenté de la façon suivante :

```
<A HREF="ODdestination.html"> <IMG SRC="Retour.gif"> </A>
```

où

Retour.gif est un fichier image dont l'activation permet d'accéder à l'OD source du lien.

- Le lien d'ancre Sommaire existe dans toutes les unités informatives (sauf l'UI du domaine); il permet de retourner rapidement à l'UI du domaine. Ce lien est représenté comme suit :

```
<A HREF="accueil.html"> <IMG SRC="Sommaire"> </A>
```

5.3.2.3 Avantages et limites du système hypertexte

Le système hypermédia WWW utilisé pour l'exploitation hypertextuelle de la BdC présente des qualités indéniables mais aussi souffre de quelques insuffisances.

Il a comme avantages :

- sa disponibilité sur toutes les plates formes,
- son utilisation partout dans le monde,
- un formalisme bien défini et normalisé,
- un accès facile aux ressources intra ou inter-systèmes,

- un moyen idéal de vulgarisation des connaissances stockées dans la BdC.

Ses insuffisances sont dues à HTML et aux navigateurs.

En effet, HTML le langage de création de documents hypertextes n'apporte aucune aide à l'auteur :

- ni dans la conception des documents,
- ni dans la gestion des documents (création, suppression, modification.)

Quant aux navigateurs, ils n'apportent pas d'aide à l'utilisateur dans sa consultation des pages Web; il n'existe pas de graphe de parcours permettant de limiter le risque de désorientation.

C'est pour toutes ces raisons que nous avons proposé dans le cadre d'une BdC, la notion de point de vue qui permet de réduire le nombre d'informations manipulées par l'utilisateur afin de limiter sa charge cognitive qui est une source de désorientation. Nous avons également proposé le système de gestion et d'exploitation hypertexte d'une BdC (CoDB-Web) qui apporte une aide à l'utilisateur dans la conception et dans la maintenance de la BHD.

5.3.3 Le langage d'implémentation de la BdC

Pour le développement du système CoDB-Web nous avons choisi le langage JavaScript étendu d'Intrabuilder, un système de développement d'applications client/serveur fondés sur le Web [BOR 96].

Ce langage est une version enrichie de JavaScript qui est lui même une extension de Java, langage de développement d'applications distribuées pour le Web.

Ce choix a été motivé par notre souci d'avoir un environnement homogène qui facilite la communication entre les différentes composantes de la BdC : le Système hypertexte, CoDB-Web et le SGBD.

5.3.3.1 Le langage Java

Java est un langage orienté objet créé par Sun Microsystems pour le développement d'applications distribuées; sa création répondait à une nécessité de trouver un langage compréhensible par les navigateurs, dont la vocation est d'être disponible et indépendantes de toute machine et de tout système d'exploitation.

Java est un langage de programmation à part entière; sa programmation s'apparente à celle du C et du C++, mais d'utilisation plus simple. Il présente les caractéristiques suivantes :

- Héritage : pour la réutilisation du code de la super classe,
- Traitement d'exception : le code principal est séparé des traitements d'erreurs (d'où élégance et lisibilité du code) pour plus d'efficacité,
- Multiprogrammation : coexistence de plusieurs fils d'exécution (Threads) au même moment.

Java est un programme compréhensible par les navigateurs; il permet d'introduire du dynamisme aux pages Web grâce à de mini applications appelées applets Java.

Les applets Java sont intégrées dans les pages HTML au moyen d'une balise dont la syntaxe se présente comme suit :

```
<APPLET CODE="app.class" Attribut=Valeur>
  <Param Name = Type = Valeur>
  Contenu alternatif
</APPLET>
```

où

- app.class est le nom de l'application écrite en Java qui doit être exécutée par le navigateur,
- Valeur désigne la valeur de l'attribut de l'applet,
- Name désigne l'attribut de la balise Param et Type, le type de la valeur du paramètre indiquée par Valeur,
- Contenu alternatif est utilisé lorsque le navigateur ne comprend pas l'applet.

Lorsque un navigateur rencontre cette balise, il télécharge le code de l'applet et l'exécute.

Il existe beaucoup de dialectes Java : les plus connus sont JavaScript de Netscape et VBScript de Microsoft.

5.3.3.2 Le langage JavaScript

Le code source des programmes JavaScript est directement inséré dans les pages HTML sans compilation préalable (comme Java); ces programmes, délimités par la balise <SCRIPT> sont exécutés par le navigateur en même temps que le code HTML . La syntaxe de cette balise se présente de la façon suivante :

```
<SCRIPT>
    code JavaScript
</SCRIPT>
```

JavaScript se différencie de Java par les qualités suivantes :

- sa syntaxe est simple,
- il est interprété et non compilé, ce qui permet de modifier les scripts aussi facilement que les pages HTML,
- les scripts peuvent être directement intégrés n'importe où dans les pages HTML sans compilation préalable; de plus il permet le traitement des événements (par exemple le fait de cliquer sur un bouton) grâce à des modules de traitement d'événements (event handler),
- son interpréteur est intégré au navigateur, d'où la possibilité d'accéder directement aux fonctions du navigateur telles que les fenêtres, les cadres, les documents, les formulaires, les liens, les ancres, etc. afin d'exécuter des actions particulières.

Exemple de script :

```
<SCRIPT LANGUAGE="JavaScript">
    function navigate()
    {index = document.ficheit.sous-concepts.selecedIndex;
    if (index == 0)
        window.location = "fiche_concept1.htm";
    else if ...
    }
</SCRIPT>
```

```
<Input Type="button" Name="go" Value="Aller à la page" onClick="navigate();">
```

Le script ci-dessus est utilisé pour la sélection d'un concept et l'affichage de ses fiches terminologiques.

Un clic sur le bouton intitulé "Aller à la page" déclenche l'événement `onClick` qui lance l'exécution du code JavaScript `navigate()`.

Malgré ces qualités, JavaScript a des insuffisances notamment :

- sa lenteur du fait de son interprétation,
- l'absence de fonctions de communication externe avec les serveurs, et
- le manque de fonctions graphiques et multimédias puissantes.

D'où le couplage de JavaScript avec d'autres langages afin de palier à ces insuffisances (Java, CGI, ...). Le langage JavaScript Intrabuilder répond en partie à ces préoccupations.

5.3.3.3 Le langage JavaScript Intrabulder

Intrabuilder est un environnement de développement d'applications de gestion de données fondée sur le Web. Il permet la gestion et l'exploitation dynamique des données d'une base de données relationnelle depuis un navigateur Web. Le développement d'une application avec Intrabuilder s'effectue en trois étapes :

- la première étape consiste à la création des tables de la base de données,
- la seconde étape à trait à la conception des fiches Intrabuilder d'accès aux données,
- la dernière étape est l'exécution de la fiche depuis un navigateur Web.

La figure 5.7 ci-dessous présente l'environnement de développement d'Intrabuilder. La fenêtre Inspecteur permet de définir les propriétés des contrôles placés sur les fiches.

Toutes les fiches Intrabuilder sont transformées en codes JavaScript Intrabuilder afin de pouvoir être exécutées par les navigateurs.

Le langage JavaScript Intrabuilder intègre deux sortes de JavaScript : JavaScript client (ayant des fonctionnalités identiques au JavaScript décrit ci-dessus) et JavaScript serveur.

JavaScript serveur se caractérise par ses fonctions de communication avec les bases de données localisées sur les serveurs HTTP. Ces fonctions sont implémentées à l'aide de cinq

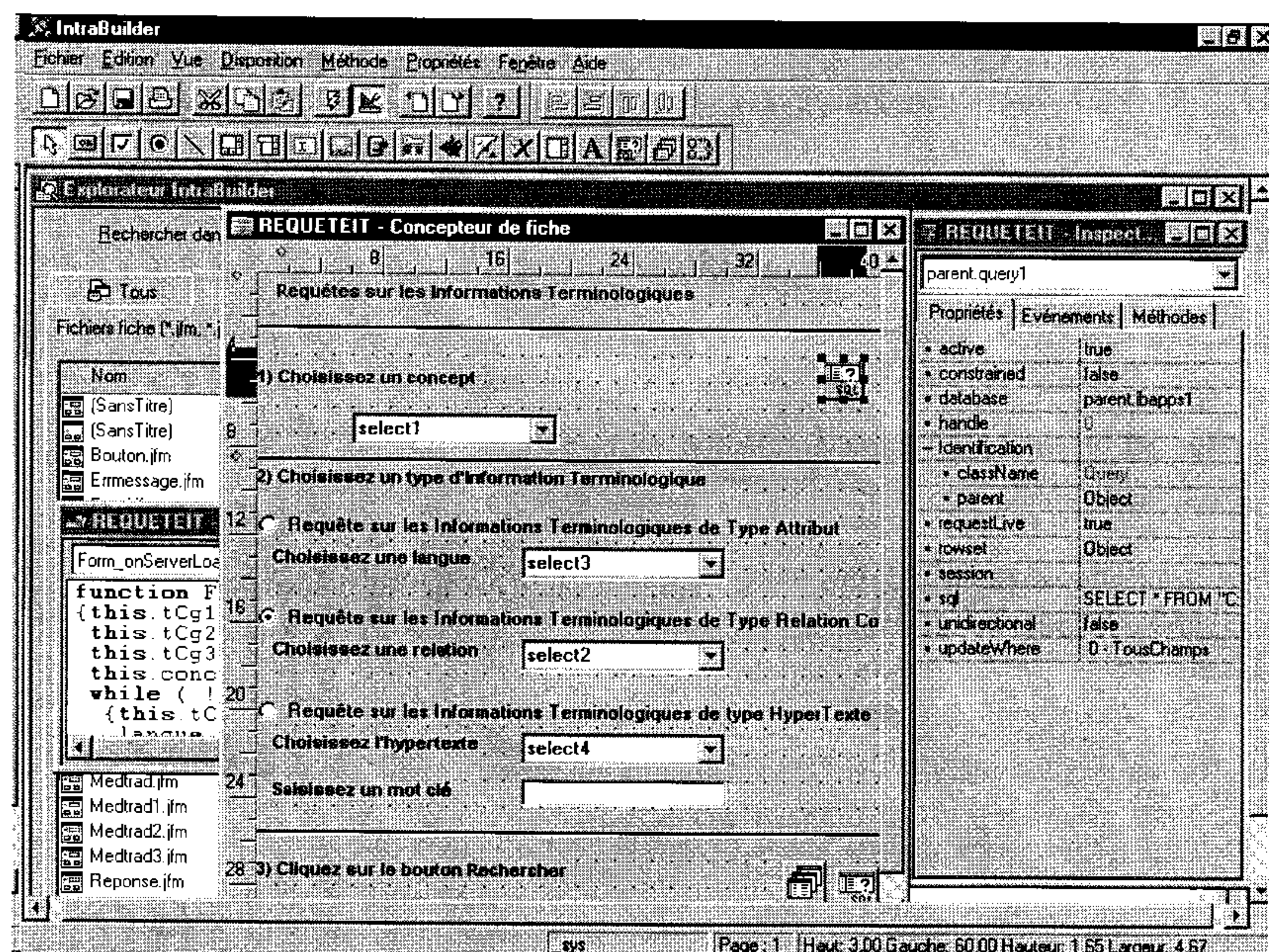


FIG. 5.7 – Environnement de développement d’Intrabuilder .

classes :

- Session : pour la gestion des sessions utilisateurs de bases de données,
- Database : pour l'accès à un serveur SQL ou à une base de donnée ODBC,
- Query : pour la définition de requêtes SQL d'accès aux données des tables,
- Rowset : pour l'utilisation des tuples renvoyés comme réponses d'une requête,
- Field : pour l'exploitation des champs d'une table.

Exemple de code JavaScript serveur :

```
function suivantButton_onServerClick()  
(1) {var nC1;  
(2)   this.tCg1 = new Array();
```

```
(3)  nC1 = parseInt(this.form.numConcept.value);
(4)  this.form.query1.rowset.first();
(5)  while(!this.form.query1.rowset.endOfSet)
(6)      {if (nC1 == this.form.query1.rowset.fields["Numconcept"].value)
(7)          {this.tCg1.add(this.form.query1.rowset.fields["Terme"].value);}
(8)      this.form.query1.rowset.next();
      } }
```

Dans cet exemple, la référence `this` désigne un objet intrabuilder (fiche, état, ...) qui a appelé la méthode.

La propriété `sql` de l'objet `query1` contient la requête SQL permettant de rechercher dans les tables de la BdC tous les concepts génériques :

```
sql = SELECT Concept.Numconcept, Concept_1.Terme
      FROM Concept, Relation_conceptuelle, Concept_1
      WHERE (Concept.Numconcept = Relation_conceptuelle.Numconcept) AND
      (Relation_conceptuelle.Numconcept2 = Concept_1.Numconcept) AND
      (Relation_conceptuelle.Nomrc = "concept générique");
```

Quant à la propriété `rowset` de l'objet `query1`, elle contient les lignes résultats de la requête; l'accès à ces lignes est effectué par les curseurs de ligne `first()` (premier enregistrement, ligne 4), et `next()` (enregistrement suivant, ligne 8). Le test de fin d'enregistrement est réalisé par la propriété `endOfSet` de l'objet `rowset` (ligne 5).

Lorsqu'on clique sur le bouton suivant, cette fonction recherche les concepts génériques du concept suivant en vue de les afficher (ligne 7 et 8). Ces concepts sont stockés dans le tableau `tCg1` (ligne 2)

5.3.4 Le système de gestion de base de données relationnelle

Le système de gestion de base de données utilisé est le SGBD relationnel ACCESS de Microsoft. ACCESS permet de créer facilement et rapidement des applications fonctionnant sous l'interface graphique de Windows.

5.3.4.1 Présentation d'ACCESS

Une BD ACCESS est composée des objets suivants :

- tables : elles stockent les données dans des champs de types classiques (numérique, texte); Ces champs peuvent aussi avoir le type Memo qui permet de stocker un texte important, ou le type Liaison OLE pour les images. Les champs de types Memo ou Liaison OLE sont utilisés pour le stockage des informations terminologiques de type hypertextes (Définition, Contexte et Image).
- requêtes : elles permettent l'accès aux données stockées dans les tables grâce au langage SQL ou par l'intermédiaire d'une interface interactive de type QBE. Les réponses à une requête sont enregistrées dans une feuille de réponses dynamiques appelée Dynaset.
- formulaires : un formulaire est une interface conviviale d'accès aux données d'une table ou d'une requête, pour une saisie ou une consultation de données.
- états : ils permettent de préparer les données en vue d'une impression ultérieure.
- macros : elles permettent d'automatiser les tâches répétitives (recherche d'enregistrements, ouverture de formulaires, ...).
- modules : un module regroupe des procédures et fonctions écrites en VB (Visual Basic, langage hôte d'ACCESS); ces sous-programmes peuvent servir aussi à automatiser les tâches répétitives.

Tous ces objets sont regroupés dans un seul fichier commun ayant l'extension .MDB (Microsoft DataBase).

Pour chacun de ces objets, ACCESS dispose d'un assistant qui aide l'utilisateur dans son travail.

5.3.4.2 ACCESS et la BdC

Comme nous l'avons souligné au § 4.4.1.1, l'utilisateur dispose de deux possibilités d'accès aux données de la BdC :

- soit directement par navigation dans la base d'hyperdocuments,
- soit par une requête sur la BD suivie d'une navigation dans la BHD.

A ces deux possibilités, il faut aussi permettre à l'utilisateur la navigation dans la base de données. Dans la BD, tous les objets documentaires sont représentés par des formulaires de consultation; la structuration hiérarchique de ces OD est assurée par des macros matérialisées par des boutons. L'activation de ces boutons permet d'ouvrir les formulaires contenant les OD composantes. Les liens de référence sont aussi visualisés sous forme de boutons (cf. annexe D).

L'inconvénient de ce mode d'utilisation de la BdC est que l'utilisateur ne peut pas accéder à la représentation hypertexte des informations terminologiques de type hypertexte.

5.4 Méthodologie d'instantiation de la BdC

Afin de mettre en place efficacement une base de concepts pour un domaine donné, nous proposons les six étapes suivantes :

1. Délimitation du domaine

Cette délimitation consiste à cerner l'ensemble des connaissances du domaine à étudier afin d'isoler les concepts clés du domaine (cf. § 6.2.1).

2. Collecte des informations

Il s'agit de collecter tous les documents de référence du domaine étudié tels que les dictionnaires, les règlements, les livres etc.

Ces documents permettront d'isoler les concepts clés du domaine et fourniront des renseignements utiles à la détermination des valeurs des informations terminologiques et des caractères définitoires (cf. § 6.2.2).

3. Modélisation du domaine sous forme d'arbres conceptuels

Il s'agit d'organiser les concepts du domaine de façon arborescente afin de faire apparaître les relations hiérarchiques ou sémantiques qui existent entre eux.

Les relations hiérarchiques permettront de trouver les concepts génériques, et les relations sémantiques, les concepts liés (cf. § 6.2.3).

4. Modélisation de l'application sous forme d'un schéma conceptuel

Pour la mise en place du schéma conceptuel, nous préconisons l'utilisation du modèle Entité/Association car son formalisme n'est pas difficile à comprendre. Dans ce formalisme, nous représentons chaque concepts par une entité, et la relation sémantique entre deux concepts traduite par une association (cf. § 6.2.4).

Le schéma conceptuel aide à la détermination des caractères définitoires des concepts par application des règles présentées au § 3.4.3 page 58.

5. Etablissement des fiches :

Une fiche représente un concept à l'aide d'un ensemble de champs textuels contenant les valeurs propres au concept. Ces champs contiennent les informations terminologiques ainsi que les caractères définitoires d'un concept.

Ces fiches facilitent la saisie des données dans les tables de la BdC (cf. § 6.2.5).

6. Utilisation du système CoDB-Web

Deux phases dans l'utilisation du système CoDB-Web : la saisie et la consultation. La saisie des informations de la BdC est réalisée par des fiches Intrabuilder de saisie. Deux possibilités sont offertes pour la saisie des informations terminologiques de type hypertexte (Définition, Contexte et Image) :

- la saisie directe de leurs contenus informatifs et de leurs structures logiques dans les tables de la BdC, ou
- la linéarisation d'un hypertexte existant qui les modélise; cette linéarisation consiste à séparer le contenu informatif de l'hypertexte de sa structure logique afin de les stocker dans les tables de la BdC. Dans ce cas, il n'y a plus d'hypérisation des informations terminologiques de type hypertexte, sauf dans les opérations de mise à jour.

La consultation des concepts de la BdC est effectuée par navigation dans la base d'hyperdocuments, grâce une interface hypertextuelle basée sur les fiches terminologiques (cf. § 6.2.6).

Cette méthodologie a été utilisée pour la conception de l'application MEDTRAD que nous présentons dans le chapitre suivant.

5.5 Conclusion

Dans ce chapitre, nous avons présenté l'architecture de fonctionnement général de la BdC, en insistant plus particulièrement sur le processus de communication entre ses différentes composantes que sont : le système hypermédia WWW, le SGBD Access et son système de gestion et d'exploitation hypertextuelle CoDB-Web.

- Le système hypermédia WWW assure l'exploitation hypertextuelle de la BdC dans une architecture de type client/serveur; son langage HTML servant de langage de création

de la base d'hyperdocuments.

- Le SGBD Access est utilisé pour la gestion des concepts.
- Le système CoDB-Web peut être considéré comme une interface particulière entre une base de données (BdC) et un hypertexte.

Il est développé en JavaScript, bon fédérateur d'applications; c'est le langage de développement d'applications interactives distribuées pour le Web.

Nous avons également présenté une méthodologie d'instantiation de la BdC afin d'être efficace dans la mise en place d'une base de concepts d'un domaine donné.

Chapitre 6

Application : le système MEDTRAD

6.1 Introduction

Dans les pays en voie de développement, et plus particulièrement en Afrique, la médecine traditionnelle devient de plus en plus une alternative très intéressante à la médecine dite moderne. Ce regain d'intérêt de la médecine traditionnelle est dû à son efficacité et à son coût de prise en charge moins élevé.

En effet, la majeure partie de la population africaine est rurale et elle dispose de très peu de moyens financiers pour se faire soigner dans les hopitaux modernes. Elle préfère donc se tourner vers les guérisseurs ou tradipraticiens dont le savoir et le savoir-faire est basée sur l'expérience et la connaissance ancestrale des plantes médicinales.

Cette connaissance est transmise de façon orale de génération en génération, mais malheureusement elle tend à disparaître avec le décès de ces tradipraticiens. A ce propos un sage disait, «En Afrique, quand un vieillard meurt, c'est une bibliothèque qui brûle».

L'objectif du système MEDTRAD (MEDecine TRADitionnelle) est de capitaliser toute cette connaissance à travers une BdC afin de péreniser cette théapeutique ancestrale.

Le problème de la médecine traditionnelle a été abordé dans le cadre du système SEI-BOGA [FRA 92]. Ce système expert permet de reconnaître de façon icônique les plantes et leurs vertus curatives. Il montre également comment utiliser ces plantes pour soigner une maladie suite à un diagnostic interactif d'un patient.

Ce système présente des insuffisances, en effet, les connaissances sur les maladies sont

représentées par des règles. Cette approche induit des limites, car l'utilisateur ne peut interroger la BD pour établir les relations entre Maladie, Plante et Localité. Ce système est basé sur une exploitation graphique, nécessitant parfois une certaine compétence de l'utilisateur (découpage de l'écran en plusieurs zones, choix de fonctions et d'opérateurs de comparaison, etc.).

L'application MEDTRAD [BRO 96] que nous proposons est basée sur l'exploitation d'une base de concepts sur les plantes médicinales africaines. Cette approche nous semble intéressante pour plusieurs raisons :

- elle permet de capitaliser dans une base de concepts les connaissances en médecine traditionnelle par la description des concepts utilisés dans ce domaine (Plante, Maladie, Localité et Remède),
- elle donne la possibilité à l'utilisateur d'interroger la base de concepts grâce à une interface hypertextuelle, bien adaptée à une mémorisation à long terme et à une compréhension profonde (assimilation) [FOR 93],
- elle fournit aux tradipraticiens une aide dans le diagnostic des maladies ainsi que dans la prescription des traitements thérapeutiques.

Notre but n'est pas de contester la thérapeutique moderne, mais plutôt de mettre à la portée de l'utilisateur, des plantes médicinales d'usage courant, éprouvées par de nombreux spécialistes [MAY 90] [SIJ 93] pour soulager un malade ou pour renforcer un traitement médical.

6.2 Présentation de la BdC de MEDTRAD

Pour l'instantiation de la BdC avec les concepts utilisés en médecine traditionnelle, nous suivrons la méthodologie de conception d'une BdC que nous avons proposée dans le chapitre précédent.

6.2.1 Délimitation du domaine

Le cadre de notre étude est celui de la médecine traditionnelle africaine à base de plantes médicinales.

Dans cette médecine, les tradipraticiens ont recours à des pratiques ésotériques et à la phytothérapie (traitement des maladies par des plantes) pour le diagnostic et la thérapeutique des maladies. Nous avons limité notre domaine à la phytothérapie car elle présente des aspects rationnels et des résultats convaincants éprouvés par de nombreux chercheurs africains, notamment sur la composition chimique, les propriétés thérapeutiques, le dosage et les contre-indications des plantes. Les plantes que nous utilisons sont les plantes médicinales du Sénégal.

6.2.2 Collecte des informations

Nous avons utilisé comme sources d'informations pour la réalisation de Medtrad :

- l'ouvrage intitulé "Plantes médicinales du sahel" [MAY 90]. Cet ouvrage décrit de façon complète (étude ethnobotanique, propriétés thérapeutiques, composition chimique, emploi et posologie) une soixantaine de variétés de plantes médicinales du Sénégal dont l'efficacité et la non toxicité ont été éprouvées en milieu africain, autant par les tradipraticiens que par les chercheurs.
- Nous nous sommes également inspirés de l'article intitulé "Interface visuelle pour l'aide au diagnostic médical en médecine traditionnelle" [FRA 92].
- Pour la descriptions des symptômes des maladies et leur traitement, nous avons utilisé comme base de travail, le rapport de synthèse réalisé par [DIE 95]. Ce rapport présente dans le formalisme des règles de production le savoir (règles de diagnostic) et le savoir-faire (règles thérapeutiques) des tradipraticiens du Sénégal (cf. annexe C).

6.2.3 Modélisation du domaine sous forme d'arbres conceptuels

Nous faisons la distinction entre les concepts clés du domaine (Plante, Maladie, Localité, Remède) et les autres concepts qui sont soit leurs instances, soit leurs caractères définitoires de type complexe (CDTC ou CDTR).

- **Plante** : désigne toutes les plantes médicinales; ses CD complexes sont : Racine, Tige, Feuille, Fleur, Fruit, Limbe, Composition chimique et Propriété thérapeutique.
- **Maladie** : désigne toutes les maladies; son CD complexe est : Symptômes.
- **Localité** : désigne toutes les localités où poussent les plantes; son CD complexe est : Domaine de végétation.

- Remède : désigne tous les remèdes à base de plantes médicinales.

A chaque concept générique correspond un arbre conceptuel du domaine (Fig. 6.2).

Les relations de composition permettent de déterminer les caractères définitoires complexes des concepts.

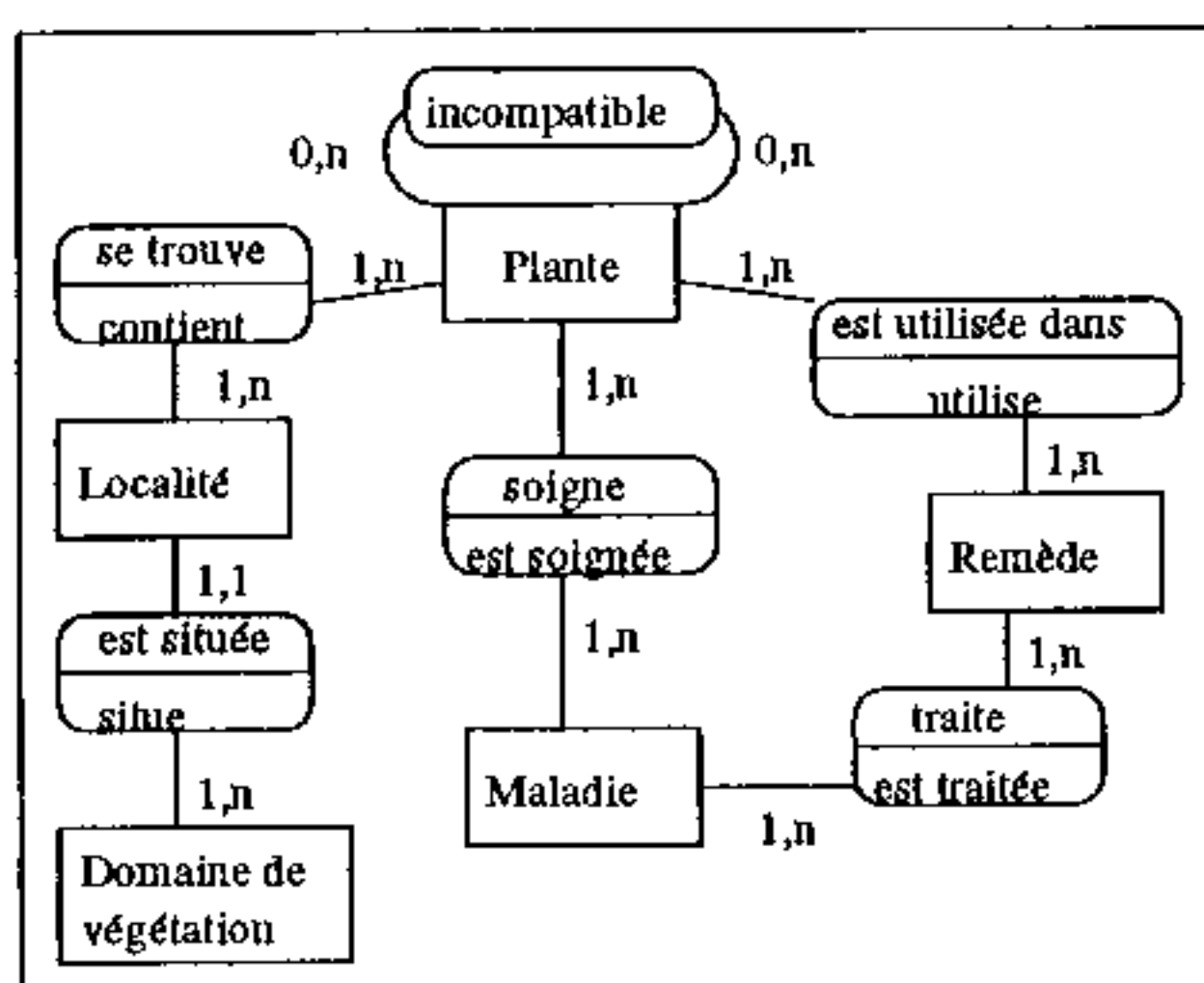


FIG. 6.1 – Relations sémantiques entre les concepts du domaine.

La figure 6.1 présente à l'aide du formalisme Entité/Association les relations sémantiques qui existent entre les arbres conceptuels du domaine :

- se trouve : une plante se trouve dans une ou plusieurs localité(s).
- contient : une localité contient une ou plusieurs plante(s).
- est située : une localité est située dans un seul domaine de végétation.
- est situe : un domaine de végétation permet de situer les localités .
- Est soignée : une maladie peut être soignée par des plantes.
- Soigne : une plante soigne des maladies.
- Incompatible: une plante peut être incompatible avec une ou plusieurs plante(s).
- Est traitée : une maladie peut être traitée par des remèdes.
- Traite : un remède traite des maladies.
- Utilisée dans : une plante est utilisée dans un remède.
- Utilise : un remède utilise des plantes dans sa préparation.

Par application de la règle 1, nous déduisons de ce schéma que les concepts Localité, Maladie, et Remède sont des concepts liés au concept Plante et vice versa. De même, les concepts Maladie et Remède sont des concepts liés.

La figure 6.2 présente leurs arbres de domaine.

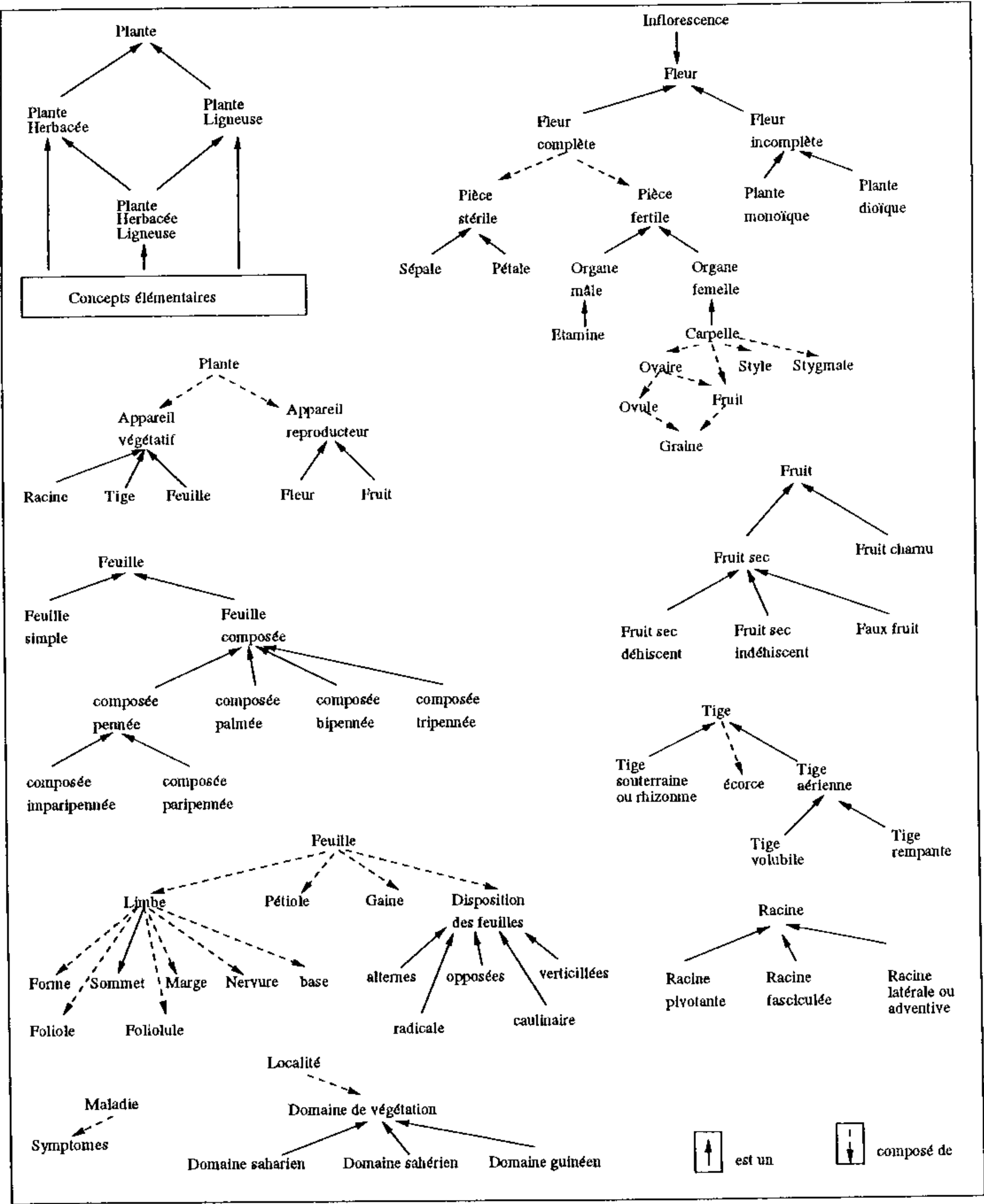


FIG. 6.2 - Arbres conceptuels du domaine.

6.2.4 Modélisation de l'application sous forme d'un schéma conceptuel

Le schéma conceptuel aide à la détermination des caractères définitoires des concepts par application des règles énumérées au § 3.4.3.

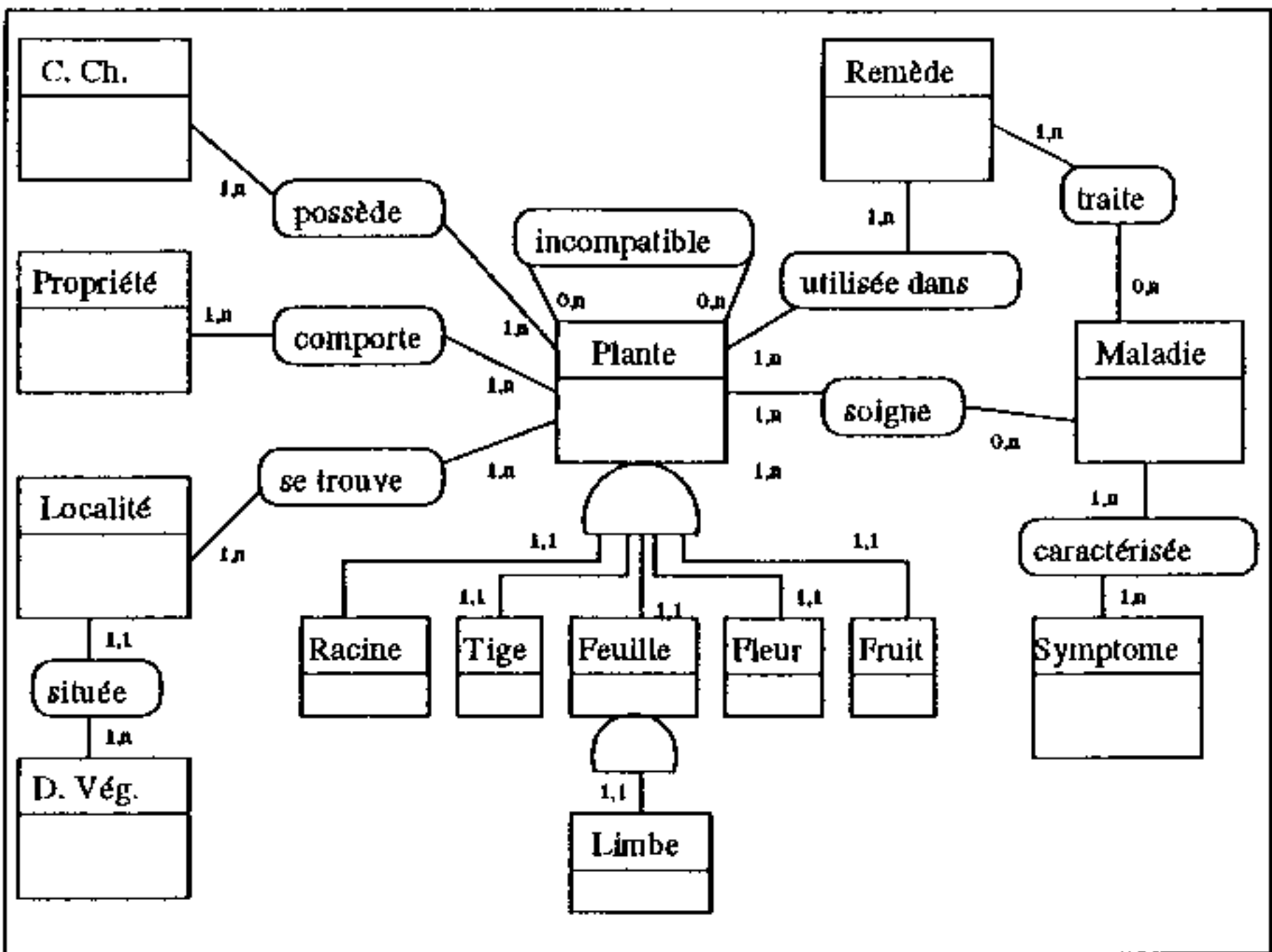


FIG. 6.3 – Schéma conceptuel de l'application.

De ce schéma, nous déduisons les caractères définitoires complexes suivants, par application des règles que nous avons présentées au § 3.4.3 page 58 :

- Les entités Racine, Tige, Feuille, Fleur et Fruit sont des caractères définitoires de type concept (CDTC) du concept Plante, par application de la règle 2 .
- Les entités Composition chimiques et Propriété sont des caractères définitoires de type relation (CDTR) du concept Plante, par application de la règle 3.
- L'entité Limbe est CDTR du concept Feuille, par application de la règle 3.
- L'entité Symptôme est CDTR du concept Maladie, par application de la règle 3.
- L'entité Domaine de végétation est CDTC du concept Localité, par application de la règle 2.

6.2.5 Etablissement des fiches

Une fiche représente un concept à l'aide d'un ensemble de champs textuels contenant les valeurs propres au concept. Ces champs contiennent les informations terminologiques ainsi que les caractères définitoires d'un concept.

Il existe deux sortes de fiches : la fiche information terminologique, et la fiche caractère définitoire (cf. § 4.4.1).

Il s'agit de déterminer dans un premier temps le type des informations du domaine qui vont permettre d'alimenter les champs de ces fiches, et dans un deuxième temps d'instancier ces fiches pour chaque concept.

Nous présentons ci-dessous (Fig. 6.4) les informations retenues pour la fiche information terminologique, ainsi que la liste détaillée de tous les caractères définitoires de chaque concept du domaine (6.5).

Concept	Plante	Maladie	Localité	Remède	Domaine de végétation
Terme	nom scientifique	nom scientifique	nom	nom	nom
Langue	latin	français	français	français	français
Synonyme	en latin, français, langues locales	en français, langues locales	en français, langues locales	en français, langues locales	en français, langues locales
définition	caractères botaniques remarquable	description	situation géographique	description	description
Contexte	étude ethnobotanique	causes de la maladie	description du milieu	conditions d'utilisation	type de végétation
Concept générique	Domaine	Domaine	Domaine	Domaine	Domaine
Sous-concept	Plante ligneuse Plante herbacée Plante herbacée-ligneuse				Domaine saharien Domaine sahérien Domaine guinéen
Concept lié	Maladie Localité Remède	Plante Remède	Plante	Plante Maladie	
Image	photo de la plante				carte du domaine
Homonyme					

FIG. 6.4 – Description des informations terminologiques.

Concepts	CD simples	CD de type concept	CD de type relation
Plante	espèce genre variété inflorescence hauteur minimale hauteur maximale	Racine Tige Feuille Fleur Fruit	Composition chimique Propriété
Feuille	disposition gaine pétiole couleur longueur minimale longueur maximale largeur minimale largeur maximale		Limbe
Limbe	forme base nombre de nervures nombre de paires de pinules nombre de paires de folioles nombre de paires de foliolules		
Tige	forme couleur présentation longueur minimale longueur maximale		
Fleur	couleur longueur minimale longueur maximale largeur minimale largeur maximale		
Fruit	couleur forme nombre de graines longueur minimale longueur maximale largeur minimale largeur maximale		
Maladie	localisation		Symptôme
Localité	région	Domaine de végétation	
Domaine de végétation	situation géographique pluviométrie nature du milieu		
Racine			
Remède	recette posologie avertissement		

FIG. 6.5 – Liste des caractères définitoires.

6.2.6 Utilisation de MEDTRAD

Après avoir rempli les fiches terminologiques (avec les informations terminologiques et les caractères définitoires qui ont été collectés à l'étape précédente) et lancé le système CoDB-Web, la base d'hyperdocuments est générée; deux modes de consultation sont proposés à l'utilisateur : la navigation grâce à un navigateur et la recherche initiant la navigation.

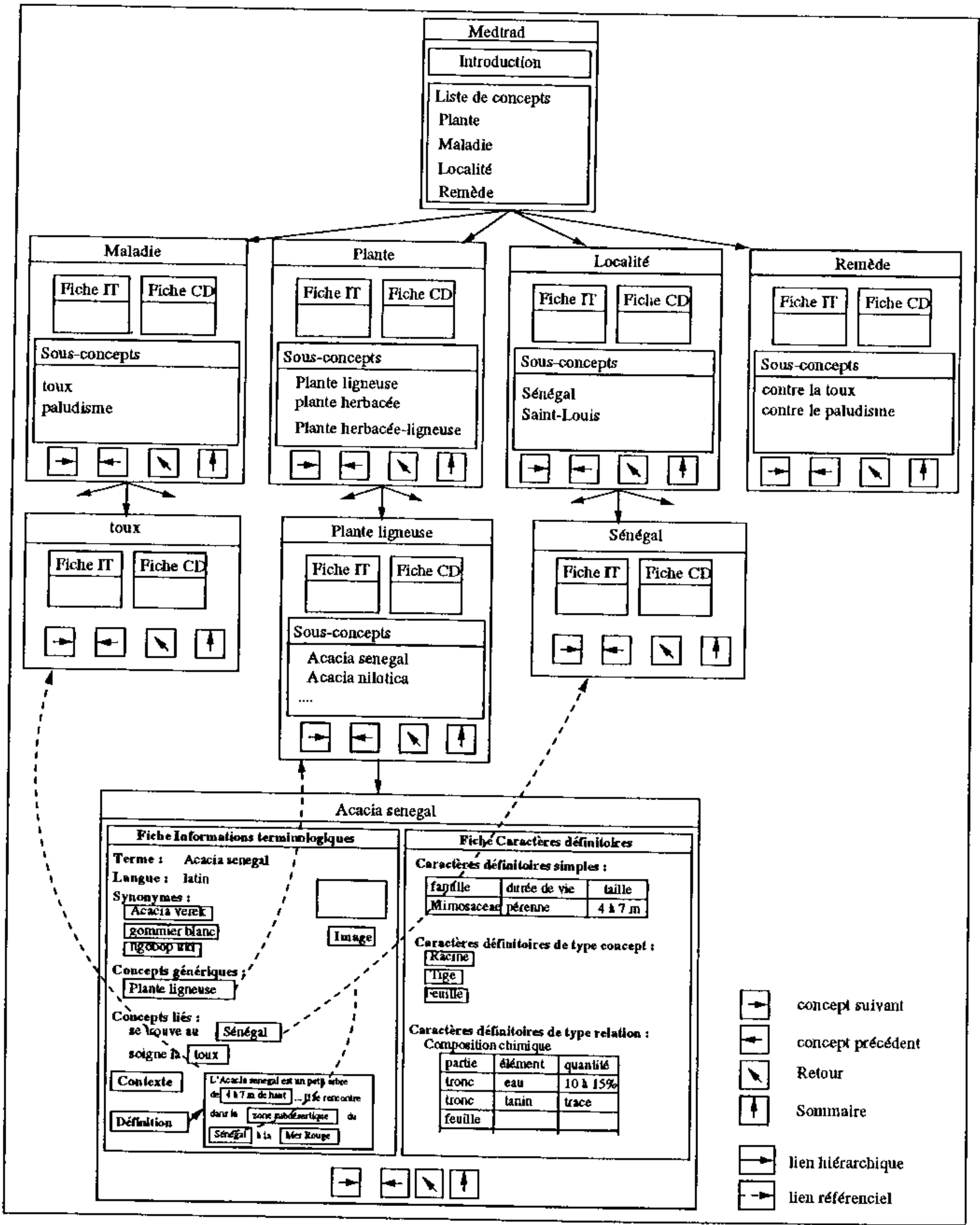


FIG. 6.6 – Base d'hyperdocuments de Medtrad.

La figure 6.6 schématise la structuration de la base d'hyperdocuments de Medtrad. Les traits pleins symbolisent les liens hiérarchiques, et les traits en pointillé, les liens de référence.

La page d'accueil est composée d'une introduction qui décrit l'application et d'un sommaire constitué par la liste des concepts clés domaine ; ces concepts sont des ancrs de liens permettant d'accéder à leurs objets documentaires. Chaque concept est représenté par deux fiches terminologiques comme celle du concept *Acacia senegal*. La descente dans l'arbre conceptuel se fait grâce à l'information terminologique *Sous-concept* (liens hiérarchique). Les liens référentiels assurent la liaison entre les différents arbres conceptuels.

Nous présentons dans les § suivants, les images écran de la maquette de l'application MEDTRAD.

6.2.7 La page d'accueil de Medtrad

La page d'accueil de Medtrad comporte une introduction qui pose la problématique de la médecine traditionnelle en Afrique, et d'un sommaire constitué par les deux modes d'accès à Medtrad.

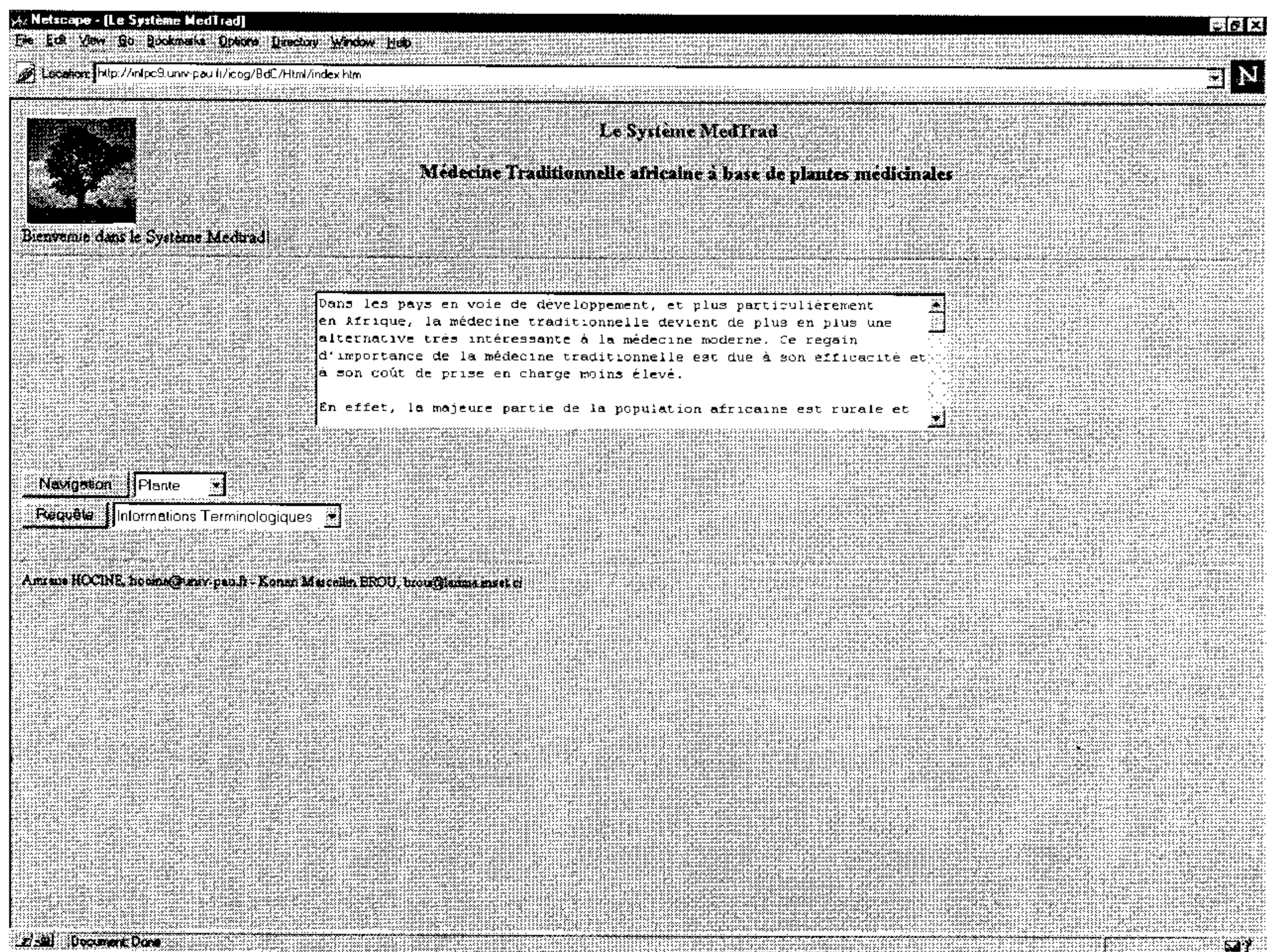


FIG. 6.7 – le menu général de Medtrad.

La naviagtion est matérialisée par le bouton Concepts génériques associé à une liste déroulante contenant la liste des concepts clés du domaine (Fig. 6.7); l'activation de ce bouton permet à l'utilisateur de naviguer dans la BdC afin de prendre connaissance de la sémantique des concepts grâce à leurs fiches terminologiques.

La recherche est symbolisée par le bouton Requête associé à une liste déroulante contenant les deux types d'information manipulés dans la BdC : informations terminologiques et caractères définitoires. Chaque type d'information est l'ancre d'une interface de requête. La réponse à une requête est une liste de concepts satisfaisant aux critères de recherche; la sélection d'un de ces concepts entraine la navigation dans Medtrad.

La liste déroulante "Concepts génériques" contient les concepts les plus abstraits du domaine: Plante, Maladie, Localité et Remède. Après avoir choisi un concept (ici Plante) l'utilisateur clique sur le bouton "Concepts générique" afin d'accéder aux fiches terminologiques du concept sélectionné (Fig. 6.8).

6.2.8 Accès à Medtrad par navigation

6.2.8.1 Fiches terminologiques du concept plante

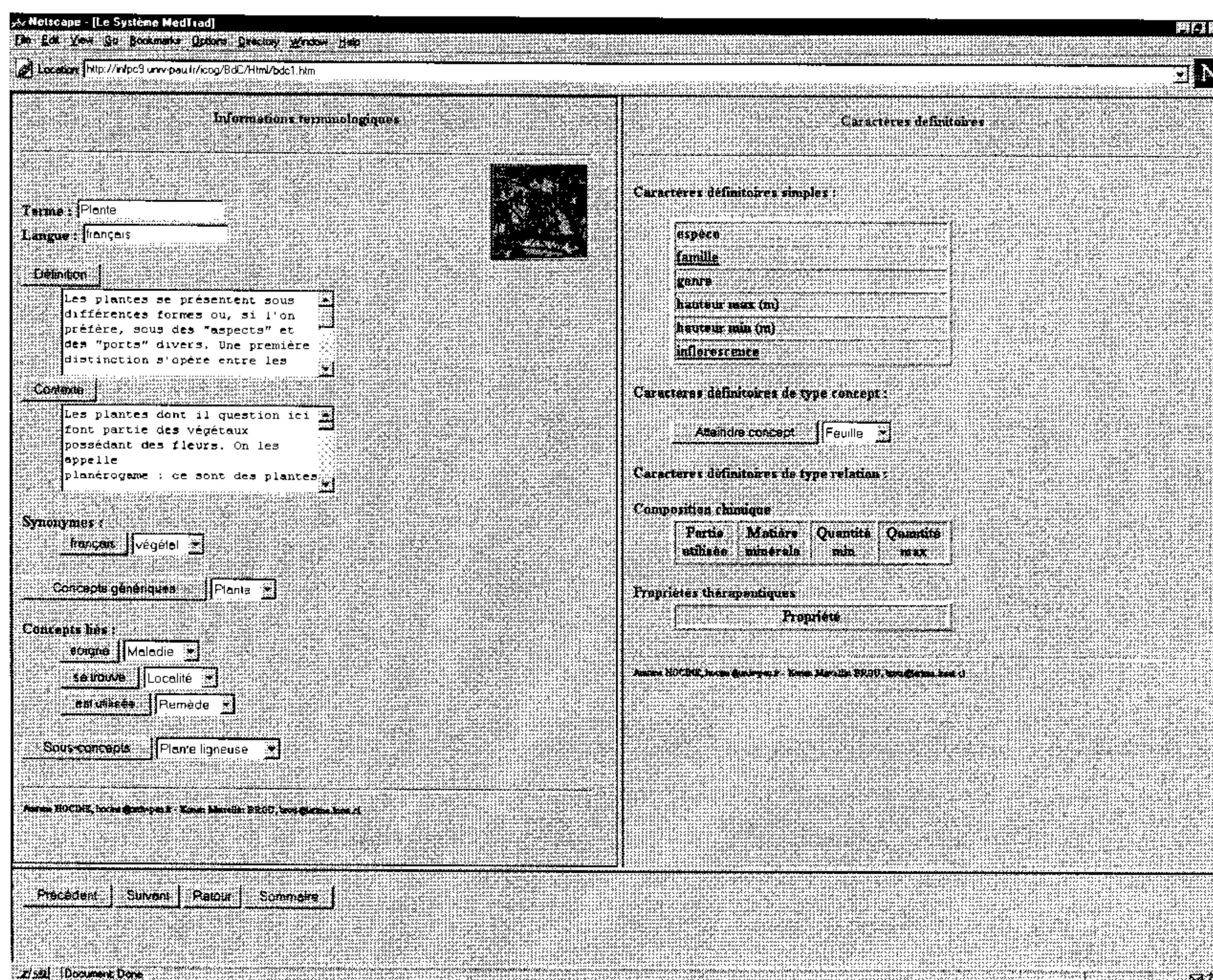


FIG. 6.8 – Fiches terminologiques du concept Plante.

La figure 6.8 présente l'interface de consultation de la BdC; elle comporte trois zones: les deux fiches terminologiques et la partie des boutons de navigation; elle a été conçue à l'aide de frames HTML.

La zone description n'est pas physiquement représentée, elle est réalisée à l'aide d'une fenêtre d'incrutation qui apparaît lors de l'activation d'un lien de description (cf. figure 6.13

page 141).

Les champs des fiches Caractères définitoires des concepts les plus abstrait tels que le concept plante sont vides.

6.2.8.2 Descente dans l'arbre de concepts

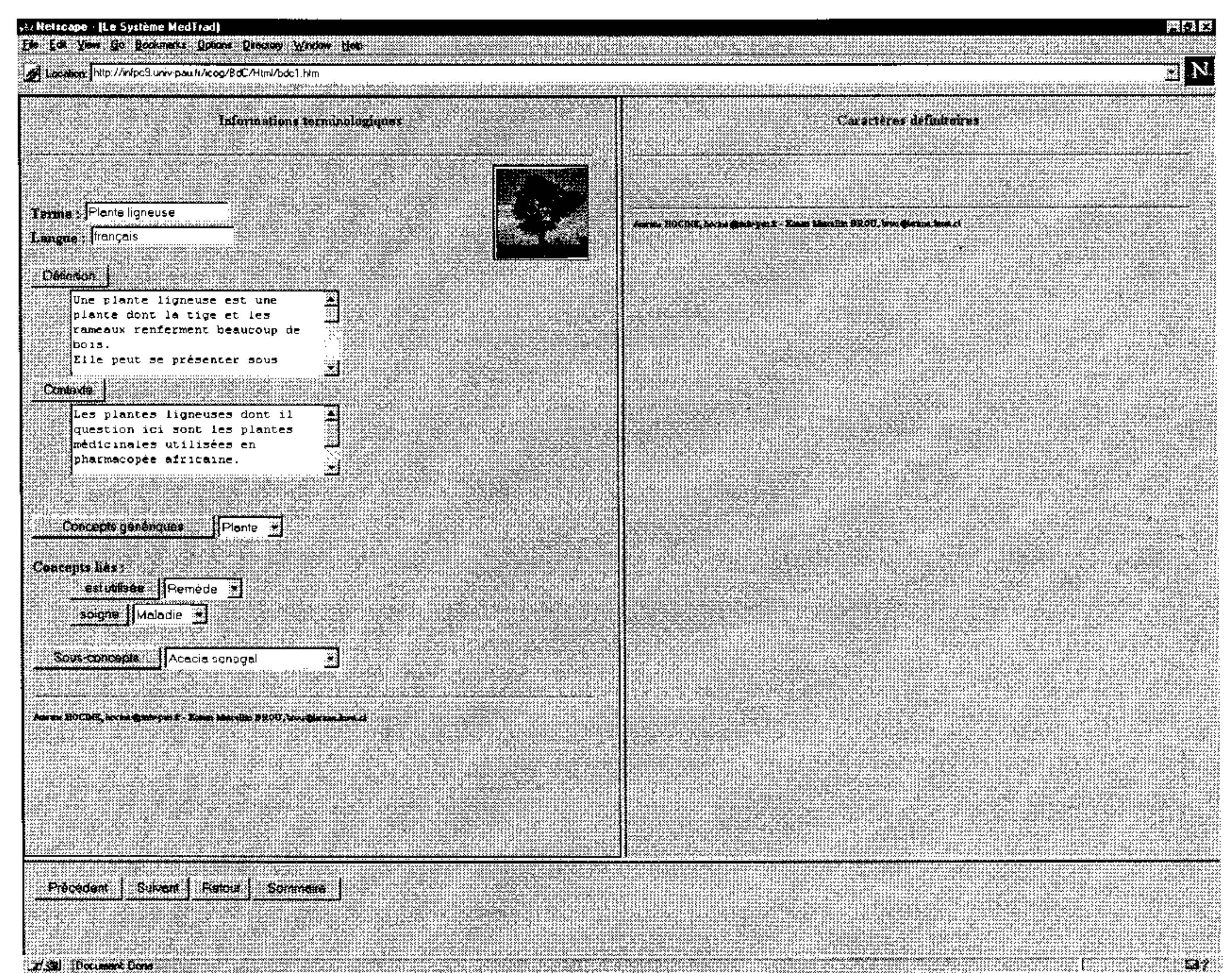
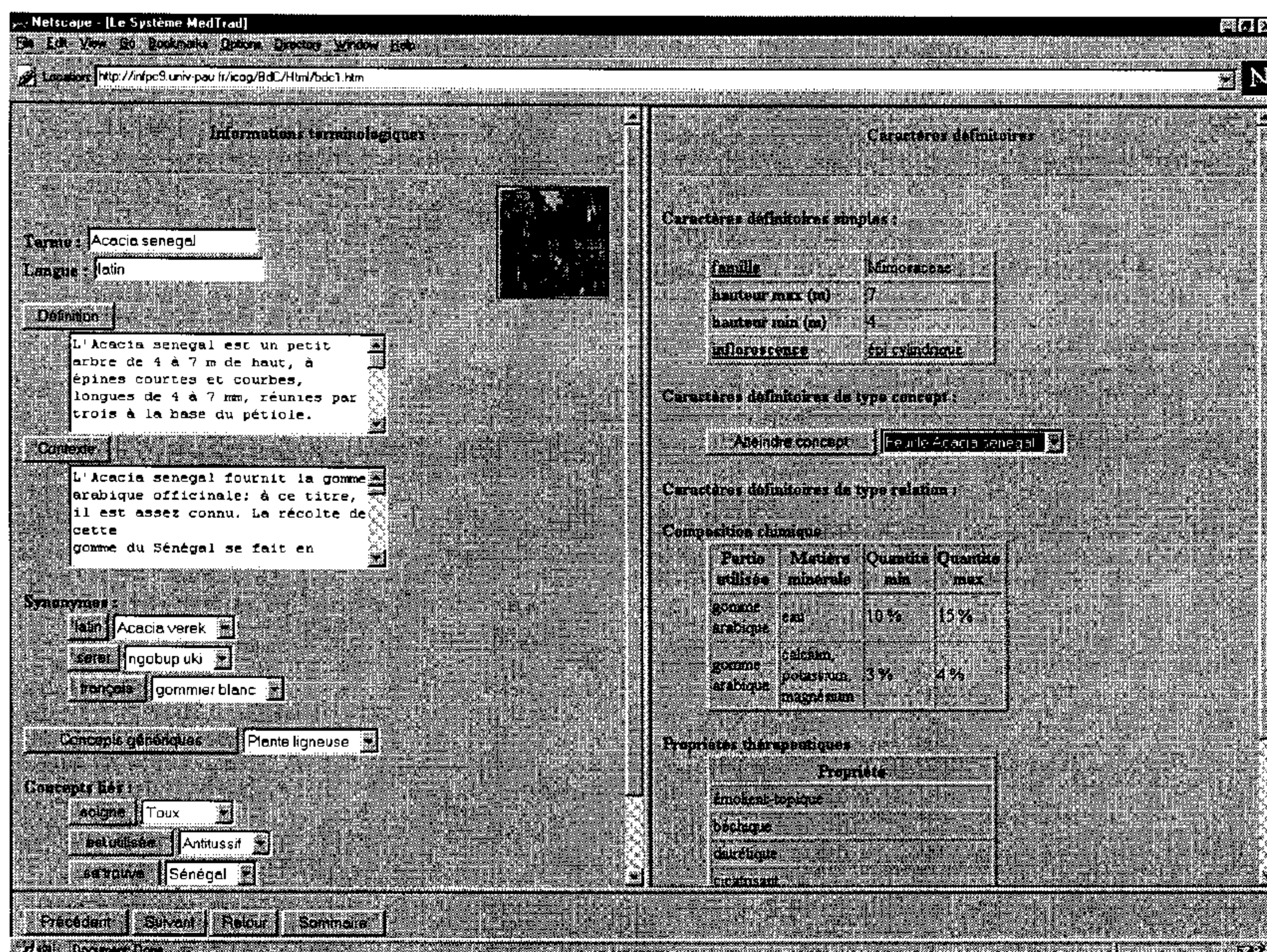


FIG. 6.9 – Fiches terminologiques du concept Plante ligneuse.

Les concepts complexes possèdent des sous-concepts accessibles dans la liste déroulante "Sous-concept". La sélection du concept Plante ligneuse dans la liste des sous-concept de Plante donne la figure 6.9.

Tous les sous-concepts du concept Plante ligneuse sont tous des concepts élémentaires; ces concepts ont des caractères définitoires bien définis.

FIG. 6.10 – Fiches terminologiques du concept *Acacia senegal*.

6.2.8.3 Fiches terminologiques d'un concept élémentaire

Le concept *Acacia senegal* est un concept élémentaire; le champ Sous-concept de sa fiche Informations terminologiques est vide car il ne possède pas de sous-concepts. Par contre les champs de ses caractères définitoires ne sont pas vides.

6.2.8.4 Activation d'un lien référençant un hypertexte

Le déclenchement d'un lien hypertexte lance l'exécution d'un programme JavaScript qui va récupérer dans les tables de la BdC, le texte et les liens des informations terminologiques de type hypertexte afin de générer dynamiquement l'hypertexte correspondant (hypérisation).

L'activation du bouton "Définition" affiche l'hypertexte associé à la définition du concept *Acacia senegal* à la même place que sa fiche Caractères définitoires (Fig. 6.11);

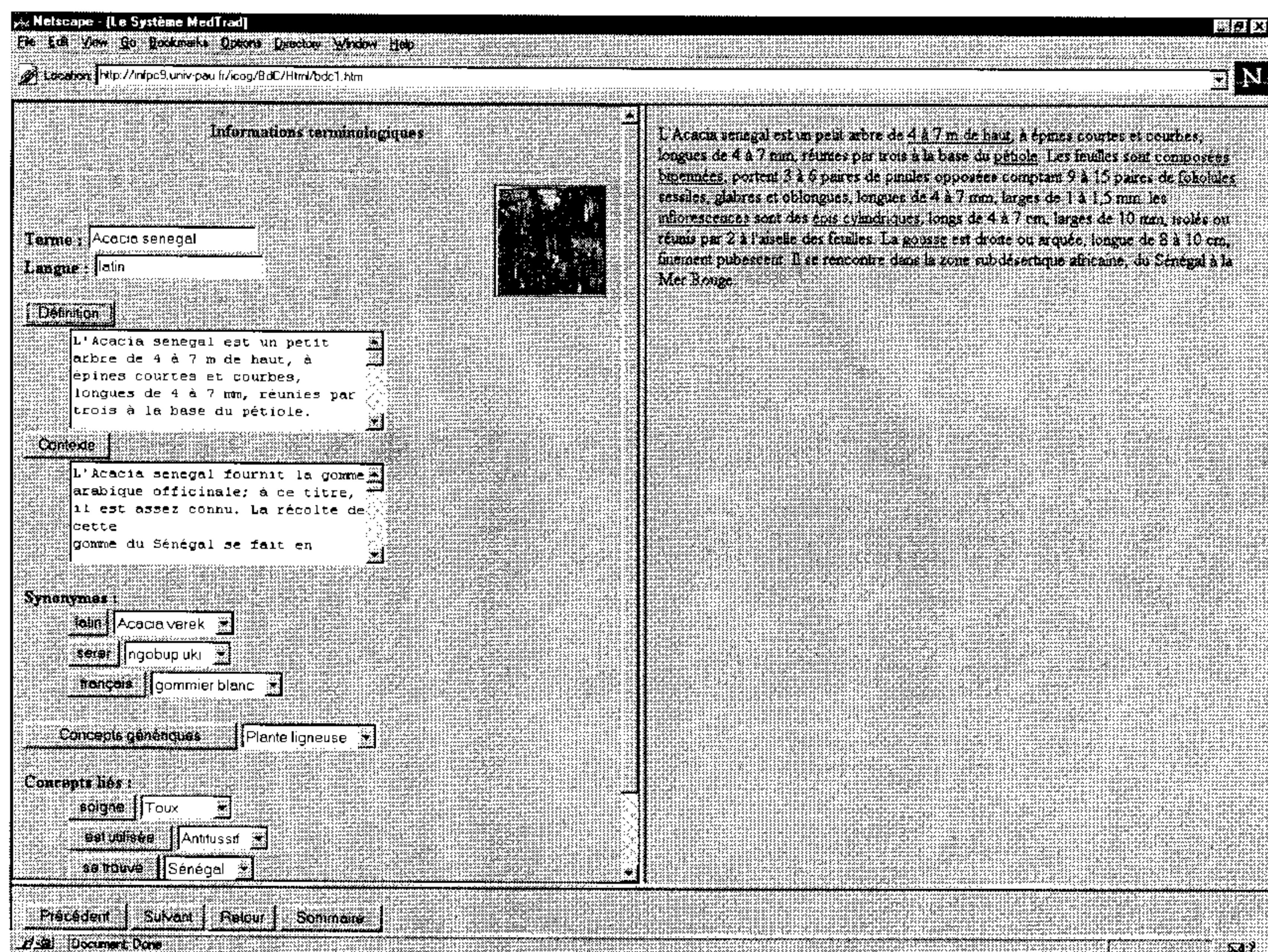


FIG. 6.11 – Activation du lien hypertextuel Définition.

L'utilisateur peut ensuite naviguer dans ce texte par activation des liens qu'il renferme. Si un lien fait référence à un concept, ses fiches terminologiques sont affichées; par contre, si c'est un lien de description (exemple l'ancre épis cylindriques) il y a affichage d'une fenêtre d'incrustation (Fig. 6.13) contenant un texte bref d'explication du concept. Le processus est identique pour les informations terminologiques Contexte et Image.

Le déclenchement d'un lien hypertexte image affiche l'image agrandie du concept. Cette image contient des zones sensibles dont l'activation permet d'acéder à d'autres objets documentaires de la BdC.

La figure figure 6.12 présente l'image du concept Feuille; elle permet aussi de découvrir les concepts Limbe, Nervure, Pétiole et Renflement.

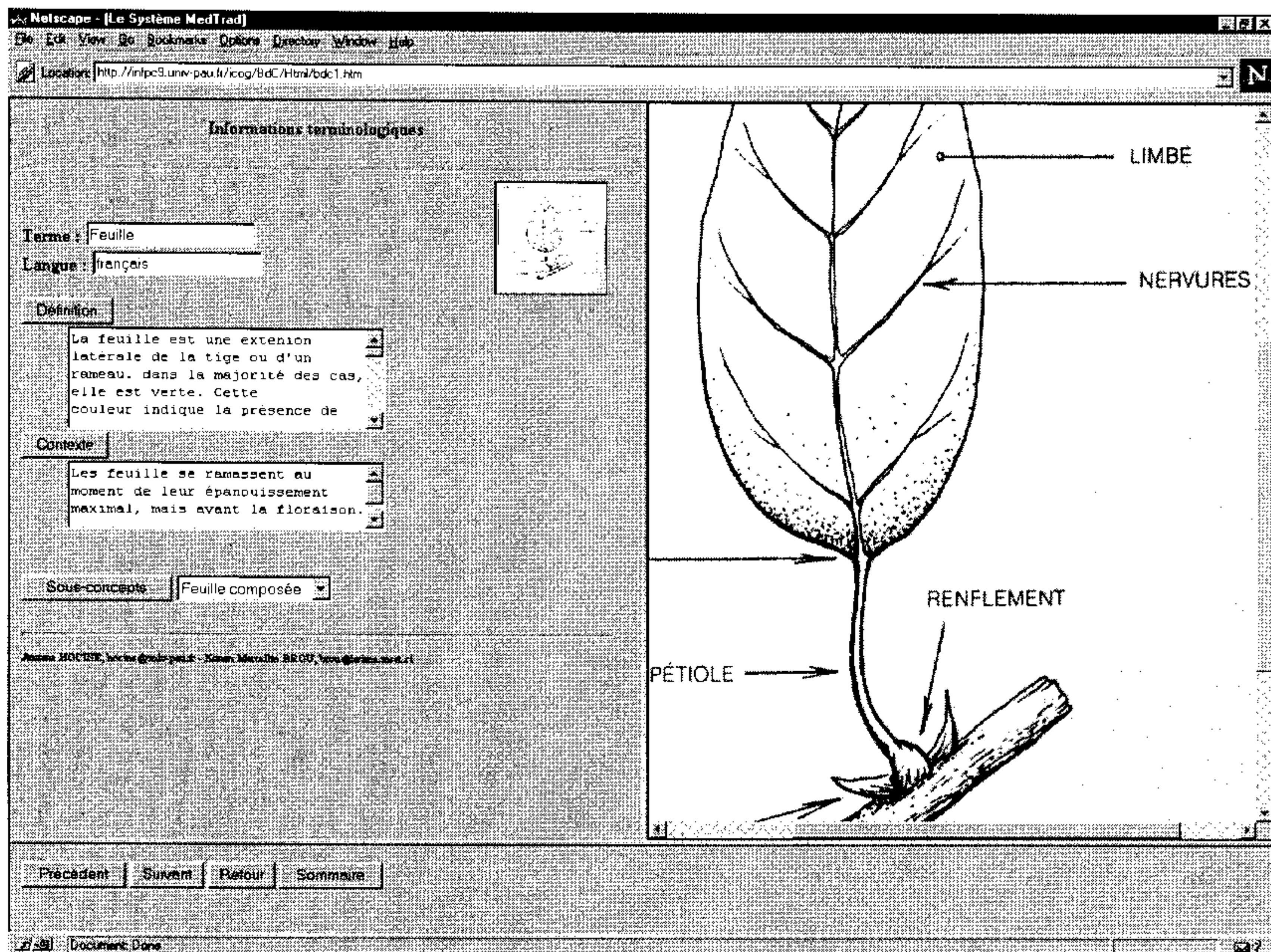


FIG. 6.12 – Activation du lien hypertextuel Image.

6.2.8.5 Activation d'un lien de description

Une fenêtre de description est identique à la notion de note en hypertexte; elle peut contenir un texte et/ou une image qui ne contient pas de liens.

La figure 6.13 présente une de fenêtr d'incrustation qui a été généré à l'activation du lien de description d'ancre épis cylindriques.

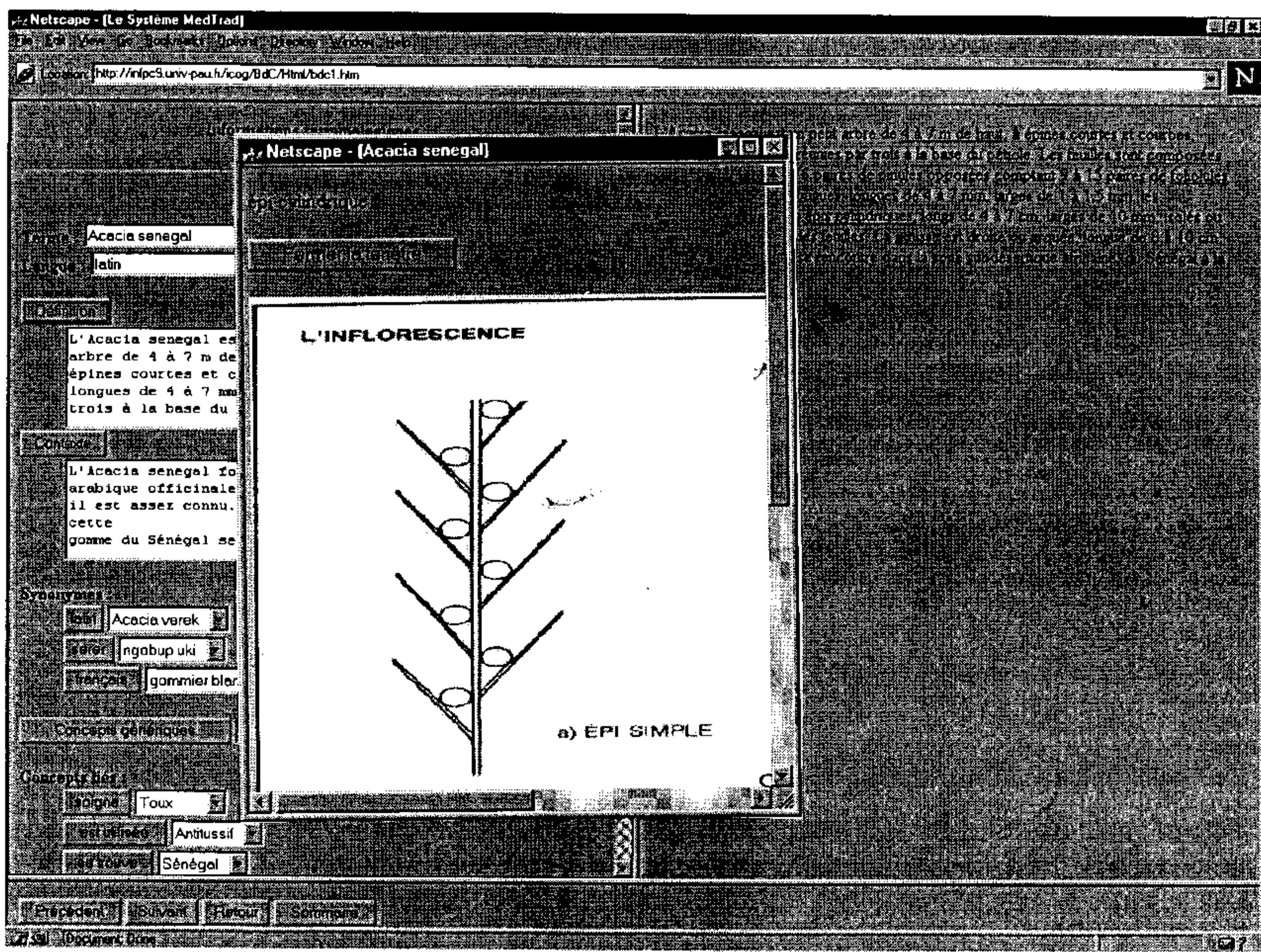


FIG. 6.13 – Activation d'un lien de description.

6.2.8.6 Changement d'arbre conceptuel

L'accès à un autre arbre conceptuel que celui du concept courant se fait par activation d'un lien de référence dont l'ancre peut être soit un concept lié, soit un concept qui est CDTC, soit un CDS, soit un attribut d'une relation qui est un CDTR du concept courant.

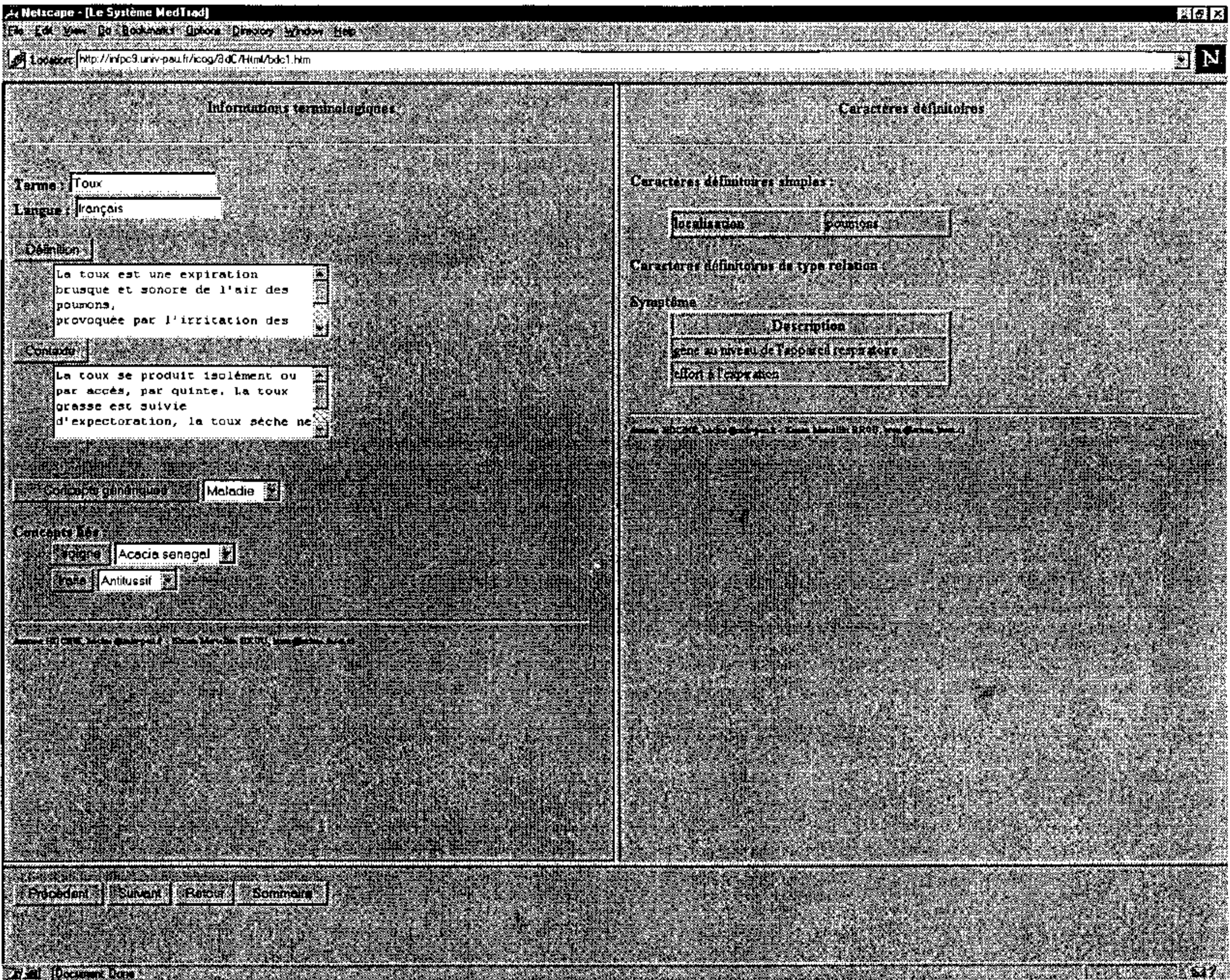


FIG. 6.14 – *Changement d’arbre conceptuel.*

La figure 6.14 présente les fiches terminologiques du concept toux qui est un concept lié au concept Acacia senegal.

6.2.9 Accès à Medtrad par une recherche initiant la navigation

Après avoir choisi le type d’information qu’il souhaite manipuler et cliqué sur le bouton Requête, l’utilisateur accède à une interface de requête lui permettant d’interroger la BdC de Medtrad. Cette interface est constituée de deux fiches : une fiche pour des requêtes portant sur les informations terminologiques, et une autre pour celles portant sur les caractères définitoires (figure 6.15).

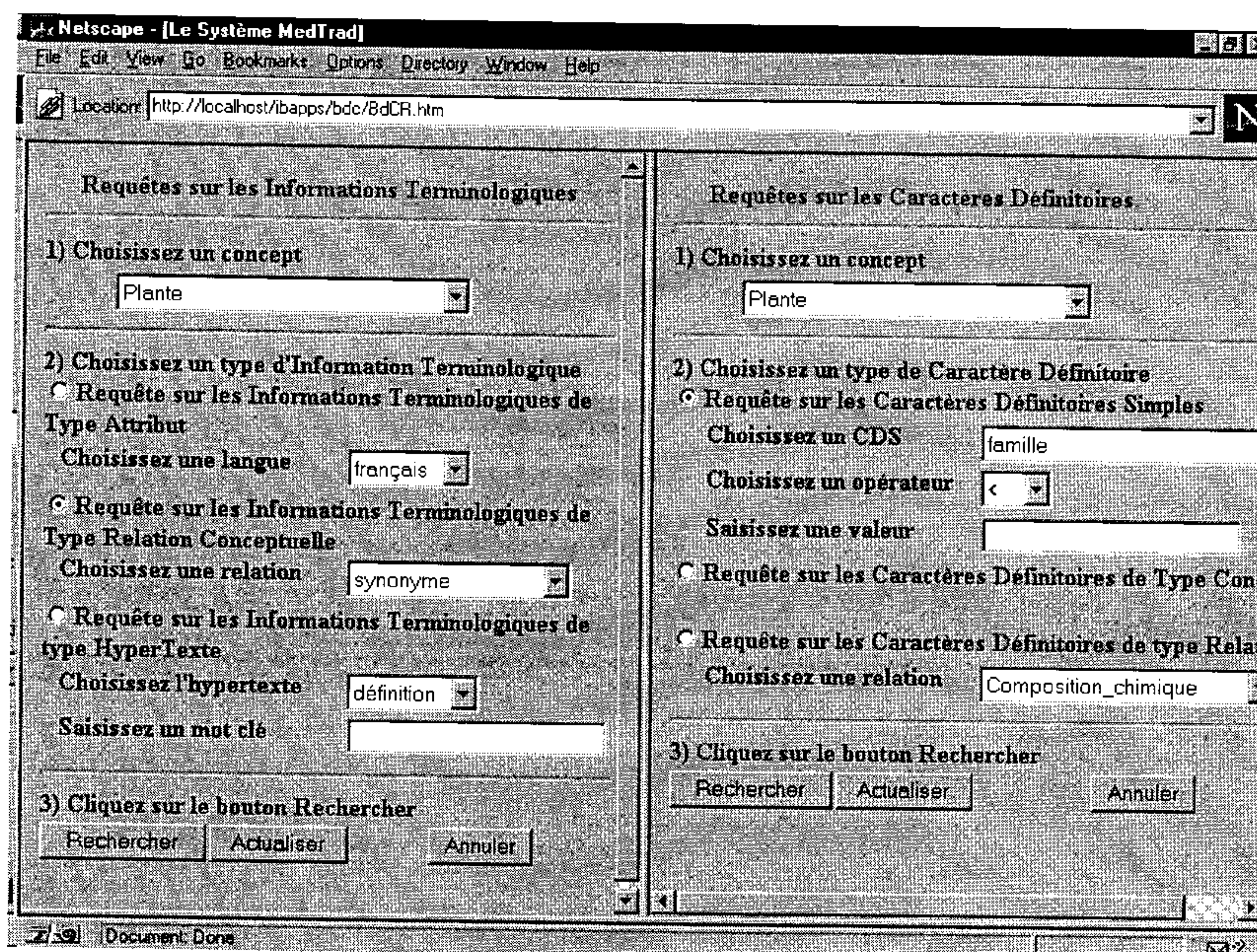


FIG. 6.15 – Interface de requête.

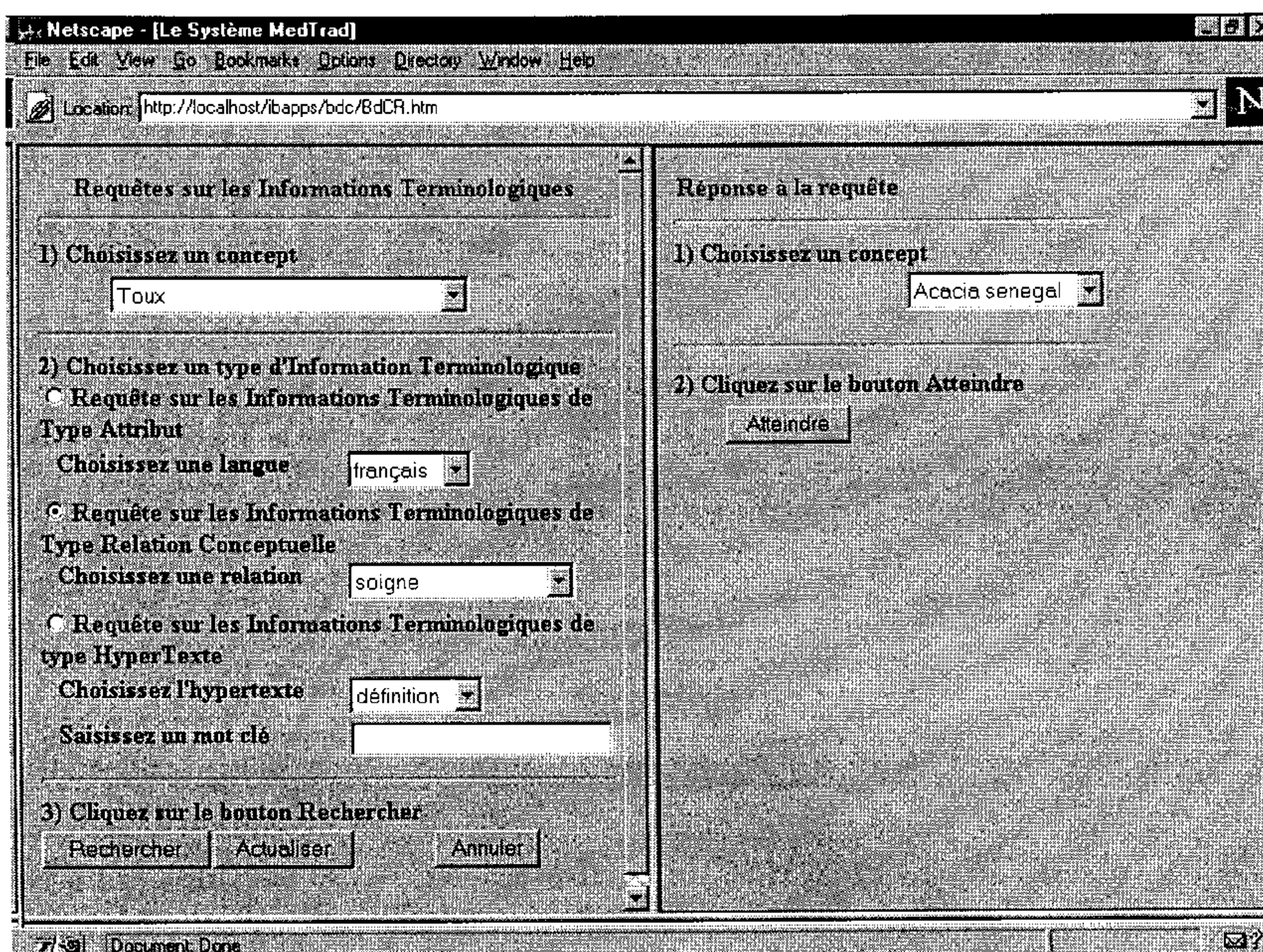
6.2.9.1 Requête sur les informations terminologiques

Nous présentons deux exemples de requête sur les informations terminologiques de type relation conceptuelle. Les préoccupations de l'utilisateur sont exprimées comme suit :

- 1) Quelles sont les plantes qui soignent la toux?
- 2) Quelle sont les maladies que soigne Acacia senegal?

Dans le premier cas, il faut sélectionner dans la liste déroulante le concepts toux, dans le second cas c'est Acacia senegal. Dans les deux cas la relation conceptuelle est soigne.

La figure 6.16 correspond à la requête 1).

FIG. 6.16 – *Requête sur les informations terminologiques.*

L'activation du bouton Rechercher affiche la liste des plantes qui soignent la toux (figure 6.16). Un clic sur le bouton Atteindre permet de naviguer dans la BdC de Medtrad à travers les fiches terminologiques de la plante qui aura été sélectionnée.

6.2.9.2 Requête sur les caractères définitoires

La figure 6.17 présente un exemple de requête sur les caractères définitoires simples. La question de l'utilisateur est :

Quels sont les plantes qui ont plus de 5 m de hauteur?

De même, l'activation du bouton Rechercher affiche la liste des plantes qui répondent aux critères de sélection.

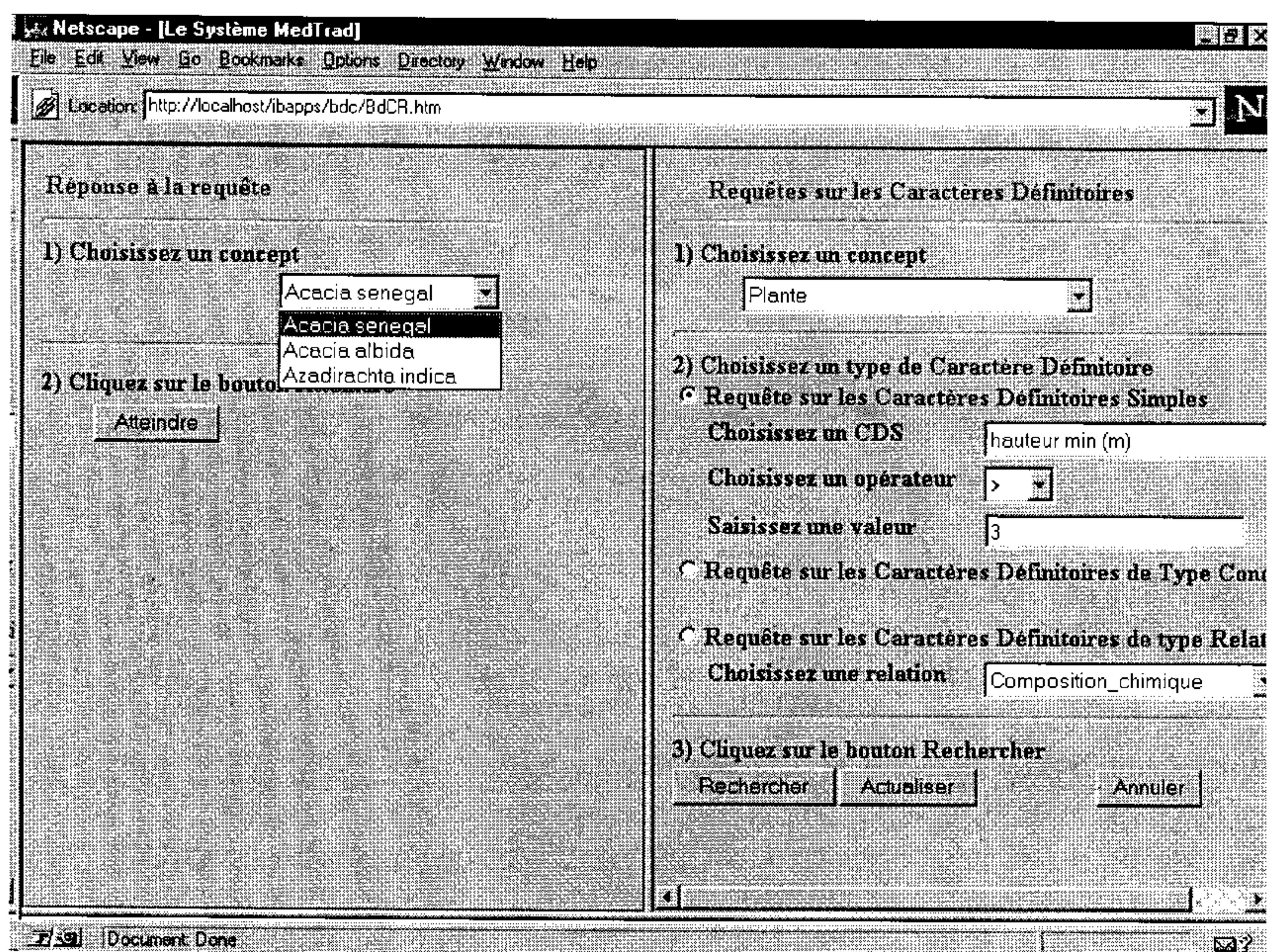


FIG. 6.17 – Requête sur les caractères définitoires.

6.3 Extension future de MEDTRAD : MEDTRAD+

Afin de rendre Medtrad plus performant et efficace, nous proposons d'étendre ses fonctionnalités pour qu'il devienne un véritable système de diagnostic et de thérapeutique. Cette extension est réalisée par un couplage avec un système expert.

A travers cette extension de MEDTRAD, nous présentons une utilisation possible d'une BdC comme composante explicative de la connaissance statique (savoir de l'expert du domaine) d'un système à base de connaissances. Cette connaissance permet d'une part la modélisation des aspects descriptifs et stucturels du domaine et d'autre part la description d'objets, de concepts, d'attributs et de relations entre les divers élément du domaine [HOC 91].

6.3.1 Architecture du système MEDTRAD+

La figure 6.18 présente les principales composantes du système MEDTRAD+, i.e. l'interface de communication, la BdC, le Système Expert et l'interface de communication BdC/SE.

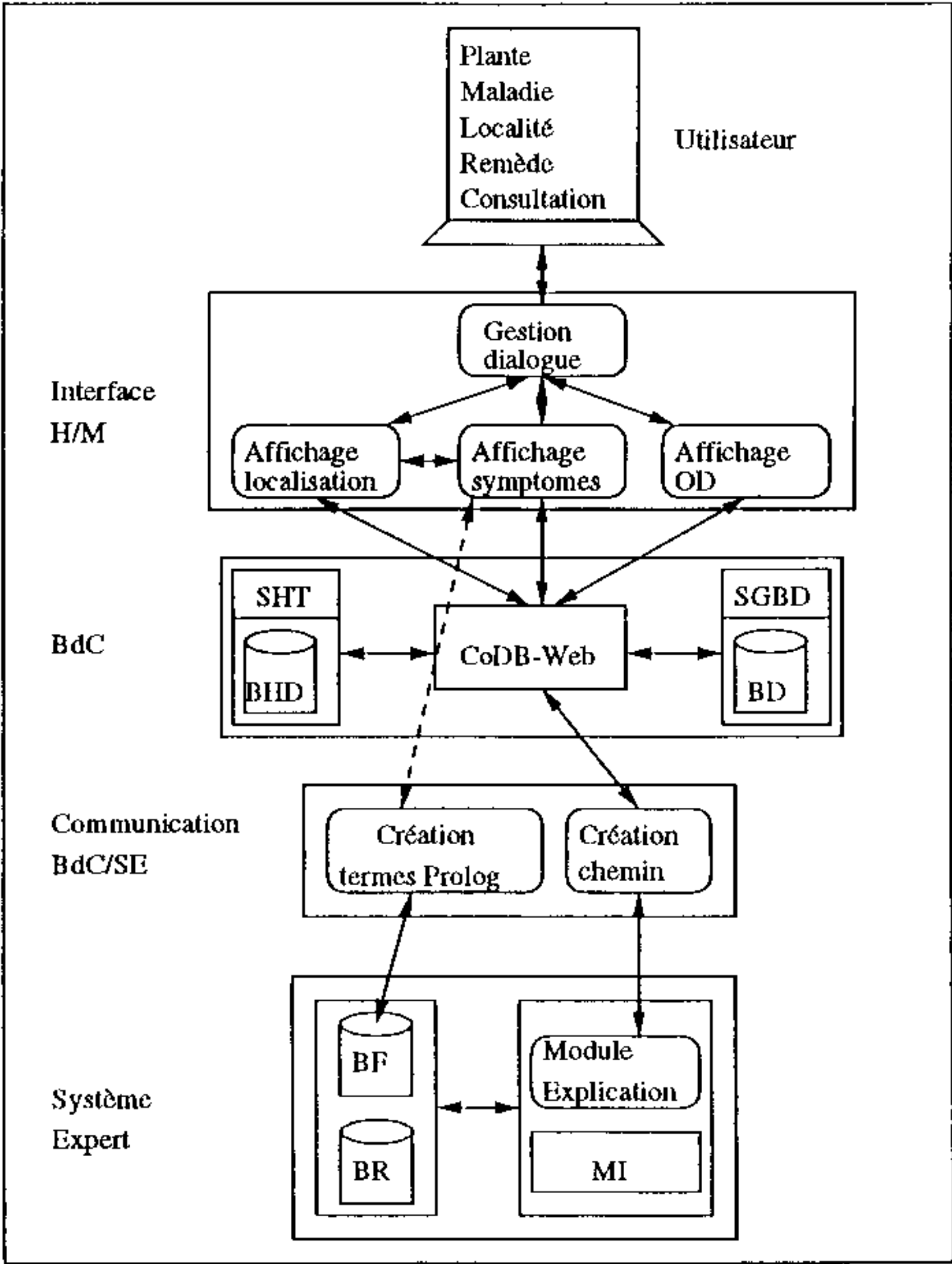


FIG. 6.18 – Architecture du système MEDTRAD+.

6.3.2 Fonctionnement du système MEDTRAD+

Nous présentons ci-dessous le fonctionnement du système MEDTRAD+ en insistant plus particulièrement sur son mécanisme de diagnostic des maladies, de proposition d'une thérapeutique adaptée, et sur sa capacité à expliquer son raisonnement sous forme d'un parcours orienté dans la consultation des objets documentaires de la BdC.

6.3.2.1 L'interface Homme/Machine

Elle gère le dialogue entre l'utilisateur et MEDTRAD+. Les quatre premiers choix qu'il propose permettent d'accéder directement aux concepts grâce au module d'affichage (Affichage OD) des objets documentaires de la base d'hyperdocuments (cf. 6.18).

Le dernier choix permet d'apprendre à l'utilisateur comment poser un diagnostic et comment prescrire une thérapeutique grâce au Système Expert.

Pour effectuer une consultation les actions suivantes sont exécutées dans l'ordre :

1. Le choix de la localisation des symptômes :

l'utilisateur doit indiquer ses symptômes en choisissant une parmi celles présentées par le module **Affichage localisation**; on distingue deux sortes de symptômes :

- les symptômes localisés dans les différentes parties du corps : tête, membre et corps; par exemple les symptômes maux de tête et bourdonnements d'oreille concernent la tête.
- les symptômes physiologique qui n'ont pas une localisation précise; par exemple les symptômes perte de poids et malaise général.

2. L'extraction des symptômes de la BdC :

grâce au système CoDB-Web, il y a l'extraction de la BdC des symptômes associée à la localisation qui a été choisie; ces symptômes sont présentés à l'utilisateur par le module **Affichage symptômes**.

3. Le choix des symptômes :

l'utilisateur choisit les symptômes qui correspondent à sa maladie.

Ces trois étapes peuvent être répétées plusieurs fois. Les symptômes ainsi recueillis sont transformées en termes Prolog et stockés dans un fichier. Ce fichier de termes sert à initialiser la base des faits du Système Expert.

6.3.2.2 La base de concepts

Nous avons présenté au § 6.2 le contenu de la BdC de MEDTRAD+. Elle est sollicitée à deux niveaux : premièrement pour l'extraction des symptômes, deuxièmement pour la présentation des objets documentaires à afficher par le module **Affichage OD**.

6.3.2.3 L'interface BdC/SE

Elle assure la communication entre la BdC et le Système hypertexte par l'intermédiaire de deux modules :

- le module **Création termes Prolog** transforme les symptômes recueillis auprès de l'utilisateur en termes Prolog et les stocke dans un fichier qui alimente la base des faits

du Système Expert.

- le module `Création chemin` transforme la trace du raisonnement (constituée de termes prolog) fournie par le module `Explication` du Système Expert, en un fichier ASCII; il envoie ensuite ce fichier sous forme de requête au système CoDB-Web afin de déterminer les objets documentaires à présenter à l'utilisateur.

6.3.2.4 Le Système Expert

Le Système Expert diagnostique les maladies à partir des symptômes recueillis auprès du patient; ces symptômes sont transformés en termes Prolog par le module `Création termes Prolog` afin d'initialiser la base des faits. Grâce aux règles de diagnostic, le SE détermine la maladie, puis il propose une liste de plantes médicinales susceptibles de soigner cette maladie en se servant des règles thérapeutiques.

La trace de son raisonnement est stockée dans le module `Explication`. Les concepts qui figurent dans les règles déclanchées serviront à la création d'un chemin de parcours orienté dans la base d'hyperdocuments afin d'avoir une explication plus complète et hypertextuelle.

a) Mécanisme de raisonnement du SE

La représentation des connaissances du domaine est basée sur le formalisme des règles de production. La base de connaissances contient les faits et les règles exploités par un moteur d'inférences en chaînage avant. Ce mode de fonctionnement permet de partir des symptômes pour aboutir à une proposition de plantes nécessaires pour soigner la maladie diagnostiquée. Si le raisonnement aboutit à un succès, une liste de plantes permettant de soigner la maladie déduite est proposée. On distingue deux sortes de règles :

- Les règles de diagnostic : ces règles expriment les symptômes à vérifier pour déterminer une maladie; elles sont de la forme suivante :

```
SI symptôme_1
  ET symptôme_2
  ...
  ET symptôme_n
ALORS maladie
```

Exemple : règle de diagnostic pour l'hypertension artérielle

```
SI vertiges
```

ET maux de tête
 ET malaise général
 ET bourdonnement d'oreille
 ALORS hypertension artérielle

- Les règles thérapeutiques : ces règles expriment une liste de plantes à proposer pour guérir une maladie donnée; elles de la forme suivante :

SI maladie
 ALORS plante_1
 ET/OU plante_2
 ...
 ET/OU plante_m

Exemple : règle thérapeutique pour l'hypertension artérielle

SI hypertension artérielle
 ALORS Anacardium occidentale
 ET Cathanrantus roseus

Une liste détaillée de ces règles est présentée à l'annexe C.

b) Explication du raisonnement du SE

Au lieu de se contenter de lister les faits contenus dans les règles déclanchées, comme c'est le cas dans les SE classiques, l'explication est fournie à l'utilisateur sous la forme d'un chemin de consultation de la BdC. Dans la BdC, les faits sont représentés par des concepts liés les uns aux autres par des relations sémantiques qui permettent de comprendre le "comment" des faits déduits.

L'exemple ci-dessus donne le chemin suivant :

Symptômes (maux de tête, malaise général, bourdonnement d'oreille) $\Rightarrow$
 Maladie (hypertension artérielle) $\Rightarrow$ Remède (hypotenseur) $\Rightarrow$
 Plantes (Anarcadium occidentatle, Cantharantus roseus)

Seuls ces concepts seront accessibles dans la base d'hyperdocuments.

6.4 Conclusion

Dans ce chapitre, nous avons présenté une instanciation de la BdC à travers l'application MEDTRAD qui permet de capitaliser la connaissance ancestrale en médecine traditionnelle

africaine. Les concepts manipulés dans ce système sont ceux utilisés en médecine traditionnelle (plante, maladie, localité et remède).

Nous avons ensuite proposé une extension possible de MEDTRAD, basée sur un couplage avec un système expert, afin qu'il devienne un système fiable d'aide aux tradipraticiens africains dans leur processus de diagnostic des maladies et de proposition d'une thérapeutique à base de plantes médicinales.

Dans ce couplage, la BdC peut être perçue comme composante explicative d'un système expert. L'explication de son raisonnement est fourni sous la forme d'un chemin de navigation dans la base d'hyperdocuments afin de prendre connaissance de la sémantique des concepts qui figurent dans les règles déclanchées.

Une maquette de MEDTRAD a été réalisée mais elle n'a pas encore été validée sur le terrain.

Conclusion

Nous avons présenté dans cette thèse nos travaux concernant la modélisation d'une base de concepts qui se situe dans le cadre d'une base de connaissances généralisées.

L'objectif de nos recherches a été de concevoir cette base de concepts de manière à ce qu'elle apporte une aide à la structuration et à l'apprentissage des concepts d'un domaine donné le plus simplement et le plus naturellement possible.

L'approche que nous avons adoptée pour la modélisation de cette base de concepts s'appuie sur les concepts et techniques de la terminologie pour décrire les relations sémantiques qui existent entre les concepts.

Pratiquement son implantation est fondée sur la coopération entre le système de gestion de base de données MS ACCESS et le système hypermédia WWW, cadre idéal de vulgarisation de la connaissance stockée dans la BdC.

L'aspect base de données confère à la BdC les fonctionnalités des SGBD : gestion classique des données, langage d'interrogation, sécurité et cohérence des données.

L'aspect hypertexte apporte à la BdC son interface de consultation, sa souplesse de représentation des informations et surtout sa facilité de navigation.

Outre la proposition du modèle de BdC fondé sur une architecture à trois niveaux : terminologique, définitoire et représentation, nous avons également proposé une méthode de gestion hypertextuelle des textes complexes et volumineux stockés dans les bases de données.

En effet, nous avons opté pour une gestion séparée de leur contenu informatif et de leur structure logique (constitué d'unités informatives et de liens). L'hypertexte résultant est obtenu par un processus d'hypérisation de ces deux aspects, stockés séparément dans la BdC.

L'intérêt d'une telle approche, est d'une part de faciliter la gestion des hyperdocuments fondés sur les BD notamment dans la maintenance et la vérification de la cohérence de la base

d'hyperdocuments, d'autre part, de permettre la création de points de vue correspondant aux informations bien maîtrisées de la BdC qu'un utilisateur souhaite manipuler.

Tout ceci est mis en œuvre grâce à CoDB-Web qui est le système de gestion et d'exploitation hypertextuelle d'une BdC fondée sur le Web. Nous l'avons développé en JavaScript Intrabuilder, langage de développement d'applications de gestion dynamique de données basées sur le Web.

Une instantiation de la BdC a été réalisée à travers une application appelée MedTrad, portant sur la médecine traditionnelle africaine. Cette application permet de capitaliser la connaissance ancestrale africaine sur la thérapie à base de plantes médicinales.

L'extension de Medtrad par couplage avec un système expert est une utilisation possible d'une BdC comme composante explicative de la connaissance statique d'un système à base de connaissances. Elle permettra d'aider les tradipraticiens africains pour leur prise de décision dans le diagnostic des maladies, et dans la proposition d'une thérapeutique adaptée. On peut encore l'améliorer en lui associant une fonction d'apprentissage de l'expérience acquise au cours des manipulations successives.

Le système CoDB-Web et l'application MedTrad sont en cours de développement. Tous les algorithmes d'hypérisation de la BdC ont été entièrement implémentés dans l'environnement de JavaScript Intrabuilder, ceci a permis la génération de la base d'hyperdocuments d'un prototype de MedTrad.

Nous pensons que le modèle de base de concepts décrit dans ce mémoire peut être amélioré au niveau des aspects suivants :

- poursuite du développement du système CoDB-Web de manière à y intégrer les aspects non encore implémentés (gestion de la BD, gestion des points de vue, ...)
- prise en compte d'autres SGBD quelque soit la plate forme (Windows, UNIX, ...) et
- utilisation du formalisme des logiques terminologiques pour l'instantiation de la BdC, afin de tirer partie de leur opérateur de subsumption permettant la classification automatique des concepts en hiérarchie de concepts ainsi que la déduction de nouvelles relations d'héritage entre concepts.

Bibliographie

- [ABI 87] S. Abiteboul, S. Grumbach, "Bases de Données et Objets Structurés", TSI, 1987, vol. 6, n°5, pp 383-403.
- [ALA 95] N. Alaya: "De la gestion des documents à la gestion des connaissances", actes du congrès INFORSID'95, Grenoble, 4-7 juin 1995, pp 99-112.
- [AND 93] E. Andonoff, C. Mendiboure, C. Morin: OIIQL: "Un langage graphique pour l'interrogation d'une base de données", Ingénierie des systèmes d'informations, Vol. 1, n°2, 1993, pp 203-227.
- [ANO 96] L. Anonson: "HTML3, la toute première fois", MiniDoc, Edition Simon & Schuster Macmillan, avril 1996.
- [ASC 90] Axiome: "Modélisation dans la conception des systèmes d'information", Editions Masson 1990.
- [BLA 93] J. M. Blanco, A. Illarramendi, A. Goñi, J. M. Pérez: "Using a terminological System to integrate relational databases", Information System Design And Hypermedia, BIWIT 94, Cépaduès-Edition, pp 321- 334.
- [BON 94] M. Bonjour, G. Falquet, M. Léonard: "Base de concepts et intégration de bases de données", actes du congrès INFORSID'94, Aix-en-Provence, 17-20 mai 1994, pp 263-280.
- [BON 86] G. Bonfils, S. Miranda, M. Rueher: "Systèmes Experts et SGBD: un état de l'art et une réalisation par couplage d'un SGBD Relationnel avec Prolog", actes des journées AFCET de la Rochelle, septembre-octobre 1986, Edition Eyrolles, pp 172-181.
- [BOR 95] A. Borgida: "Description Logics in Data Management", IEEE Transaction of Knoledge and Data Engineering, vol. 7 october 1995, pp 671-682.
- [BOR 96] Borland: "Borland Intrabuilder pour Windows 95 & Windows NT", manuel d'introduction, 1996.

- [BOU 94] M. Bouzeghoub, G. Gardarin, P. Valduriez : "Les Systèmes de Bases de Données Objet : du C++ à Merise Objet", Edition Eyrolles, pp 113-153.
- [BRO 91] P. J. Brown : "Hypertext: dreams and reality", in Hypermedia/Hypertext and Object Oriented Databases, Ed Chapman & Hall, 1991.
- [BRO 96] K. M. Brou : "Gestion des connaissances du domaine: une approche fondée sur les bases de concepts", rapport interne ICOG, 19 novembre 1996.
- [BRO 97] K. M. Brou : "Base de concepts: une approche pour la gestion et la manipulation des concepts d'un domaine: le système CoDB-Web", rapport interne ICOG, 10 février 1997.
- [BUS 45] V. Bush : "As we may think", Atlantic Monthly, Vol. 176, july 1945, pp 101-108.
- [CAM 96] M. V. Campenhoudt : "Abrégé de terminologie multilingue", Document Web, Termisti@infoboad.be.
- [CHE 90] J. P. Cheiney, G. Kierman, R. le Maout, F. Pasquer, "Un sous système de gestion d'objets complexes pour un SGBD relationnel", TSI, Vol. 9 n° 1, janvier 1990, pp 19-31.
- [CHR 88] C. Chisment, G. Zurfluh, "Bases d'informations généralisées - Systèmes multimédias - Architecture, Manipulation, Interrogation", Ecole d'automne Bases de données, 1988.
- [COD 70] E. F. Codd, "A Relational Model for Large Shared Data Banks", CACM, Vol. 13, n° 6, june 1970, pp 377-387.
- [CON 87] J. Conklin : "Hypertext: An Introduction and Survey", IEEE Computer, Vol. 20 n°19, septembre 1987.
- [COR 87] J. M. Cornily : "Couplage Prolog/base de données : comparaison de trois approches classiques et d'une approche parallèle", Des Bases de Données aux Bases de Connaissances, AFCET 1987, Edition Editest, pp 229-245.
- [DAN 90] M. C. Daniel-Vatonne, "Hypertextes: des principes communs et des variations", TSI, vol 9, n° 6, 1990, pp 475-492.
- [DEL 87] N. M. Delisle, M. D. Schwartz: "Neptune: a partitioning concept for hypertext", ACM transaction on Office Information System, Vol. 5, n°2, april 1987, pp 168-186.
- [DEL 85] C. Delobel, M. Adiba : "Bases de données et systèmes relationnels", Edition Dunod Informatique, 1985.

- [DES 95] Renaud Deschamps, "Bases de connaissances généralisées : une approche fondée sur un modèle hypertexte expert", Thèse de doctorat de l'Université Paul Sabatier de Toulouse, janvier 1995.
- [DES 90] R. Deschamps, A. Hocine, N. Lavielle, K. Tamine "Deux approches de couplage entre base de connaissances centrées objet et base de données relationnelle", actes du congrès INFORSID'90, Biarritz, 15-17 mai 1990.
- [DIE 95] A. W. Diene, A. K. Konaté: "Exploitation par système expert d'une base de concepts", Mémoire de Maîtrise Informatique, Université de Saint-Louis, juillet 1995.
- [ELL 87] A. Ellul, "Couplage entre mécanisme de déduction et base de données", Des Bases de Données aux Bases de Connaissances, AFCET 1987, Edition Editest, pp 207-227.
- [EST 90] T. Estier, G. Falquet, "QFE: un générateur d'interfaces pour l'interrogation des bases de données à l'aide de contextes sémantiques", actes du congrès INFORSID'90, Biarritz, mai 1990, pp 189-215.
- [FAL 96] G. Falquet, J. Guyot, I. Prince: "Generating Hypertext Views on Databases", actes du congrès INFORSID'96, Bordeaux, 4-7 juin 1996, pp.
- [FID 88] J. Fiderio: "A Grand Vision", BYTE, octobre 1988, pp. 236-246.
- [FIS 89] G. Fischer, R. McCall: "Janus: Integrating hypertext with a knowledge-based design Environment", Hypertext'89 Proceedings, november 1989, pp 105-116.
- [FOR 93] E. N. Forte, J. M. Herzog, M. C. Wentland: "Identification de concepts et parcours orienté dans un hypertexte pédagogique", Environnement Interactif d'Apprentissage avec Ordinateur, Edition Eyrolles, pp 169-180.
- [FOR 94] N. Forest, "Bases de connaissances généralisées : le modèle de gestion d'informations de types hypertexte", Thèse de doctorat de l'Université de Pau et des pays de l'Adour, janvier 94.
- [FOR 96] J. P. Forestier, "Les SGBD dans un contexte Web", Databases Journal, n°4 et n°5, juillet-septembre-octobre 1996, pp 8-15.
- [FRA 92] C. Frasson, F. Houts, P. Obenson: "Interface visuelle pour l'aide au diagnostic médical en médecine traditionnelle", ICO, Vol. 4, n° 1 et 2, 1992, pp 17-25.
- [FRI 88] Mark Frisse: "From text to hypertext, Byte, octobre 1988, pp. 247-254.
- [GAL 87] GALACSI: "Système d'information et base de données", Edition Bordas Informatique, mars 1987.

- [GAR 83] G. Gardarin : "Base de données, les systèmes et leur langages, Editions Eyrolles 1983.
- [GAR 90] G. Gardarin, P. Valduriez : "SGBD avancées, Base de données objets, déductives, réparties, Editions Eyrolles 1990.
- [GOU 92] D. Gouadec : "Terminologie et Terminotique, Outils, Modèles et Méthodes", actes de la première Université d'automne en Terminologie, Rennes 2-21 au 26 septembre 1992, Edition la Maison du dictionnaire, pp 67-120.
- [HAB 88] Henri Habrias : "Le modèle relationnel binaire, méthode I. A. (NIAM)", Editions Eyrolles 1988.
- [HAL 87] F. G. Halasz, "Reflection on NoteCards: seven Issue for the Next Generation of Hypermedia Systems, ACM Hypertext'87 Proceedings, Chapel Hill, N. C., november 1987.
- [HEB 89] G. Hebrail, B. Mans, F. Olleon, "Couplage du Moteur d'Inférence GENESIA2 et du SGBD Oracle : principe, mise en œuvre et application", Modèles et Bases de Données, octobre 1989, pp 115-126.
- [HOC 91] A. Hocine : "Explication et gestion hypertexte des ACPE, un système expert d'aide au choix de produits d'épargne", ICO'91, Montreal 1991, pp 127-134.
- [HOC 95] A. Hocine, N. Lavielle : "Méthodologie de conception d'un document hypertexte expert", rapport interne, 1995.
- [HOU 88] G. Houbart : "Hypercard : Un hommage à Smalltalk", Micro-Systèmes, janvier 1988, pp 101-108.
- [HOP 93] Hoppe T., Kinderman C., Quani J., Schniedel A., Ficher M. "BACKV5: Tutorial & manual", Technische Universität Berlin, Institut Für Software und Theoretische Informatik, Projekt KIT-BACK, Berlin 10, Germany 1993.
- [JUL 88] C. Julien : "Bases d'Informations Généralisées : contribution à l'étude des mécanismes de consultation d'objets multimédia", thèse de doctorat de l'IRIT de l'Université Paul Sabatier de Toulouse, 25 octobre 1988.
- [KEM 91] G. kemp, D. Melvin, "A graphical interface for object-oriented database", Hypermedia/Hypertext and Object-Oriented Database, pp 307-322.
- [LAR 96] O. Larab, A. Benharkat : "Propriétés sémantiques et raffinement des relations de correspondances dans l'intégration de schémas hétérogènes basée sur le modèle terminologique", actes du congrès INFORSID'96, Bordeaux, 4-7 juin 1996, pp 179-197.

- [LAR 88] E. Larcher, N. Klingsor, B. Labjen: "L'hypertexte, plus qu'une technique, une vraie révolution culturelle", Décision n°204, 31 octobre 1988, pp. 18-25.
- [LEB 93] F. Lebastard: "Bases de données experte et grandes base de connaissances: l'état de l'art", Génie logiciel & Systèmes Experts, n°31, juin 1993, pp 4-16.
- [LEM 96] L. Lemay, M. Moncur: "JavaScript", Le Labo, Edition Simon & Schuster Macmillan, novembre 1996.
- [MAI 94] J. Le Maître: "Cours Base de données textuelles", actes du congrès INFORSID'94, Aix-en-Provence, 17-20 mai 1994.
- [MAR 96] Y. Marcoux: "Place de SGML parmi les nouvelles architectures documentaires", document Web, <http://tornado.ere.umontreal.ca/marcoux/ottawa/marcoux.html>.
- [MAY 90] G. Maynard, M. Lô, D. Fortin "Plantes médicinales du sahel", Série Etudes et Recherches, 1990.
- [MEN 94] C. Mendiboure, DOHQL: "une interface avancée pour l'interrogation de base de données orientées-objets et documentaire", Thèse de doctorat de l'Université Paul sabatier de Toulouse, février 94.
- [MIR 96] I. Mirbel: "Une méthode d'intégration de schémas de conception", actes du congrès INFORSID'96, Bordeaux, 4-7 juin 1996, pp 161-178.
- [MOE 96] T. Moeglin: "Intégration dans un hyperdocument d'un mécanisme expert d'explication de concepts", rapport de DEA d'informatique de l'Université du Maine, juin 1996.
- [MOR 91] C. Morin V. Rougier, C. Sallaberry, "Un langage hypertexte pour une base de données multimédia", acte du congrès INFORSID'91, Paris, pp 41-62.
- [MUL 96] Projet MULTEXT: "Multilingual Text Tools and Corpora", document Web, <http://www.lpl.univ-aix.fr/projects/multext/>
- [NAN 89] : J. Nanard, M. Nanard: "MacWeb, un outil pour élaborer des documents", Woodman'89, Bigre, n° 63-64, Mai 1989, pp 18-29.
- [NAN 93] J. Nanard, M. Nanard, A. Massotte, A. Djemaa, A. Joubert, H. betaille, J. Chauché: "Integrating knowledge-based hypertext and database for task-oriented access documents", actes DEXA'93, pp 721-732.
- [NAR 89] J. Nardot, G. Pujolle, A. Teste: "Une interface relationnelle de type hypertexte", acte du congrès INFORSID'89, Nancy, 23-26 mai 1989, pp 223-244.

- [NEL 88] T. H. Nelson : "Managing Immense Storage : Project Xanadu provides a model for the possible future of mass storage", Byte, Vol. 13, n°1, jannuary 1988, pp 225-238.
- [O2W 96] O2 Technology : "O2 web user Manual", release 4.6, January 1996.
- [ORA 89] A. Abdellatif, J. Lebihan, M. Limame, "Oracle Le Système de gestion de base de données relationnel", Edition Eyrolles, 1989.
- [ORA 89] Oracle Corporation, "Reference", Oracle for Macintosh, 1989.
- [ORA 96] Oracle Corporation, "Guide des solutions mico et Internet", 1996.
- [OUN 95] I. Ounis : "Une Dénotation pour les Graphes Conceptuels : comparaison avec les Logiques Terminologiques en Recherche d'Informations", actes du congrès INFOR-SID'95, 1995, pp 147-164.
- [PIN 91] J. M. Pinon, F. Biennier : "Base de documents hypermédias", Actes du congrès BD'91, Alger, 1991, pp 162 à 214.
- [REM 89] C. Remy : "Les SGBD : au-delà du relationnel", Micro-Systèmes, mai 1989, pp 254-266.
- [RIC 89] H. Richy, J. Nanard : "Spécification et contrôle orienté objet de l'accès concurrent à des documents structurés", Woodman'89, Bigre n° 63-64, mai 1989, pp 152-164.
- [ROZ 96] D. Rozenshtein : "Gestion d'arbres en SQL", Databases Journal, n°1, 1996, pp 24-34.
- [RUD 96] D. rudloff : "Modélisation conceptuelle et optimisation des requêtes dans un système d'interface en langue naturelle à une base de données", Rapport interne ERIC, ftp eric.u-strasbg.fr.
- [SAI 94] J. C. Saillaird : "Hyperbase amélioré", rapport interne ICOG, 1994.
- [SAO 96] A. Saondi, G. Nachouki : "Utilisation d'un système terminologique pour la rétro-conception des bases de données relationnelles", actes du congrès INFORSID'96, Bordeaux, 4-7 juin 1996, pp 199-214.
- [SAV 88] Jacques Savoy : "Le livre électronique EBOOK3", TSI, Vol. 7, n°5, 1988, pp 437-450.
- [SAV 90] Jacques Savoy : "Les sources de l'hypertexte : une bibliographie commentée", TSI, Vol. 9, n°6, 1990, pp 515-524.

-
- [SAV 90] Jacques Savoy, Daniel Desbois: "Hypertexte et réseaux d'inférences bayesiens", ICO'91, Montreal, 29 avril-2 mai 1991, pp 140-148.
- [SCH 90] H. A. Schutt, N. Streitz, "HyperBase: a hypermedia engine based on a relational database management system", actes de Hypertext: concepts, systems and applications, Paris, novembre 1990, pp 96-108.
- [SIJ 93] A. Sijelmassi: "Les plantes médicinales du Maroc", Edition Le Fennec, 3e édition 1993.
- [TEC 96] Technique/SQL Tips: "Qu'est ce qu'une vue?: Database Journal, n° 4 et 5 juillet 1996, pp 28-30.
- [UTT 89] K. Utting, N. Yankelowich: Context Orientation in Hypermedia Networks, ACM Transaction on Information Systems, Vol. 7, n°1, january 1989, pp 58-84.

Algorithmes d'hypérisation

Les algorithmes que nous présentons ici ont été programmés en JavaScript Intrabuilder ; nous les avons utilisés pour la génération de la base d'hyperdocuments de Medtrad.

Soient :

- Ci un concept de numéro i
- fit.html le fichier HTML contenant ses informations terminologiques
- fcd.html le fichier HTML contenant ses caractères définitoires

1. Génération de toutes les fiches terminologiques de la BdC

ALGORITHME HypFiche(i)

DEBUT

Ouvrir_bd(ALL)

SELECT numconcept INTO tablenumconcept
FROM Concept

POUR t=1, NBC

HypFicheIT(tablenumconcept[t])

HypFicheCD(tablenumconcept[t])

FINP

Fermer_bd(ALL)

FIN

2. Génération d'une fiche information terminologique

```
ALGORITHME HypFicheIT(i)
DEBUT
    Ouvrir_bd(ALL)
    Créer_fichier(fit.html.htm, i)
    Ouvrir_fichier(fit.html, "w")

    • Création des informations terminologiques de type attribut (ITA)
        SELECT terme, langue INTO ITA
        FROM Concept
        WHERE numconcept = i

        Ecrire_fichier(fit.html, "Terme: ")
        Ecrire_fichier_ln(fit.html, ITA[1])
        Ecrire_fichier(fit.html, "Langue: ")
        Ecrire_fichier_ln(fit.html, ITA[2])

    • Création des informations terminologiques de type hypertexte (ITIHT)
        Hyptexte(i, definition)
        Hyptexte(i, contexte)
        Hyptexte(i, image)

    • Création des informations terminologiques de type relation conceptuelle (ITRC)
        Hyprc(i, Concepts génériques)
        Hyprc(i, Sous-Concepts)
        Hyprc(i, Synonymes)
        Hyprc(i, Homonymes)
        Hyprc(i, Concepts liés)

    • Enregistrement du fichier fit.html dans la table Unité informative
        Fermer_fichier(fit.html)
        Ajouter_fichier(fit.html, Unité informative, i)
        Fermer_bd(ALL)
FIN
```

3. Hypérisation d'une information terminologique de type hypertexte

```
ALGORITHME Hyptexte(i, typelien)
```

DEBUT

Ouvrir_bd(ALL)

fit.htm ← trouver_fichier(i)

- Récupération du texte de l'information terminologique de type hypertexte
typelien peut avoir comme valeur définition ou contexte, image ou description

SELECT typelien INTO htexte

FROM Concept

WHERE numconcept = i

- Création de l'ancre correspondant à l'ITHT dans l'OD source fit.htm
nomfht.htm est le fichier hypertexte associé à l'ITHT typelien

nomfht ← créer_fichier(typelien, i)

Insérer_lien(fit.htm, typelien, , nomfht.htm)

Copier_fichier(nomfht.htm, htexte)

Ouvrir_fichier(nomfht.htm, "w")

- Récupération des liens de l'information terminologique de type hypertexte

SELECT début, longueur, étiquette, numui2 INTO tablelien

FROM Lien

WHERE ((type = typelien) OR (type = "description")) AND numui IN

(SELECT numui

FROM unité_informative

WHERE numconcept = i);

- Création des liens de référence dans le fichier hypertexte nomfht.htm

TANTQUE EOF(nomfht.htm) différent de VRAI

POUR t = 1, NBlien

début ← tablelien[t,1]

longueur ← tablelien[t,2]

positionner(nomfht.htm, début)

ancresource ← Créer_ancre(nomfht.htm, début, longueur)

Insérer_lien(nomfht.htm, ancresource, étiquette, tabelien[t,4])

FINP

FINT

- Enregistrement du fichier nomfht.htm dans la table Unité informative

```
Fermer_fichier(nomfht.htm)
Ajouter_fichier(nomfht.htm, Unité informative, i)
Fermer_bd(ALL)
FIN
```

4. Création des liens conceptuels

```
ALGORITHME Hyprc(i, nomrco)
DEBUT
  Ouvrir_bd(ALL)
  fit ← trouver_fichier(i)

  • Récupération des valeurs de l'ITRC Synonyme ou Homonyme,
  Concept générique ou Sous-concept
  SI nomrco <> "Concept lié" ALORS
    Ecrire_fichier_ln(fit.htm, nomrco)

    SELECT numconcept, terme INTO tablerc
    FROM Concept
    WHERE numconcept IN
      (SELECT numconcept2
      FROM Relation_conceptuelle
      WHERE (numconcept = i) AND (nomrc = nomrco);
    POUR t=1, NBRC
      Insérer_lien(fit.htm, tablerc[t,2], tablerc[t,1])
      Ecrire_fichier_ln
    FINP
  SINON
```

```
  • Récupération des valeurs de l'ITRC Concept lié
    Ecrire_fichier_ln(fit.htm, "Concepts liés")

    SELECT numconcept, terme, nomrc Into tablecl
    FROM Concept
    WHERE numconcept IN
      (SELECT numconcept2
      FROM Relation conceptuelle
      WHERE (numconcept = i) AND (nomrc NOT IN [Synonyme, Homonyme,
```

Concept générique, Sous-concept]);

```

POUR t=1, NBCL
  Ecrire_fichier(fit.htm, tablecl[t,3])
  Insérer_lien(fit.htm, tablecl[t,2], tablecl[t,1])
  Ecrire_fichier_ln
FINP
Fermer_bd(ALL)
FIN

```

5. Création d'une fiche caractère définitoire

ALGORITHME HypFicheCD(i)

DEBUT

Ouvrir_bd(ALL)

Créer_fichier(fcd.htm)

Ouvrir_fichier(fcd.htm, "w")

- Caractères définitoires simples

Ecrire_fichier_ln(fcd.htm, Caractères définitoires simples)

SELECT nomcs, valeurs INTO tablecds

FROM Concept, Comporte_cds, CD_simple

WHERE (Concept.numconcept = i) AND

(Concept.numconcept = Comporte_cds.numconcept) AND

(Comporte_cds.numcds = CD_simple.numcds);

POUR t=1, NBCDS

Ecrire_fichier(fcd.htm, tablecds[t,1] + ": ")

Ecrire_fichier_ln(fcd.htm, tablecds[t,2])

Lien_description(i, "CDS")

FINP

- Caractères définitoires de type concept

Ecrire_fichier_ln(fcd.htm, "Caractères définitoires de type concept")

SELECT numconcept, terme, Into tablecdtc

```

FROM Concept
WHERE numconcept IN
  (SELECT numconcept2
   FROM Relation__conceptuelle
   WHERE (numconcept = i) AND (nomrc = "CDTC"));

```

```

POUR t=1, NBCDTC
  fichierdest ← trouver_fichier(tablecdtc[t,1])
  Insérer_lien(fcd.htm, tablecdtc[t,2], ,fichierdest)
FINP

```

- Caractères définitoires de type relation

```

Ecrire_fichier_ln(fcd.htm, "Caractères définitoires de type relation
de type relation")

```

```

SELECT numcdtr, numenreg INTO tablecdtr
FROM Comporte_cdtr, CD_type_relation
WHERE (numconcept = i) AND (CD_type_relation.numcdtr =
  Comporte_cdtr.numcdtr);

```

```

Créer_table(tablecdtr)
Lien_description(i, "CDTR")
FINP

```

- Enregistrement du fichier fcd.htm dans la table Unité_informative

```

Fermer_fichier(fcd.htm)
Ajouter_fichier(fcd.htm, Unité_informative, i)
Fermer_bd(ALL)

```

FIN

6. Création d'un lien de description

```

ALGORITHME Lien_description(i, typecd)
DEBUT
  Ouvrir_bd(ALL)
  fcd.htm ← trouver_fichier(i)

```

- Récupération des liens d'un caractère définitoire

```
SELECT début, longueur, étiquette, numui2 INTO tablecd
FROM Lien
WHERE (type = typecd) AND numui IN
  (SELECT numui
   FROM unité_informative
   WHERE numconcept = i);
```

- Création des liens de référence dans le fichier hypertexte fcd.htm

```
TANTQUE EOF(fcd.htm) différent de VRAI
  POUR t = 1, NBlien
    début ← tablecd[t,1]
    longueur ← tablecd[t,2]
    positionner(fcd, début)
    ancresource ← Créer_ancre(, début, longueur)
    Insérer_lien(fcd.htm, ancresource, étiquette, tablecd[t,4])
  FINP
FINT
```


Le modèle relationnel de la BdC

Ce modèle a été implanté sous le SGBD Access et validé dans le cadre de l'application MEDTRAD (cf. images écran annexe D).

les relations

1. Concept(#numconcept, terme, définition, contexte, image, langue)
2. Relation_conceptuelle(#numconcept, #numconcept2, nomrc)
3. CD_simples(#numcds, nomcds)
4. CD_type_relation(#numcdtr, nomcdtr)
5. Comporte_cds(#numconcept, #numcds, valeurcds, typecds)
6. Comporte_cdtr(#numcdtr, numconcept, numcdtr, numenreg)
7. Description(#numdesc, texte, image)
8. Décrit_concept(#numconcept, #numdesc)
9. Hyperdoc(#nomdocht, titre, date,introduction, image)
10. Auteur(#numa, nom, prénom), adresse
11. Unité_informative(#numui, nomui, entete, pied, numui2, nomdocht, numconcept)
12. Lien(#numlien, début, longueur, étiquette, type, numui, numui2)
13. Ecrit(#numa, #nomdocht)

Définition des attributs des relations de la BdC

Relations	Attributs	Clé		Type	Lg	Explication
		primaire	étrangère			
Auteur	numa	O		Numérique		Code de l'auteur
	nom			Texte	50	nom de l'auteur
	prénom			Texte	50	prénom de l'auteur
	adresse			Texte	50	adresse électronique de l'auteur
CD_simple	numcds	O		Numérique		numéro du caractère définitoire simple
	nomcds			Texte	50	nom du caractère définitoire simple
CD_type_relation	numcdtr	O		Numérique		numéro d'une relation qui caractère définitoire complexe
	nomcdtr			Texte	20	nom d'une relation qui caractère définitoire complexe
Comporte_cds	numconcept		O	Numérique		
	numcds		O	Numérique		
	valeurcds			Texte	255	valeur du caractère définitoire simple
	typecds			Texte	15	type du caractère définitoire simple
Comporte_cdtr	numcdtr		O	Numérique		
	numconcept		O	Numérique		
	numcdtr		O	Numérique		
	numenreg		O	Numérique		
Concept	numconcept	O		Numérique		numéro du concept
	terme			Texte	50	désignation du concepts dans la langue de spécialité
	langue			Texte	50	langue dans laquelle est exprimé le concept
	définition			Memo		définition du concept
	contexte			Memo		contexte d'utilisation du concept
	image			Texte	11	image du concept
Description	numdesc	O		numérique		numéro de la description
	texte			Memo		Texte de description du concept
	image			texte	11	fichier image de description du concept
Décit_concept	numconcept		O			
	numdesc		O			
Ecrit	numa		O	Numérique		
	nomdocht		O	Texte	11	
Hyperdoc	nomdocht			Texte	11	nom d'un fichier hypertexte
	titre			Texte	255	titre du document
	date			Date		date de création du document
	introduction			Memo		Description de l'application qui est modélisée
	image			texte		Fichier image qui illustre l'application qui est modélisée
Lien	numlien	O		Numérique		numéro du lien
	début			Numérique		position du premier caractère de l'ancre
	longueur			Numérique		nombre de caractères que comporte l'ancre
	étiquette			Texte	50	ancre d'arrivée dans le document destination du lien
	type			Texte	10	information terminologique de type ITIT ayant le lien
	numui			Numérique		numéro de l'unité informative contenant l'ancre source
	numui2			Numérique		numéro de l'unité informative contenant l'ancre d'arrivée
Relation_conceptuelle	numconcept		O	Numérique		numéro du premier concept lié dans une relation
	numconcept2		O	Numérique		numéro du second concept lié dans une relation
	nomrc			Texte	50	nom de la relation qui lie deux concepts
Unité_informative	numui	O		Numérique		numéro de l'unité informative
	nomui			Texte	50	nom d'un fichier HTML contenant de l'unité informative
	entete			Texte	255	entete de l'unité informative
	pied			Texte	255	pied de page de l'unité informative
	numui2		O	Numérique		numéro de l'unité informative composante
	nomdocht		O	Texte	50	
	numconcept		O	Numérique		

La base de règles du système Medtrad

Règles de diagnostic

1. règle([symptome("selles trop fréquentes et trop liquides")], [diagnostique("diarrhée")])
2. règle([symptome("vertige"), symptome("maux de tête"), symptome("malaise général"), symptome("bourdonnements d'oreille")], [diagnostique("hypertension artérielle")])
3. règle([symptome("température à plus de trente sept degrés")], [diagnostique("fièvre")])
4. règle([symptome("urines fréquentes surtout la nuit"), symptome("infections cutanées inexplicables"), symptome("notion d'hérédité")], [diagnostique("diabète")])
5. règle([diagnostique("fièvre"), diagnostique("diarrhée"), symptome("sueur"), symptome("frissons"), symptome("courbatures"), symptome("maux de tête"), symptome("vomissements bilieuses"), [symptome("anémie"), symptome("ictère"), symptome("splénoragie")], [diagnostique("paludisme")])
6. règle([diagnostique("paludisme"), symptome("signes neurologiques à type de convulsion")], [diagnostique("paludisme stade de complication")])
7. règle([symptome("lésion avec ouverture de la peau")], [diagnostique("plaie")])
8. règle([symptome("sensation de brûlure du creux épigastrique"), symptome("douleur réveillée par la faim ou le tabac")], [diagnostique("ulcère d'estomac")])
9. règle([symptome("douleur au niveau de l'oreille")], [diagnostique("otite")])
10. règle([symptome("écoulement au niveau de l'oreille")], [diagnostique("otite supérieure")])
11. règle([symptome("douleur à caractère rapidement progressif"), symptome("douleur à caractère à caractère pulsatile"), symptome("douleur articulaire"), symptome("chaleur locale")], [diagnostique("arthrite")])

12. règle([symptome("diminution de la fréquence des selles"), symptome("douleur abdominale")], [diagnostique("constipation")])
13. règle([diagnostique("diarrhée"), symptome("ballonnement abdominale"), symptome("sensation de présence de gaz"), symptome("douleur abdominale"), symptome("vomissement")], [diagnostique("indigestion")])
14. règle([symptome("inspiration incomplète"), symptome("expiration difficile"), symptome("respiration prolongée"), symptome("respiration sifflante")], [diagnostique("asthme")])
15. règle([diagnostique("diarrhée"), symptome("sensation de faim anormale"), symptome("nausée surtout le matin"), symptome("perte de poids")], [diagnostique("parasitose intestinale")])
16. règle([symptome("gêne au niveau de l'appareil respiratoire"), symptome("effort à l'expiration")], [diagnostique("toux")])
17. règle([symptome("maux d'estomac")], [diagnostique("maux d'estomac")])
18. règle([symptome("douleur au niveau du bas ventre"), symptome("faux besoins"), symptome("selles sous forme de mœurs"), symptome("selles avec de la glaise ou du sang"), symptome("courbatures")], [diagnostique("dysentérie amibienne")])
19. règle([symptome("douleur pelvienne")], [diagnostique("dysmenorrhée")])
20. règle([symptome("douleur pelvienne"), symptome("pertes vaginales malodorantes")], [diagnostique("vaginose à trichomonas")])
21. règle([symptome("champignon de la peau")], [diagnostique("champignon de la peau")])
22. règle([symptome("digestion")], [diagnostique("digestion")])
23. règle([symptome("maux de dents")], [diagnostique("maux de dents")])
24. règle([symptome("douleurs articulaires saisonnières")], [diagnostique("rhumatisme")])

Règles thérapeutiques

25. règle([diagnostique("diarrhée")], [therapie("Acacia nilotica"), therapie("Guiera senegalensis"), therapie("Maytenus senegalensis"), therapie("Parkinsonia aculeata"), therapie("Psidium guajava")])

26. règle([diagnostique("hypertension artérielle")], therapie("Anacardium occidentale"),
[therapie("Catharanthus roseus")])
27. règle([diagnostique("fièvre")], [therapie("Azadirachta indica"), therapie("Cymbopogon
giganteus"), therapie("Moringa oleifera"), therapie("Parkinsonia aculeata")])
28. règle([diagnostique("diabète")], [therapie("Anacardium occidentale"),
therapie("Catharanthus roseus")])
29. règle([diagnostique("paludisme")], [therapie("Azadirachta indica")])
30. règle([diagnostique("paludisme stade de complication")], [therapie("Azadirachta in-
dica")])
31. règle([diagnostique("plaie")], [therapie("Borreria verticillata"), therapie("Centella asia-
tica"), therapie("Moringa oleifera"), therapie("Ocimum basilicum"), therapie("Psidium
guajava")])
32. règle([diagnostique("ulcère d'estomac")], [therapie("Bryophyllum pinnatum")])
33. règle([diagnostique("otite")], [therapie("Bryophyllum pinnatum")])
34. règle([diagnostique("otite supérieure")], [therapie("Bryophyllum pinnatum")])
35. règle([diagnostique("arthrite")], [therapie("Calotropis procera")])
36. règle([diagnostique("constipation")], [therapie("cassia occidentalis"), , therapie("Ricinus
communis"), therapie("Tamarindus indica")])
37. règle([diagnostique("indigestion")], [therapie("cassia occidentalis")])
38. règle([diagnostique("asthme")], [therapie("cassia occidentalis"), therapie("Euphorbia
hirta"), therapie("Moringa oleifera")])
39. règle([diagnostique("parasitose intestinale")], [therapie("Chenopodium ambrosioides")])
40. règle([diagnostique("toux")], [therapie("Combretum glutinosum"), therapie("Acacia se-
negal"), therapie("Guiera senegalensis")])
41. règle([diagnostique("maux d'estomac")], [therapie("Cymbopogon giganteus")])
42. règle([diagnostique("dysentérie amibienne")], [therapie("Euphorbia hirta"),
therapie("Parkinsonia aculeata"), therapie("Moringa oleifera")])
43. règle([diagnostique("dysménorrhée")], [therapie("Gossypium barbadense")])

- 44. règle([diagnostique("vaginite à trichomonas")], [therapie("Holarrhena floribunda")])
- 45. règle([diagnostique("champignon de la peau")], [therapie("Moringa oleifera")])
- 46. règle([diagnostique("digestion")], [therapie("Ocinum basilicum")])
- 47. règle([diagnostique("maux de dents")], [therapie("Pilostigma reticulatum")])
- 48. règle([diagnostique("rhumatisme")], [therapie("Piliostigma reticulatum")])

Images écrans de Medtrad sous ACCESS

Nous présentons ci-dessous les trois types de formulaires qui sont utilisés dans la version ACCESS du système MEDTRAD.

Formulaire d'accueil de MedTrad

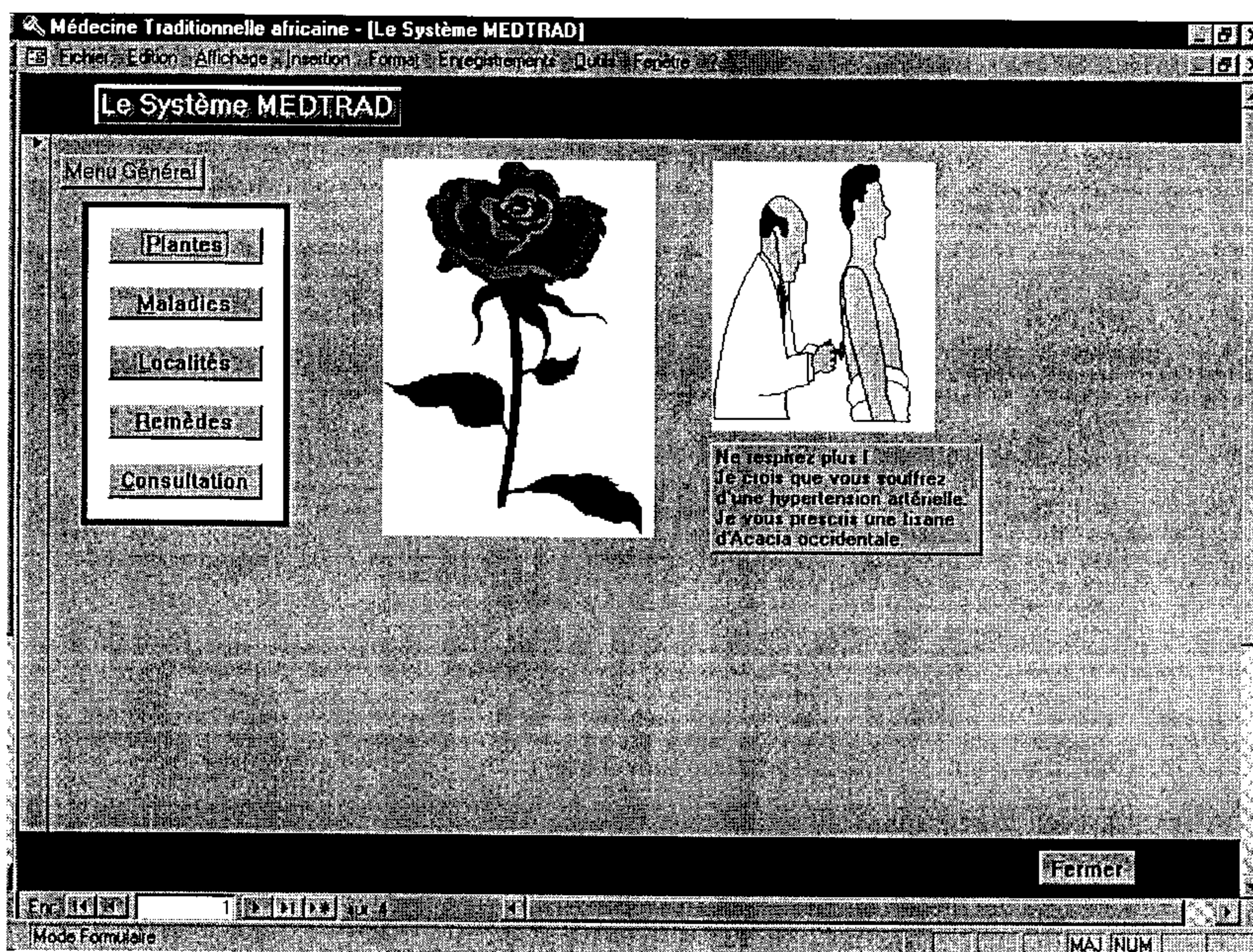


FIG. 19 – Menu général général .

Le choix d'un concept générique dans le menu lance l'ouverture du formulaire Fiches terminologiques de ce concept ; ce formulaire permet de prendre connaissance des informations terminologiques et des caractères définitoires d'un concept.

Formulaire fiches terminologiques

Le formulaire fiches terminologiques est composé de trois parties :

- la partie gauche contient les informations terminologique,
- la partie droite contient les caractères définitoires et
- la zone du bas contient les boutons de navigation.

Médecine Traditionnelle africaine - [Fiches terminologiques]

Numconcept: 4

Termo: Plante **image:** pl3.jpg

Langue: français

Définition: Les plantes se présentent sous différentes formes ou, si l'on préfère, sous des "aspects" et

Contexte: Les plantes dont il question ici font partie des végétaux possédant des fleurs. On les appelle

Synonyme

Caractères définitoires simples

nomcds	valeurcds
espèce	
famille	
genre	
hauteur maximale	

Ent: 1 sur 4

Caractères définitoires de type concept

concept2	terme
5	Fruit
6	Feuille
7	Tige
8	Racine

1 sur 5

Caractères définitoires de type relation

tr	nomcdctr
1	Composition chimique
2	Propriété thérapeutique

1 sur 2

Atteindre un concept

Entrer le numéro du concept

OK Annuler

1

Sous concepts

numconcept	terme

Atteindre Concept Atteindre CDTR Fermer

Ent: 1 sur 1 (Filtré)

Désignation du concept dans une langue de spécialiste

FILT MAJ NUM

FIG. 20 – Formulaire fiches terminologiques.

La figure 20 présente les fiches terminologiques du concept Plante.

L'activation du bouton **Atteindre concept** permet d'accéder directement aux fiches terminologiques de ses sous-concepts.

Le bouton **Atteindre CDTR** permet d'atteindre le formulaire permettant de visualiser les caractères définitoires de type relation d'un concept.

Le bouton **Fermer** ferme le formulaire actif afin de revenir au formulaire qui l'a ouvert.

Fiche terminologique du concept *Acacia senegal*

Médecine Traditionnelle africaine - [Fiches terminologiques]

Fichier Edition Affichage Insertion Format Enregistrement Outils Fenêtre ?

Fiches terminologiques Numconcept: 1

Terme: Acacia senegal **image:** pl2.jpg

Langue: latin

Définition: Petit arbre de 4 à 7 m de haut, à épine courtes et courbes, longues de 4 à 7 mm, réunies par trois à la base du pétiole. Les

Contexte: L'Acacia senegal fournit la gomme arabique officinale; à ce titre, il est assez connu. La récolte de cette

Synonyme

numconcept

Concept gén

numconcept

Concepts liés

nom

soigne

se trouve

Sous concepts

numconcept **terme**

Atteindre un CD de type relation ? X

Entrer le nom de la relation

Composition chimique

OK

Annuler

Caractères définatoires simples

nomcds	valeurcds
espèce	
famille	Mimosaceae
genre	
hauteur maximale	7

Err: 14 1 sur 6

Caractères définatoires de type concept

numconcept	terme
19	Feuille AS
20	Tige AS
21	Fruit AS
22	Fleur AS

1 sur 5

Caractères définatoires de type relation

nomcdctr
Composition chimique
Composition chimique
Composition chimique
Composition chimique

1 sur 6

Atteindre Concept **Atteindre CDTR** **Fermer**

Err: 14 1 sur 1 (Filtre)

Mode Formulaire

FILT NTIM

FIG. 21 – *Fiches terminologiques de Acacia senegal.*

Après avoir saisi la valeur 1 (numéro du concept *Acacia senegal*) dans la zone de saisie de la boîte de dialogue **Atteindre concept** et cliqué sur le bouton **Ok**, on obtient les fiches terminologiques du concept *Acacia senegal* (figure 21).

Formulaire informations terminologiques de type relation

Après avoir saisi la valeur Composition chimique (une table dont certains enregistrements sont caractères définitoires du concept Acacia senegal) dans la zone de saisie de la boîte de dialogue Atteindre CDTR et cliqué sur le bouton Ok, on accède à ces caractères définitoires (figure 22).

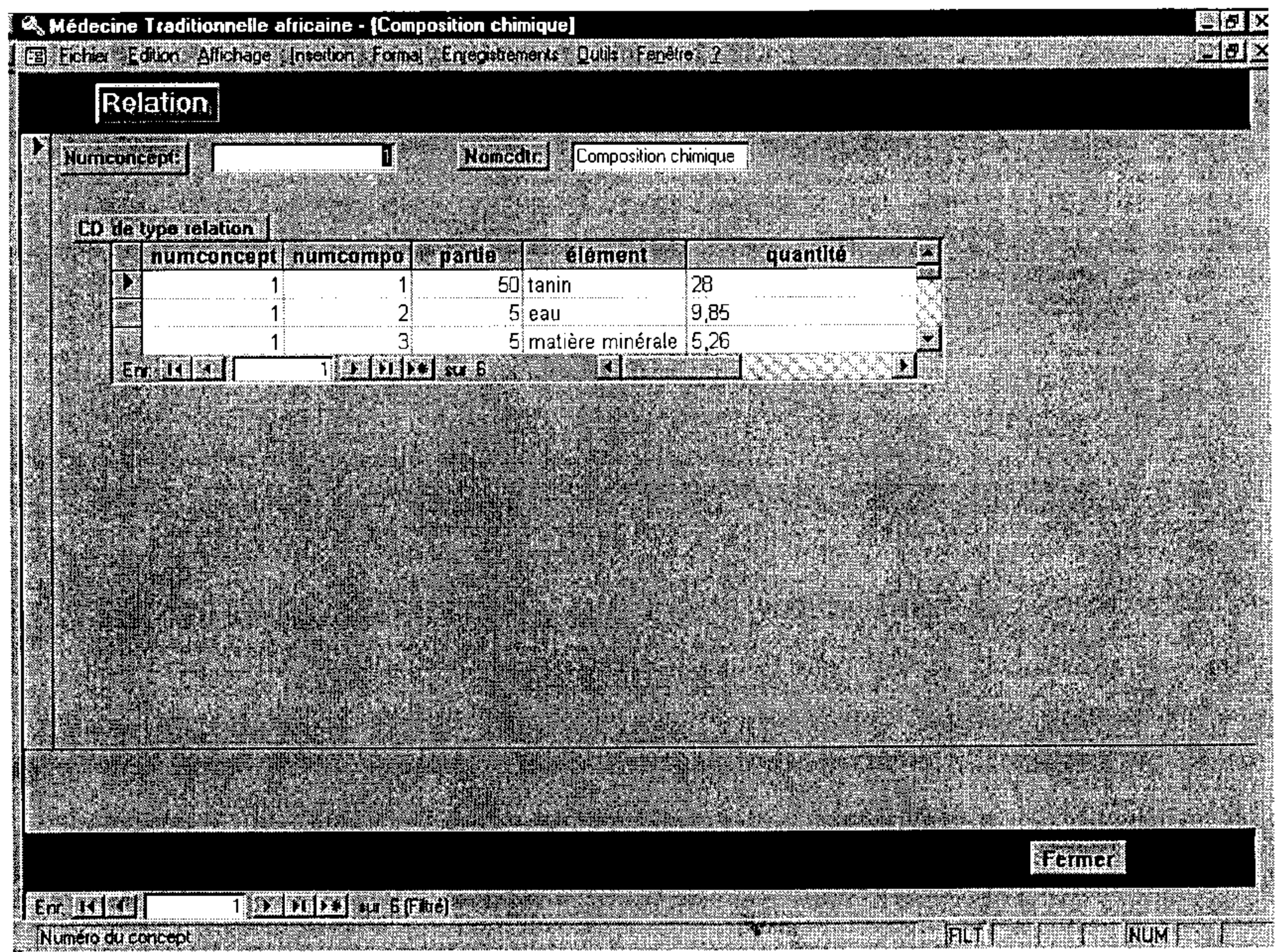


FIG. 22 – Caractères définitoires de type relation de Acacia senegal.

La figure 22 présente les enregistrement de la table Composition chimique dont certains enregistrements sont caractères définitoires de type relation de Acacia senegal.

Glossaires

Glossaire des sigles

BCG : Base de Connaissances Généralisées, 1

BdC : Base de concepts, 6, 10,

BHD : Base d'HyperDocument, 43, 44, 85, 90, 100

BNF : Backus Naur Form, 56, 81, 84, 161

CD : Caractère Définitoire, 55

CDC : Caractère Définitoire Complexe, 55, 56

CDS : Caractère Définitoire Simple, 55, 63, 73, 76

CDTC : Caractère Définitoire de type Concept, 56, 60, 63, 73, 76

CDTR : Caractère Définitoire de type Relation, 56, 63, 73, 76

CoDB-Web : Concept DataBase Web, 100, 102, 139

HTE : HyperText Expert, 95

HTML: HyperText Markup Language, 33, 36, 87, 89, 93, 105

MEDTRAD : MEDecine TRADitionnelle, 6, 122, 130, 138

IT : Information Terminologique, 72,

ITA : Information Terminologique, 54, 73

ITHT : Information Terminologique de type HyperTexte, 54, 73

ITRC : Information Terminologique de type Relation Conceptuelle, 55, 73

OD : Objet Documentaire, 67, 92, 107, 109

- ODCC : Objet Documentaire Concept Complexe, 109
- ODCE : Objet Documentaire Concept Elémentaire, 109
- UI : Unité Informative, 65, 66, 96
- UID : Unité Informative Destination, 68, 69, 94
- UICD : Unité Informative Caractère Définitoire, 73, 78
- UIIT : Unité Informative Information Terminologique, 72, 73, 78
- UIS : Unité Informative Source, 68, 69
- SGML: Standard Generalisation Markup Language, 33, 40
- SHT : Système HyperText, 101, 103

Glossaire des concepts peu courants

- Association : Relation sémantique qui existe entre deux entités dans le modèle Entité/ Association, 57, 58, 60.
- Hypérisation : c’est le travail qui permet de transformer un document textuel linéaire en réseau de nœuds hypertextes, 21, 87.
- Linéarisation : c’est l’opération inverse de l’hypérisation, elle consiste à séparer le contenu informatif d’un hypertexte de sa structure logique afin de les stocker dans les tables de la BdC, 120.
- Caractère définitoire : néologisme emprunté à [BON 94] afin d’insister sur le fait que les caractères sont plus que de simples attributs car ils sont porteurs d’une certaine sémantique permettant de définir ou de caractériser les concepts, 16, 55, 73.
- Relation : ou table dans les SGBDR, dont le schéma est le nom d’une entité suivi de la liste des attributs de l’entité, 24, 63, 159 .
- Relation conceptuelle : Relation sémantique qui existe entre deux concepts et qui est modélisée par l’entité Relation _ conceptuelle, 59, 60.

Syntaxe de la notation BNF

BNF est l'acronyme de Backus Naur Form; ce formalisme de description de la syntaxe du langage ALGOL a été défini par John Backus et Peter Naur. Les symboles BNF sont :

- `:=` signifie "est défini comme"
- `|` représente le OU logique
- `<>` encadrent les symboles non terminaux
- `" "` encadrent les symboles non terminaux
- `[ ]` encadrent les éléments optionnel
- `{ }` encadrent les éléments d'une liste