



HAL
open science

Stockage et exploitation de dossiers médicaux multimedia au moyen d'une base de données généralisée: **Projet Tigre**

Guadalupe Munoz-Baca

► **To cite this version:**

Guadalupe Munoz-Baca. Stockage et exploitation de dossiers médicaux multimedia au moyen d'une base de données généralisée: Projet Tigre. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1987. Français. NNT: . tel-00324082

HAL Id: tel-00324082

<https://theses.hal.science/tel-00324082>

Submitted on 24 Sep 2008

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

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

THESE

présentée par

MUNOZ-BACA Guadalupe

pour obtenir le titre de **DOCTEUR**

de **L'UNIVERSITE SCIENTIFIQUE TECHNOLOGIQUE ET
MEDICALE DE GRENOBLE**

(arrêté ministériel du 5 juillet 1984)

Spécialité : *INFORMATIQUE*

STOCKAGE ET EXPLOITATION

DE DOSSIERS MEDICAUX MULTIMEDIA

AU MOYEN D'UNE BASE DE DONNEES GENERALISEE.

Projet TIGRE

Date de soutenance : *1er Juillet 1987*

Composition du jury :

President	<i>Y.CHIARAMELLA</i>
Examineurs	<i>M.ADIBA</i>
	<i>J.DEMONGEOT</i>
	<i>M.ROUX</i>
	<i>M.SCHOLL</i>



UNIVERSITE SCIENTIFIQUE TECHNOLOGIQUE ET MEDICALE DE GRENOBLE

Président de l'Université :
M. TANCHE

Année Universitaire 1986 - 1987

MEMBRES DU CORPS ENSEIGNANT DE SCIENCES ET DE GEOGRAPHIE

PROFESSEURS DE 1ère Classe

ARNAUD Paul
ARVIEU ROBERT
AUBERT Guy
AURIAULT Jean-Louis
AYANT Yves
BARBIER Marie-Jeanne
BARBIER Jean-Claude
BARJON Robert
BARNOUD Fernand
BARRA Jean-René
BELORISKY Elie
BENZAKEN Claude
BERNARD Alain
BERTRANDIAS Françoise
BERTRANDIAS Jean-Paul
BILLET Jean
BOELHER Jean-Paul
BONNIER Jane Marie
BOUCHEZ Robert
BRAVARD Yves
CARLIER Georges
CAUQUIS Georges
CHIBON Pierre
COHEN ADDAD Jean-Pierre
COLIN DE VERDIERE Yves
CYROT Michel
DEBELMAS Jacques
DEGRANGE Charles
DELOBEL Claude
DEPORTES Charles
DESRE Pierre
DOLIQUE Jean-Michel
DOUCE Rolland
DUCROS Pierre
FONTAINE Jean-Marc
GAGNAIRE Didier
GERMAIN Jean-Pierre
GIRAUD Pierre
HICTER Pierre
IDELMAN Simon
JANIN Bernard
JOLY Jean-René
KAHANE André, détaché
KAHANE Josette
KRAKOWIAK Sacha
KUPKA Yvon
LAJZEROWICZ Jeanine
LAJZEROWICZ Joseph
LAURENT Pierre-Jean
DE LEIRIS Joel

Chimie Organique
Physique Nucléaire I.S.N.
Physique C.N.R.S
Mécanique
Physique Approfondie
Electrochimie
Physique Expérimentale CNRS
Physique Nucléaire ISN
Biochimie Macromoléculaire Végétale
Statistiques-Mathématiques Appliquées
Physique C.E.N.G- D.R.F.
Mathématiques Pures
Mathématiques Pures
Mathématiques Pures
Mathématiques Pures
Géographie
Mécanique
Chimie Générale
Physique Nucléaire ISN
Géographie
Biologie Végétale
Chimie Organique
Biologie Animale
Physique
Mathématiques Pures
Physique du Solide
Géologie Générale
Zoologie
Mathématiques Appliquées
Chimie Minérale
Electrochimie
Physique des Plasmas
Physiologie Végétale
Cristallographie
Mathématiques Pures
Chimie Physique
Mécanique,
Géologie
Chimie
Physiologie Animale
Géographie
Mathématiques Pures
Physique
Physique
Mathématiques Appliquées
Mathématiques Pures
Physique
Physique
Mathématiques Appliquées
Biologie

LLIBOUTRY Louis
LOISEAUX Jean-Marie
MACHE Régis
MAYNARD Roger
MICHEL Robert
OMONT Alain
OZENDA Paul
PAYAN Jean-Jacques
PEBAY-PEYROULA Jean-Claude
PERRIAUX Jacques
PERRIER Guy
PIERRARD Jean-Marie
PIERRE Jean-Louis
RASSAT André
RENARD Michel
RINAUDO Marguerite
ROSSI André
SAKAROVITCH Michel
SAXOD Raimard
SENGEL Philippe
SERGERAERT Francis
SOUCHIER Bernard
SOUTIF Michel
STUTZ Pierre
VALENTIN Jacques
VAN CUTSEM Bernard
VIALON Pierre

Géophysique
Sciences Nucléaires I.S.N.
Physiologie Végétale
Physique du Solide
Minéralogie et Pétrographie (Géologie)
Astrophysique
Botanique (Biologie Végétale)
Mathématiques Pures
Physique
Géologie
Géophysique
Mécanique
Chimie Organique
Chimie Systématique
Thermodynamique
Chimie CERMAV
Biologie
Mathématiques Appliquées
Biologie Animale
Biologie Animale
Mathématiques Pures
Biologie
Physique
Mécanique
Physique Nucléaire I.S.N.
Mathématiques Appliquées
Géologie

PROFESSEURS de 2^{ème} Classe

ADIBA Michel
ANTOINE Pierre
ARMAND Gilbert
BARET Paul
BECKER Pierre
BEGUIN Claude
BLANCHI J.Pierre
BOITET Christian
BORNAREL Jean
BRUANDET J.François
BRUN Gilbert
CASTAING Bernard
CERFF Rudiger
CHARDON Michel
CHIARAMELLA Yves
COURT Jean
DEMAILLY Jean-Pierre
DENEUVILLE Alain
DEPASSEL Roger
DERRIEN Jacques
DUFREYNOY Alain
GASPARD François
GAUTRON René
GENIES Eugène
GIDON Maurice
GIGNOUX Claude
GILLARD Roland
GIORNI Alain
GUIGO Maryse
GUMUCHAIN Hervé
GUITTON Jacques
HACQUES Gérard

Mathématiques Pures
Géologie
Géographie
Chimie
Physique
Chimie Organique
STAPS
Mathématiques Appliquées
Physique
Physique
Biologie
Physique
Biologie
Géographie
Mathématiques Appliquées
Chimie
Mathématiques Pures
Physique
Mécanique des Fluides
Physique
Mathématiques Pures
Physique
Chimie
Chimie
Géologie
Sciences Nucléaires
Mathématiques Pures
Sciences Nucléaires
Géographie
Géographie
Chimie
Mathématiques Appliquées

HERBIN Jacky
HERAULT Jeanny
JARDON Pierre
JOSELEAU Jean-Paul
KERCKHOVE Claude
LEBRETON Alain
LONGEQUEUE Nicole
LUCAS Robert
LUNA Domingo
MANDARON Paul
MARTINEZ Francis
MASCLE Georges
NEMOZ Alain
OUDET Bruno
PELMONT Jean
PERRIN Claude
PFISTER Jean-Claude
PIBOULE Michel
RAYNAUD Hervé
RIEDIMANN Christine
ROBERT Gilles
ROBERT Jean-Bernard
SARROT-REYNAULD Jean
SAYETAT Françoise
SERVE Denis
STOECKEL Frédéric
SOUTIF Jeanne
SCHOLL Pierre-Claude
SUBRA Robert
VALLADE Marcel
VIDAL Michel
VIVIAN Robert
VOTTERO Philippe

Géographie
Physique
Chimie
Biochimie
Géologie
Mathématiques Appliquées
Sciences Nucléaires I.S.N.
Physique
Mathématiques Pures
Biologie
Mathématiques Appliquées
Géologie
Thermodynamique CNRS - CRTBT
Mathématiques Appliquées
Biochimie
Sciences Nucléaires I.S.N.
Physique du Solide
Géologie
Mathématiques Appliquées
Mathématiques Pures
Mathématiques Pures
Chimie Physique
Géologie
Physique
Chimie
Physique
Physique
Mathématiques Appliquées
Chimie
Physique
Chimie Organique
Géographie
Chimie

MEMBRES DU CORPS ENSEIGNANT DE L' IUT 1

PROFESSEURS de 1^{ère} Classe

BUISSON Roger
DODU Jacques
NEGRE Robert

Physique IUT 1
Mécanique Appliquée IUT 1
Génie Civil IUT 1

PROFESSEURS de 2^{ème} classe

BOUTHINON Michel
CHAMBON René
CHEHIKIAN Alain
CHENAVAS Jean
CHOUTEAU Gérard
CONTE René
GOSSE Jean-Pierre
GROS Yves
KUHNS Gérard, (Détaché)
MAZUER Jean
MICHOUILLER Jean
MONLLOR Christian
NOUGARET Marcel
PEFFEN René
PERRARD Jacques
PERRAUD Robert
TERRIEZ Jean-Michel
TOUZAIN Philippe
VINCENDON Marc

EEA. IUT 1
Génie Mécanique IUT 1
EEA. IUT 1
Physique IUT 1
Physique IUT 1
Physique IUT 1
EEA. IUT 1
Physique IUT 1
Physique IUT 1
Physique IUT 1
EEA. IUT 1
Automatique IUT 1
Métallurgie IUT 1
EEA. IUT 1
Chimie IUT 1
Génie Mécanique IUT 1
Chimie IUT 1
Chimie IUT 1

MEMBRES DU CORPS ENSEIGNANT DE MEDECINE

PROFESSEURS CLASSE EXEPTIONNELLE ET 1ère CLASSE

AMBLARD Pierre	Dermatologie	C.H.R.G.
AMBROISE-THOMAS Pierre	Parasitologie	C.H.R.G.
BEAUDOING André	Pédiatrie-Puericulture	C.H.R.G.
BEZEZ Henri	Orthopédie-Traumatologie	Hopital SUD
BONNET Jean-Louis	Ophthalmologie	C.H.R.G.
BOUCHET Yves	Anatomie	Faculté La Merci
	Chirurgie Générale et Digestive	C.H.R.G.
BUTEL Jean	Orthopédie-Traumatologie	C.H.R.G.
CHAMPETIER Jean	Anatomie-Topographique et Appliquée	C.H.R.G.
	O.R.L.	C.H.R.G.
CHARACHON Robert	Anatomie-Pathologique	C.H.R.G.
COUDERC Pierre	Pneumophtisiologique	C.H.R.G.
DELORMAS Pierre	Cardiologie	C.H.R.G.
DENIS Bernard	Pharmacologie	Faculté La Merci
GAVEND Michel	Hématologie	C.H.R.G.
HOLLARD Daniel	Chirurgie Thoracique et Cardiovasculaire	C.H.R.G.
LATREILLE René	Bactériologie-Virologie	C.H.R.G.
	Gynécologie et Qbstétrique	C.H.R.G.
LE NOC Pierre	Médecine du Travail	C.H.R.G.
MALINAS Yves	Clinique Médicale et Maladies Infectieuses	C.H.R.G.
MALLION Jean-Michel	Histologie	Faculté La Merci
MICOUD Max	Pneumologie	C.H.R.G.
	Neurologie	C.H.R.G.
MOURIQUAND Claude	Hépatogastro-Entérologie	C.H.R.G.
PARAMELLE Bernard	Neurochirurgie	C.H.R.G.
PERRET Jean	Clinique Chirurgicale	C.H.R.G.
RACHAIL Michel	Anesthésiologie	C.H.R.G.
DE ROUGEMONT Jacques	Physiologie	Faculté La Merci
SARRAZIN Roger	Biophysique	Faculté La Merci
STIEGLITZ Paul	Biochimie	Faculté La Merci
TANCHE Maurice		
VERAIN André		
VIGNAIS Pierre		

PROFESSEURS 2ème CLASSE

BACHELOT Yvan	Endocrinologie	C.H.R.G.
BARGE Michel	Neurochirurgie	C.H.R.G.
BENABID Alim Louis	Biophysique	Faculté La Merci
BENSA Jean-Claude	Immunologie	Hopital Sud
BERNARD Pierre	Gynécologie-Obstétrique	C.H.R.G.
BESSARD Germain	Pharmacologie	ABIDJAN
BOLLA Michel	Radiothérapie	C.H.R.G.
BOST Michel	Pédiatrie	C.H.R.G.
BOUCHARLAT Jacques	Psychiatrie Adultes	Hopital Sud
BRAMBILLA Christian	Pneumologie	C.H.R.G.
CHAMBAZ Edmond	Biochimie	C.H.R.G.
CHIROSSSEL Jean-Paul	Anatomie-Neurochirurgie	C.H.R.G.
COLOMB Maurice	Immunologie	Hopital Sud
COMET Michel	Biophysique	Faculté La Merci
CONTAMIN Charles	Chirurgie Thoracique et Cardiovasculaire	C.H.R.G.
	Néphrologie	C.H.R.G.
CORDONNIER Daniel	Radiologie	C.H.R.G.
COULOMB Max	Radiologie	C.H.R.G.
CROUZET Guy	Médecine Interne et Toxicologie	C.H.R.G.
DEBRU Jean-Luc	Biostatistiques et Informatique Médicale	Faculté La Merci
DEMONGEOT Jacques		

DUPRE Alain	Chirurgie Générale	C.H.R.G.
DYON Jean-François	Chirurgie Infantile	C.H.R.G.
ETERRADOSSI Jacqueline	Physiologie	Faculté La Merci
FAURE Claude	Anatomie et Organogénèse	C.H.R.G.
FAURE Gilbert	Urologie	C.H.R.G.
FOURNET Jacques	Hépatogastro-Entérologie	C.H.R.G.
FRANCO Alain	Médecine Interne	C.H.R.G.
GIRARDET Pierre	Anesthésiologie	C.H.R.G.
GUIDICELLI Henri	Chirurgie Générale et Vasculaire	C.H.R.G.
GUIGNIER Michel	Thérapeutique et Réanimation	C.H.R.G.
	Médicale	C.H.R.G.
HADJIAN Arthur	Biochimie	Faculté La Merci
HALIMI Serge	Endocrinologie et Maladies	
	Métaboliques	C.H.R.G.
HOSTEIN Jean	Hépatogastro-Entérologie	C.H.R.G.
HUGONOT Robert	Médecine Interne	C.H.R.G.
JALBERT Pierre	Histologie-Cytogénétique	C.H.R.G.
JUNIEN-LAVILLAULOY Claude	O.R.L.	C.H.R.G.
KOLODIE Lucien	Hématologie Biologique	C.H.R.G.
LETOUBLON Christian	Chirurgie Générale	C.H.R.G.
MACHECOURT Jacques	Cardiologie et Maladies	
	Vasculaires	C.H.R.G.
MAGNIN Robert	Hygiène	C.H.R.G.
MASSOT Christian	Médecine Interne	C.H.R.G.
MOUILLON Michel	Ophthalmologie	C.H.R.G.
PELLAT Jacques	Neurologie	C.H.R.G.
PHELIP Xavier	Rhumatologie	C.H.R.G.
RACINET Claude	Gynécologie	C.H.R.G.
RAMBAUD Pierre	Pédiatrie	C.H.R.G.
RAPHAEL Bernard	Stomatologie	C.H.R.G.
SCHAERER René	Cancérologie	C.H.R.G.
SEIGNEURIN Jean-Marie	Bactériologie-Virologie	Faculté La Merci
SELE Bernard	Cytogénétique	Faculté La Merci
SOTTO Jean-Jacques	Hématologie	C.H.R.G.
STOEBNER Pierre	Anatomie Pathologique	C.H.R.G.
VROUSOS Constantin	Radiothérapie	C.H.R.G.

MEMBRES DU CORPS ENSEIGNANT PHARMACIE

AGNIUS-DELORD Claudine	Physique	Faculté La Tronche
ALARY Josette	Chimie Analytique	Faculté La Tronche
BERIEL Hélène	Physiologie et Pharmacologie	Faculté La Tronche
BOUCHERLE André	Chimie et Toxicologie	Faculté Meylan
CUSSAC Max	Chimie Thérapeutique	Faculté La Tronche
DEMENGE Pierre	Pharmacodynamie	Faculté La Tronche
JEANNIN Charles	Pharmacie Galénique	Faculté Meylan
LATURAZE Jean	Biochimie	Faculté La Tronche
LUU DUC Cuong	Chimie Générale	Faculté La Tronche
MARIOTTE Anne-Marie	Pharmacognosie	Faculté La Tronche
MARZIN Daniel	Toxicologie	Faculté Meylan
RENAUDET Jacqueline	Bactériologie	Faculté La Tronche
ROCHAT Jacques	Hygiène et Hydrologie	Faculté La Tronche
SEIGLE-MURANDI Françoise	Botanique et Cryptogamie	Faculté Meylan
VERAIN Alice	Pharmacie Galénique	Faculté Meylan



Je tiens particulièrement à remercier :

Monsieur Yves CHIARAMELLA, Professeur à l'Université Scientifique, Technologique et Médicale de Grenoble, de m'avoir fait l'honneur de présider le jury de cette thèse.

Monsieur Michel ADIBA, Professeur à l'Université Scientifique, Technologique et Médicale de Grenoble et Directeur de cette thèse, pour m'avoir accueilli au sein de son équipe de recherche. J'aurais beaucoup à dire pour lui exprimer m'a plus sincère gratitude pour sa totale disponibilité à mon égard et pour tous les conseils et encouragements qu'il m'a donné tout au long de ce travail. Qu'il trouve ici, si cela est possible, l'expression de mes plus profonds remerciements.

Monsieur Jack DEMONGEOT, Professeur à l'Université Scientifique, Technologique et Médicale de Grenoble, Monsieur Michel ROUX, Professeur à l'Université d'Aix-Marseille II et Monsieur Michel SCHOLL, Directeur de Recherche INRIA Paris, d'avoir voulu apporter leur jugement sur ce travail et, participer au jury.

Monsieur Philippe CINQUIN, chercheur au Centre de Biostatistiques et Informatique médicale de Grenoble et, Monsieur M. COULOMB responsable du service de Radiologie de l'hôpital Les Sablons à Grenoble, pour leur patience et leur disponibilité pour m'initier au domaine médicale qui m'était complètement nouveau.

Tous les membres de l'équipe TIGRE, pour l'amitié qu'ils m'ont témoignée. Je voudrais remercier très spécialement mes collègues et amis Francisca ANTUNES, Ofelia CERVANTES, Esperanza PEDRAZA et Rodrigo SCIOVILLE pour leur soutien amical durant la réalisation de ce travail.

Je tiens à remercier toutes les personnes qui ont bien voulu effectuer le dur travail de correction de cette thèse, Brigitte LIESSE, Christian LENNE, qu'ils soient vivement remerciés.

Je remercie également Monsieur Daniel IGLESIAS et le service de reprographie pour le soin apporté au tirage de cette thèse.

Je remercie enfin le "Consejo Nacional de Ciencia y Tecnología" (CONACYT), pour son soutien financier.



A mis Padres



A Rolf
Con todo mi amor



Résumé

Notre travail présente les besoins d'une application médicale utilisant des données multimédia. C'est une application pilote pour le modèle et le SGBD TIGRE. L'application concerne le stockage et l'exploitation de dossiers médicaux.

Une modélisation de l'application est réalisé avec le modèle TIGRE; elle montre les objets du dossier radiologique et les liens qui existent entre eux. Une deuxième modélisation et une implémentation du schéma obtenu sont réalisées avec un SGBD relationnel. Une couche multimédia entre l'application et le SGBD relationnel a été construite. Elle permet (grâce au langage SQLMED créé) la manipulation homogène de données classiques et de données multimédia.

Mots-clés : bases de données généralisées, modèle de données, données multimédia, dossier médicaux.



TABLE DES MATIERES

TABLE DES MATIERES	1
INTRODUCTION	9
1. L'INFORMATION A L'HOPITAL.	15
1.1. Les systèmes d'information hospitaliers (S.I.H.).	15
1.2. Systèmes d'information et bases de données médicales.	20
1.2.1. Technologie des systèmes de gestion de bases de données (SGBD).	22
1.2.1.1. Définition des base de données.	22
1.2.1.2. Description des données, schéma interne, schéma externe, et schéma conceptuel.	22
1.2.1.3. Modèles de données.	23
1.2.2. Applications médicales utilisant les techniques de SGBD.	27
1.2.2.1. Le médecin indépendant.	28
1.2.2.2. Le groupe médical indépendant.	28
1.2.2.3. Le médecin spécialiste.	29
1.2.2.4. La recherche médicale.	29
1.2.2.5. Les hôpitaux et les cliniques.	29
1.2.3. Quelques exemples de S.I.H. développés en France.	30
1.2.3.1. Le système SORGHO.	30
1.2.3.2. Le système DIMI.	30
1.2.3.3. Le système SYMPHONIE.	31
1.3. Le service de radiologie.	31
1.3.1. L'image radiologique.	35
1.3.2. Les comptes rendus.	40

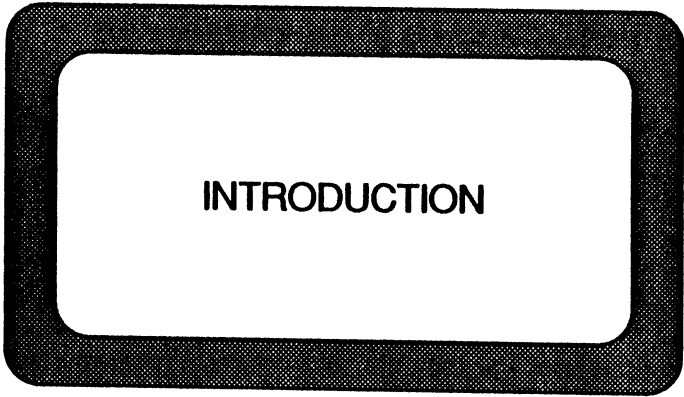
2.	ETAT DE L'ART DES SYSTEMES DE GESTION D'IMAGES RADIOLOGIQUES ET DES SYSTEMES DE GESTION DE COMPTES RENDUS.	45
2.1.	Etat de l'art des travaux réalisés sur les systèmes de gestion d'images.	45
2.1.1.	Les applications médicales.	47
2.1.1.1.	Le système IQS.	49
2.1.1.2.	Le système IDBMS.	50
2.1.1.3.	Le système GRAIN.	51
2.1.1.4.	Le système EL.	52
2.1.1.5.	Le système SIRENE.	53
2.1.1.6.	Un système de gestion d'images radiologiques.	57
2.1.2.	Les applications géographiques.	58
2.1.2.1.	Le système IBIS.	59
2.1.2.2.	Le système EORID.	59
2.1.2.3.	Le système IMAID.	61
2.1.2.4.	Le système EIDES.	62
2.1.3.	Les applications plus générales.	63
2.1.3.1.	Le système MMS.	64
2.1.3.2.	Le système MINOS.	66
2.1.3.3.	Le système IDMS.	69
2.1.3.4.	Le système ELF.	70
2.1.3.5.	Le système ADM.	71
2.2.	Etat de l'art des travaux réalisés sur les systèmes de gestion de comptes rendus.	73
2.2.1.	Le système REMEDE.	74
2.2.2.	Le langage LRFM.	77
2.2.3.	Le langage INTERMED.	77
2.2.4.	Un système de gestion d'examens de tomographie destinés à l'enseignement.	78
2.2.5.	Le système CLIP.	79
2.2.6.	Le système ROM.	83
2.2.7.	Le système RAPORT.	84
2.2.8.	Le système JOHNS HOPKINS.	86
2.2.9.	Le système AURA.	87
2.3.	Discussion.	88
2.4.	Notre système.	89

3.	LES BASES DE DONNEES GENERALISEES.	101
3.1.	Le modèle de données TIGRE.	104
3.2.	Les types de données du modèle TIGRE.	105
3.2.1.	Les types de base.	105
3.2.1.1.	Les types simples.	105
3.2.1.2.	Les types restreints.	106
3.2.2.	Les types construits	106
3.2.2.1.	Le constructeur "renommage".	106
3.2.2.2.	Le constructeur "enregistrement".	106
3.2.2.3.	Le constructeur "tableau".	107
3.2.2.4.	Le constructeur "document".	107
3.2.3.	Les types-classe.	109
3.2.3.1.	Le type-entité.	109
3.2.3.2.	Le type-association.	110
3.2.4.	Les types-temps.	111
3.2.5.	Opérateurs de types.	112
3.3.	Le modèle TIGRE et l'application médicale.	113
3.4.	Modélisation des objets du dossier radiologique.	115
	Description des entités.	115
3.4.1.	L'entité PATIENT.	115
3.4.2.	L'entité IMAGES.	116
3.4.3.	L'entité SCANNER.	117
3.4.4.	L'entité IRMN.	117
3.4.5.	L'entité IMAGEINDEP.	118
3.4.6.	L'entité IMAGEREF.	118
3.4.7.	L'entité COMPTERENDU.	119
3.4.8.	L'entité MEDECIN.	121
	Description des associations.	121
3.4.9.	L'association EXAMEN.	122
3.4.10.	L'association ENTRAITEMENT.	122
3.4.11.	L'association OBSERVATION.	123
3.4.12.	L'association DERIVATION.	123
3.4.13.	L'entité agrégée DOSSIER.	123
	Description des attributs définis sur un type document.	124
3.4.14.	Le document SECTION.	124
3.4.15.	Le document CLICHES.	124
3.4.16.	Diagramme du schéma conceptuel.	125

3.4.17.	Autres schémas conceptuels possibles.	128
3.5.	Le langage LAMBDA et les exemples de requêtes de l'application médicale.	131
3.5.1.	Les énoncés d'interrogation.	131
3.5.1.1.	Expression de fonctions.	134
3.5.1.2.	Expression des ensembles.	137
3.5.2.	Stockage et recherche des documents.	137
3.5.3.	Enoncés de mise à jour.	140
3.6.	Problèmes de mise en œuvre.	141
4.	MODELISATION DE L'APPLICATION MEDICALE AVEC UN SGBD RELATIONNEL.	145
4.1.	Schéma relationnel de l'application.	145
4.1.1.	Description de chacune des relations.	146
4.2.	Technique de signature et couplage avec un SGBD relationnel. ..	151
4.2.1.	Diverses méthodes d'accès aux données textuelles.	151
4.2.2.	Une méthode d'accès basée sur la technique de signature ...	155
4.2.2.1.	Méthode de signature par mot.	156
4.2.2.2.	Méthode de signature par codage surimposé.	157
4.2.3.	Couplage de la méthode de fichier de signatures avec le SGBD relationnel.	159
4.2.4.	Prédicats de recherche par contenu.	160
4.2.4.1.	Prédicats disjonctifs (OR).	160
4.2.4.2.	Prédicats conjonctifs (AND).	161
4.2.4.3.	Prédicats adjacents (ADJ).	162
4.2.4.4.	Prédicats composés.	163
4.3.	Fonctions et architecture du prototype réalisé.	164
4.3.1.	Interrogation de la base de données.	172
4.3.2.	Mise à jour de la base de données.	174
4.3.2.1.	Insertion de données.	174
4.3.2.2.	Modification de données.	174
4.3.2.3.	Suppression de données.	175
4.4.	Exemple de requêtes.	175
4.5.	Réalisation.	179
4.5.1.	Le module CODIF.	184
4.5.2.	Le module CLICHES.	189
4.5.3.	Le module QUERYS.	189

5. CONCLUSION	195
6. BIBLIOGRAPHIE	201
7. ANNEXE 1.	217
LISTE DES RAPPORTS TIGRE	222







INTRODUCTION

Après l'avènement du modèle relationnel au début des années 70, de nombreux Systèmes de Gestion de Bases de Données (SGBD) ont été développés et largement commercialisés. Ces systèmes offrent une vision tabulaire et ensembliste des données, des langages simples à utiliser et des facilités d'évolution dynamique des bases de données. Ils sont capables de gérer de grandes quantités de données élémentaires, qui sont en général formatées et de petite taille. Cependant, de nouvelles applications informatiques ont besoin de stocker et de manipuler des données plus complexes en taille et en structure et avec une sémantique propre. Ces données sont normalement appelées données généralisées ou multimédia. Ce sont par exemple, les textes, les images, les graphiques, la voix. Les SGBD classiques offrent des outils très pauvres pour manipuler ces types de données et, du fait de leur taille importante, ces données posent de nouveaux problèmes de stockage et d'accès.

Ces types d'applications ont permis le développement de nouveaux systèmes de gestion de bases de données offrant des fonctionnalités nouvelles pour manipuler et stocker les données généralisées. Ce sont des modèles plus riches qui prennent en compte les aspects sémantiques de l'application et offrent des outils de manipulation plus puissants en raison de la complexité des données. Dans un premier temps, ils nécessitent une étude et une formalisation des exigences de ces applications et, dans un second temps, ils permettent la conception et l'expérimentation de nouveaux types de logiciels de gestion de données. Cette nouvelle génération de modèles de données est appelée "modèle sémantique". Son but principal lors de la conception d'un modèle de données est d'améliorer son pouvoir d'expression et fournir des concepts plus riches et plus précis que ceux des modèles classiques.

La modélisation d'une application peut se faire d'une manière plus naturelle si le modèle utilisé offre la possibilité de définir des structures qui correspondent directement aux entités du monde réel. Ceci correspond à l'approche orientée objet, dont la structure de base est l'objet et pour lesquels on peut décrire son comportement, les opérateurs applicables à chaque type d'objet ainsi que des contraintes d'intégrité à respecter. L'idée principale est donc d'adapter le modèle au monde réel et non le monde réel au modèle.

Parmi toutes les applications pouvant bénéficier de ces nouveaux SGBD, nous pouvons mentionner les applications médicales, où l'on manipule des données multimédia telles que les images radiologiques, les résultats des examens de

laboratoire, les comptes rendus, les historiques des patients, etc. La plupart de ces données ne peuvent pas être gérées par les SGBD classiques, les images par exemple ont une nature complexe et volumineuse, les comptes rendus ont une nature textuelle dont les SGBD classiques offrent des outils très pauvres de manipulation et de stockage. Par exemple, il est possible de stocker un objet volumineux en utilisant un SGBD relationnel, si on le décompose en pavés et, si on stocke chaque pavé dans un constituant de type "long" (qui offre jusqu'à 64koctets de mémoire). Néanmoins, à part la récupération de l'ensemble des données contenus dans le constituant, le SGBD n'offre aucun moyen pour accéder au contenu des champs. De plus, les SGBD classiques offrent une vision des données qui dans la plus part des cas ne convient pas aux données complexes ou volumineuses (ex. information sous forme de tableaux pour le modèle relationnel).

C'est dans cette ligne de recherche de nouvelle génération de modèles que le projet TIGRE (Traitement d'Informations Généralisées et REparties) a été lancé à Grenoble. Il est mené conjointement par le Laboratoire de Génie Informatique de l'IMAG et le centre de recherche BULL. Sa finalité est la manipulation des bases de données généralisées dans un contexte bureautique. Le modèle TIGRE appartient à la classe des modèles sémantiques c'est-à-dire qu'il prend en compte les propriétés de la structure des objets ainsi que son comportement par rapport aux autres objets.

Le travail présenté ici se place dans le cadre du projet TIGRE et porte sur deux aspects fondamentaux :

- le premier consiste à réaliser une application pilote pour le modèle et le SGBD TIGRE, afin de pouvoir tester dans une situation réelle, les facilités que ce SGBD offre pour modéliser le monde réel,
- le deuxième consiste à offrir à cette application les outils informatiques qui répondent aux besoins de stockage et de manipulation des données multimédia.

Notre travail concerne une application réelle proposée par le service de radiologie d'un hôpital grenoblois. Dans le service de radiologie, on manipule tous les jours un grand nombre de données classiques et de données généralisées dont des images radiologiques et des comptes rendus associés à ces images. On voudrait donc disposer d'un système qui puisse gérer d'une manière intégrée et homogène l'ensemble des données manipulées. Les SGBD traditionnels n'étant pas aptes à gérer ces types de données, il s'avère indispensable d'utiliser de nouvelles générations de SGBD.

L'organisation de cette thèse est la suivante :

Dans le chapitre 1, nous présentons les caractéristiques de l'information dans le milieu médical, sa circulation dans un hôpital, et son intégration dans un système d'information hospitalier. Nous faisons aussi mention des avantages que peuvent tirer ces systèmes des techniques de base de données. A la fin du chapitre, nous présentons les activités du service de radiologie de notre application, et nous faisons une description des objets à gérer.

Dans le chapitre 2, nous présentons une synthèse des travaux concernant les systèmes de gestion d'images radiologiques et de comptes rendus médicaux. Nous faisons aussi une présentation générale du système que nous proposons pour gérer l'information du service de radiologie.

Dans le chapitre 3, nous présentons la modélisation de l'application en utilisant le modèle TIGRE et nous faisons une description générale du modèle. Bien que ce modèle offre des outils intéressants pour la modélisation de l'application, la réalisation du système n'a pas été faite avec le SGBD TIGRE qui était en cours de réalisation.

Finalement, le chapitre 4 fait une description de la réalisation de l'application en utilisant un modèle relationnel et une méthode d'accès aux données textuelles. Une description de l'implémentation réalisée y est donnée.



CHAPITRE I

L'INFORMATION A L'HOPITAL



CHAPITRE I

1. L'INFORMATION A L'HOPITAL.

1.1. Les systèmes d'information hospitaliers (S.I.H.).

L'hôpital est considéré comme une structure complexe constituée par un ensemble d'éléments que nous pouvons regrouper en trois types fondamentaux: les malades, le personnel, et le matériel. Dans un hôpital, on est confronté avec des personnes et des départements, qui sont directement ou indirectement liés. Dans cette relation il existe un ensemble d'information qui, la plupart du temps, est utilisé pour réaliser des fonctions ayant pour but d'améliorer la santé des malades. Parallèlement, cette information est utilisée dans la recherche ou bien dans l'enseignement (lorsque l'hôpital a une vocation universitaire).

Trois types d'information interagissent tous les jours dans le fonctionnement de l'hôpital: des informations de type administratif (nom du malade, âge, état civil, etc), des connaissances médicales (connaissances personnelles, des références humaines ou livresques), et des observations médicales au sens large (de l'observation clinique au protocole thérapeutique, en passant par les examens paracliniques) [CINQ 84]

Ces trois types d'information diffèrent non seulement par leur nature ou leur origine, mais aussi par les aspects qui les caractérisent, par exemple: les informations administratives sont fortement structurées, c'est-à-dire qu'elles peuvent être enregistrées selon un format prédéfini et identique pour tous les individus. D'autres informations ne sont que très peu, voir pas du tout structurées, c'est le cas d'observations cliniques rédigées en texte libre, selon un plan adapté à la personnalité du rédacteur et à la pathologie du malade.

Les examens paracliniques sont eux-mêmes générateurs d'un autre type d'information qui font aussi partie intégrante des informations utilisées à l'hôpital; parmi eux, nous pouvons citer les images radiologiques (analogiques et numériques), les examens de laboratoire, les électrocardiogrammes, etc.

L'information à l'hôpital joue toujours deux rôles, l'un concernant la "santé individuelle" (amélioration de la santé du malade), et l'autre concernant la "santé publique" (études épidémiologiques, recherche scientifique, enseignement). La gestion de l'information passe par des étapes de création, transmission, stockage et traitement [CDCG 86]. Les difficultés observées dans ces étapes portent sur leur déroulement et sur leur contrôle par la collectivité. Une description non exhaustive de chacune de ces étapes est donnée dans ce qui suit:

* La création de l'information peut être liée à toute personne concernée par l'hôpital. Les problèmes soulevés par cette étape concernent d'une part l'habilitation de l'agent à procéder effectivement à la création d'une information, ceci est illustré par le degré de confiance accordé à l'information par les personnes qui vont s'en servir. D'autre part, apparaît la nécessité de contrôler la création d'une information, étant donné que des examens, difficiles à obtenir, sont souvent demandés sans raisons bien justifiées. L'étape de la création devrait donc être contrôlée par des connaissances médicales, des connaissances financières et des données médicales et administratives, afin d'optimiser les ressources de tout genre [CINQ 84].

* La transmission de l'information s'avère indispensable pour sa communication, étant donné que le créateur d'une information est rarement le seul à l'utiliser. Cette étape devient plus sensible au fur et au mesure que le volume des informations grossit. Parmi les problèmes que l'on rencontre ici, nous pouvons mentionner les retards de la transmission, la transmission d'informations erronées, les difficultés à reconnaître l'existence ou la localisation d'une information, la fiabilité du support d'une information, etc.

* Le stockage de l'information touche à nouveau les deux aspects "santé individuelle" et "santé publique". Au niveau de la "santé individuelle", la problématique principale se trouve dans la dispersion physique et logique des supports de l'information concernant le malade, car chaque séjour d'un malade, dans une unité de soins différente, donne lieu à l'ouverture d'un nouveau dossier. Ensuite se pose le problème de sélection des informations à conserver, ainsi que la structure et la sémantique des informations à enregistrer. Au niveau de la "santé publique", la question de la sélection des informations à conserver est primordiale. Or cette étape hérite de tous les problèmes concernant la "santé individuelle", étant donné que l'information dite d'intérêt public est dérivée ou extraite des dossiers enregistrés au niveau individuel.

* Les traitements que l'information peut suivre à l'hôpital comportent d'une part des diagnostics et des décisions et d'autre part l'application des décisions prises. Le plus souvent, les interprétations de l'information qui permettent d'aboutir au diagnostic sont obtenues sans l'aide d'outils informatisés. Le nombre d'informations peut donc rendre difficile ou impossible la détection de changements subtils dans l'évolution des données caractéristiques d'un malade.

Les caractéristiques de l'utilisation de l'information dans l'hôpital sont ainsi décrites par [BAKK 86]:

- * la disponibilité d'une information précise peut être cruciale dans le temps (ex: résultat d'un examen pendant une intervention chirurgicale),
- * l'inter-relation d'une information avec une autre peut être cruciale (ex: un résultat d'un test est dit pathologique, si le patient est sous l'influence de la drogue X. Un problème peut se présenter si les deux informations ne sont pas disponibles au même temps).
- * l'information peut avoir un caractère confidentiel,
- * la plus grande partie de l'information doit avoir un haut niveau de disponibilité, le cas contraire pourrait affecter le fonctionnement de l'hôpital,
- * Il est possible que, dans un département X, l'information concernant le patient soit incomplète, parce qu'elle se trouve dans le département Y, qui participe aussi au traitement du patient,
- * l'information doit être stockée pendant une grande période de temps pour les raisons suivantes:
 - l'intérêt du patient,
 - des raisons légales,
 - des recherches scientifiques rétrospectives
 - l'utilisation dans l'éducation.
- * les erreurs ou les déficiences dans la manipulation de l'information peuvent être catastrophiques pour le patient (ex: dose de médicaments).

Les hôpitaux rencontrent de graves problèmes d'optimisation et d'utilisation rationnelle de l'information. D'après les statistiques données dans [VRIE 86], la quantité d'information dans un hôpital se multiplie chaque décennie, et son temps moyen de vie est estimé entre 5 et 7 années. Cette croissance accélérée pose de graves problèmes d'organisation et d'utilisation de l'information, qui peuvent

rendre l'information pauvre ou même entraîner sa perte. C'est ici que les systèmes d'information hospitalière interviennent en fournissant un cadre général et intégré et en permettant le contrôle et l'utilisation optimale de l'information.

Un système d'information hospitalier (S.I.H.) fournit les informations utiles à diverses étapes de la réflexion médico-administrative, il recueille et traite en pratique quotidienne l'information médicale nécessaire à l'archivage, à la consultation de dossiers et aux recherches diverses [HEAU 81].

Plusieurs définitions du S.I.H. sont données dans la littérature, mais elles coïncident toutes dans le contexte. Nous trouvons par exemple dans [HUET 86] une définition donnée par BALL. M.J. aux Etats Unis qui exprime: "Un S.I.H. est un système basé sur ordinateur qui reçoit les données sur les patients normalement enregistrées, il crée et maintient à partir de ces données un dossier médical informatisé pour chaque patient, et permet d'utiliser les données pour les emplois suivants: soins, gestion administrative et commerciale, monitoring et évaluation d'activité des services, recherche clinique et épidémiologique, ainsi que la planification de ressources en santé".

Pour sa part [HUET 86] donne la définition suivante: "Un S.I.H. est constitué d'une collection minimale de relations (dans le sens défini par [CODD 70]) d'objets pouvant être stockés, traités, et transférés au moyen d'une collection d'opérateurs. Leur activation est déclenchée par des événements dans le but de participer à l'action médicale, à la recherche clinique, épidémiologique et biologique, à la planification hospitalière, et à la gestion quotidienne de l'hôpital".

Parmi les objectifs principaux fixés par la structure "hôpital" demandés au S.I.H. nous pouvons mentionner [CDCG 86], [HEAU 81], [CHAP 80] :

- assurer une bonne gestion de soins,
- donner les moyens d'acquérir l'information nécessaire pour identifier d'une façon non erronée un malade, et pouvoir suivre sa trajectoire médicale,
- identifier les faits cliniques et pathologiques,
- identifier les faits qui fournissent le contexte clinique de chaque malade,
- il faut pouvoir décrire l'ensemble des principaux faits biocliniques dans leur complexité et leur variabilité individuelle,
- il faut disposer d'une identification des pathologies et des contextes cliniques qui rende aisés et efficaces l'inventaire et la sélection des groupes de sujets à partir de données recueillies dans l'ensemble des malades,
- donner l'identification de procédures médicales, c'est-à-dire, de conduites thérapeutiques usuelles,

- faire une analyse fine de l'activité des services, ainsi que l'évaluation des soins qu'ils prodiguent,
- permettre d'exploiter les données médicales à des fins épidémiologiques,
- permettre d'exploiter les données médicales dans la recherche et dans l'enseignement,
- permettre d'avoir une connaissance de la maladie et de l'état de santé de la population,
- offrir une perception claire de l'information comme étant une ressource collective dont le fonctionnement doit être défini et contrôlé par la collectivité.

En termes généraux, l'intérêt d'un système d'information hospitalier est de fournir au praticien ayant en charge le patient, le maximum d'éléments fiables et cela le plus rapidement possible. Il convient de mentionner qu'avoir en charge un patient ne signifie pas obligatoirement avoir devant soi physiquement le patient; ceci est plus complexe et comprend aussi par exemple l'analyse des examens anatomo-pathologiques ou l'analyse d'une image tomo-densitométrique en temps différé par rapport à l'examen lui-même. Ceci concerne les données "en cours", c'est-à-dire, l'ensemble de données qui concernent les malades "actifs". Mais le plan d'activité peut d'ailleurs durer au-delà de la sortie physique du malade, parce que le dossier demeure "actif" une certaine période de temps après cette sortie, en participant à des activités autres que les soins au malade telles que la documentation clinique ou la recherche.

La réalisation d'un système d'information hospitalier repose sur une approche qui doit tenir compte des facteurs locaux, régionaux et nationaux, des objectifs des utilisateurs potentiels hospitaliers et des contraintes techniques [CHAP 80]. Il est donc un outil utilisé par le système hospitalier ayant des dimensions individuelles et collectives, et des caractéristiques à respecter.

Un S.I.H. présente un aspect "organisationnel" et une notion d'intégration qui correspond à sa dimension collective [CINQ 84]. L'aspect "organisationnel", regroupe les ressources humaines, matérielles et logicielles permettant d'atteindre les objectifs fixés. Tout agent de l'hôpital manipule des informations d'une manière plus ou moins consciente, ils sont donc tous des utilisateurs potentiels du S.I.H. Cependant, trois catégories d'individus sont distingués parmi ces utilisateurs: ceux qui ont à concevoir le S.I.H. dont l'un des rôles est de comprendre et de traduire les besoins de chaque catégorie d'utilisateurs, ensuite ceux qui développent et maintiennent le S.I.H. (cette catégorie est normalement formée par un noyau de spécialistes), et finalement les utilisateurs non spécialistes, lesquels peuvent être

des utilisateurs occasionnels du S.I.H. ou bien des utilisateurs qui travaillent avec le système pendant une proportion importante de leur temps de travail. Il est important que chacun de ces utilisateurs prenne conscience de l'importance du traitement de l'information dans leur travail; ceci peut être amélioré grâce à une motivation et à une association de chacun d'entre eux à la phase de conception.

La notion d'intégration représente la nécessité d'avoir une vue globale du système d'information [CDCG 86]. cette intégration peut s'étudier du point de vue du malade, du point de vue de l'utilisateur, ou selon un plan technique. Du point de vue du malade, l'intégration se traduit pendant par une démarche harmonieuse de la communication de l'information entre les divers services médicaux qui soignent le malade et les services administratifs. Cela signifie aussi l'optimisation de procédures de diagnostic et de soins, en fonction du malade et non en fonction des seuls intérêts de différentes unités concernées par ces procédures. Pour l'utilisateur, l'intégration se présente comme la possibilité d'accéder "en ligne" à toutes les données dont il a besoin et qui sont contenues dans le système. Cette facilité l'aidera à réaliser l'effort de collecter et d'enregistrer des données dont il n'a pas usage, mais qui peuvent être utiles à d'autres membres de la collectivité dont lui-même fait partie. Stockées dans le système, ces informations contiennent des éléments précieux pour les soins au malade, à condition de pouvoir être accessibles à tous ceux qui sont concernés.

Sur le plan technique, l'intégration se traduit par l'utilisation de matériels standardisés ou au moins compatibles entre eux. La conception centralisée des divers modules remplissant les fonctions du S.I.H. permet d'éviter leur redondance et assurer ainsi leur cohérence. Ainsi conçu, le S.I.H. devient un système très médicalisé et un outil puissant de communication entre les divers départements, services et médecins concernés par les soins.

1.2. Systèmes d'information et bases de données médicales.

Dans le secteur de la santé, les travaux ont davantage investi le domaine de la réflexion sur la finalité, la teneur et la circulation de l'information, que celui des applications de l'informatique. Le travail en milieu hospitalier exige entre autres choses:

- une haute fiabilité de l'information: elle doit donc être saisie le plus près possible de la source, et contrôlée immédiatement (transactions conversationnelles);
- une circulation efficace de l'information: dès qu'elle est connue, une information doit être accessible à tous ses utilisateurs potentiels (unités de soins, secrétariats médicaux, services médico-techniques, administration, etc);
- une bonne résistance aux pannes: une panne d'un composant du système doit le moins possible remettre en cause le fonctionnement de l'ensemble.

L'ensemble de ces exigences nécessite un produit informatique défini, standardisé et adaptable aux spécificités qualitatives et quantitatives des différents départements de l'hôpital. Une des principales fonctions du S.I.H. est l'intégration, le classement et la recherche de l'information, et les moyens informatiques généralement utilisés pour ces fins sont les bases de données. Aujourd'hui, parler d'un S.I.H. entraîne implicitement l'utilisation d'un système de gestion de base de données (SGBD). Ceci est justifié par la nécessité d'avoir un système dans lequel existe une intégration de l'information, partagée par plusieurs utilisateurs à la fois, dans des applications spécifiques à ces besoins. Ce système doit veiller sur la cohérence, la confidentialité et la sécurité des données. Les nouvelles techniques de base de données offrent de plus en plus de moyens pour pouvoir satisfaire les besoins des S.I.H. L'utilisation de ces technologies informatiques devrait permettre d'améliorer la prise en charge des malades, la qualité de soins et faciliter l'évaluation de l'activité médicale.

Nous présentons maintenant une brève description du développement qui ont suivi les techniques de base de données. Notre but ici n'est pas d'étudier les bases de données en détail mais de montrer leur insertion en milieu hospitalier. Nous faisons d'abord une introduction des technologies de SGBD, et ensuite nous présentons le lien entre ces technologies et le milieu médical.

1.2.1. Technologie des systèmes de gestion de bases de données (SGBD).

1.2.1.1. Définition des base de données.

Dès les années cinquante, sont apparus les systèmes de fichiers spécifiques d'une application, stockés sur cartes, bandes ou disques magnétiques, et organisés dans ce dernier cas en fichiers séquentiels, directs et indexés. Vers 1962 - 1963, le concept de base de données fait son apparition, "une base de données étant un ensemble structuré de données enregistrées sur des supports accessibles par ordinateur, pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en temps opportun" [DEAD 82]. Une autre définition de base de données est donnée par [SAUT 80] : "une base de données est une collection (probablement très grande) structurée et organisée de données d'un domaine d'information, sur laquelle il est possible de définir et de manipuler des applications spécifiques qui utilisent une sous-structure des données. Cependant, le terme "bases de données" est souvent utilisé dans la littérature dans un sens moins strict, et fait référence à un ensemble de fichiers qui utilisent des méthodes classiques d'accès.

Le logiciel qui permet à l'utilisateur de communiquer avec une base de données est un *système de gestion de base de données (SGBD)*. Il permet principalement de créer une base de données, d'organiser les données sur les supports périphériques et de fournir des procédures de recherche et de sélection de ces mêmes données.

1.2.1.2. Description des données, schéma interne, schéma externe, et schéma conceptuel.

Un SGBD doit mettre à la disposition de l'utilisateur un outil pour décrire l'ensemble des données qui participent à son application et qui seront stockées dans la base [DEAD 82]. En général, une application est caractérisée par :

- * propriétés statiques telles que les objets, les propriétés des objets (attributs), et les relations entre les objets,
- * propriétés dynamiques telles que les opérations et des relations entre les objets,
- * règles d'intégrité des objets.

La description des données peut se faire à différents niveaux:

- logique: qui est la description des objets et les relations entre eux, en considérant les contraintes qui serviront à maintenir l'intégrité des données, et la sécurité de la base,
- physique: fait référence à la structure physique de stockage des données sur les organes périphériques ainsi qu'aux chemins d'accès.

Ces niveaux de description sont à l'origine de trois niveaux de représentation des données:

- le niveau interne ou schéma interne, qui spécifie comment les données seront stockées sur les organes périphériques de l'ordinateur,
- le niveau conceptuel ou schéma conceptuel, qui décrit en termes abstraits mais fidèles une certaine réalité d'une organisation et de ses processus de gestion qui nécessitent la mise en œuvre d'une base de données,
- le niveau externe ou schéma externe correspond à une vision de tout ou d'une partie du schéma conceptuel par un groupe d'utilisateurs concerné par une application et chargé de mettre en œuvre des programmes d'utilisation.

1.2.1.3. Modèles de données.

Depuis l'avènement des premiers SGBD, aux alentours de 1962, on a assisté en vingt quatre ans à une évolution qualitative (progrès théoriques et technologiques) et à une évolution quantitative (nombre et variétés d'applications de base de données, conjuguées à une grande diversité de matériels informatiques).

Un SGBD est essentiellement caractérisé par le modèle de données qu'il supporte. Un modèle de données décrit le contenu de l'information en termes d'objets et de relations entre ces objets. Les objets sont des éléments de la vie réelle, lesquels sont caractérisés par des attributs ou propriétés. La conception du modèle de données est basée sur deux étapes:

- la perception du monde réel,
- l'abstraction du monde réel en ne considérant que ce qui intéresse le domaine d'application.

Le modèle est représenté au niveau informatique par une structure physique des données. Dans un S.I.H., les objets pourraient être par exemple: *patient, médecin, admission, etc.*

C'est ainsi que l'on parle de la première génération des SGBD s'appuyant sur les structures de données hiérarchiques et réseaux. Ils résultent d'une approche de conception de systèmes qui vise à étendre le système de fichiers par des possibilités de liaison interfichiers matérialisées par des pointeurs [GARD 85].

La deuxième génération est née vers 1970 avec l'apparition du modèle relationnel, mais ses produits sont apparus sur le marché vers 1980. Le modèle relationnel a été introduit par E.F. CODD [CODD 70], qui a précisé les premiers objectifs du modèle comme suit:

- 1.- permettre un haut degré d'indépendance des programmes d'application et d'activités interactives par rapport à la représentation interne de données,
- 2.- fournir une base solide pour traiter les problèmes de cohérence et redondance des données.

Une base de données relationnelle est composée de relations qui peuvent être vues comme un ensemble de tables à deux dimensions. Chaque relation est composée de colonnes caractérisées par un nom, et par des lignes qui représentent les données des colonnes. Les colonnes sont appelées constituants, et un domaine de valeurs (entiers, chaîne de caractères, etc) est associé à chacun. Pour sa part, les lignes sont appelées n-uplets, chaque n-uple est identifié par une ou plusieurs valeurs de constituants appelés clés, et il n'existe pas deux n-uples identiques dans une même relation.

Les opérations principales effectuées dans une base de données relationnelle sont: la création de relations, récupération des valeurs des n-uples en faisant intervenir une ou plusieurs relations, insertion, suppression et modification de n-uples.

Le succès du modèle relationnel est dû à la puissance et à la simplicité de ces concepts. Ces avantages sont principalement ses facultés à assurer une entière indépendance entre les descriptions de données logiques (en termes conceptuels) et physiques (en termes de fichiers), ainsi que des opérateurs puissants de l'algèbre relationnelle entre les relations (union, différence, produit cartésien, sélection, projection et jointure). De même, la souplesse d'utilisation est complète avec les modèles relationnels: si, à l'usage, on souhaite modifier l'organisation logique des données, en ajouter, ou en supprimer, on peut le faire aisément sans remettre en cause la cohérence du schéma de la base de données.

Contrairement aux systèmes de fichiers spécifiques de chaque application, les systèmes de base de données permettent donc d'utiliser les associations qui existent entre divers types d'information. Ils permettent d'augmenter la cohérence du

stockage des données, évitant en particulier leur redondance. Ce partage de données rend possible la centralisation, la coordination, l'intégration et la diffusion de l'information archivée. Il existe d'autres fonctions très importantes qui sont assurées par les SGBD relationnels tels que [DEAD 82]:

* **Description.**- Le SGBD doit mettre à la disposition de l'utilisateur un outil pour décrire l'ensemble des données qui seront stockées dans la base de données.

* **Utilisation.**- Le SGBD doit offrir à l'utilisateur une interaction avec la base de données sous forme d'un dialogue, pour chercher, sélectionner et modifier les données.

* **Intégrité.**- Le SGBD doit offrir à l'utilisateur la possibilité de définir des règles qui permettent de maintenir l'intégrité de la base de données. Ces règles, appelées contraintes d'intégrité, correspondent à des propriétés qui devront toujours être vérifiées dans la base de données, quelles que soient les valeurs enregistrées.

* **Concurrence.**- Le SGBD doit pouvoir gérer un accès multiple des utilisateurs à la base de données d'une façon concurrente.

* **Confidentialité.**- Le SGBD doit offrir les mécanismes permettant d'autoriser ou d'interdire l'accès aux informations.

* **Sécurité de fonctionnement.**- Le SGBD doit permettre la prise de point de contrôle pour remettre la base de données dans un état satisfaisant après une panne du système.

On voit ici que les principales fonctions (description, utilisation, intégrité, confidentialité, concurrence d'accès, sécurité de fonctionnement) des SGBD assurent une véritable sécurité des données qui leur sont confiées. On peut d'ailleurs, par l'utilisation des contraintes d'intégrité, faire découvrir à l'utilisateur une erreur dans une information qu'il croyait valide. Il est montré que le concept de relation s'adapte bien à la vérification de la cohérence des informations stockées par rapport à leur signification.

La simplicité du modèle relationnel a jusqu'à un certain point limité le traitement des données. Dans la plupart des cas, les SGBD de ce type sont conçus pour gérer des données alphanumériques structurées, et ils ne sont pas capables de gérer les données qualifiées de multi-média (telles que les images, les textes et plus

généralement les documents). D'autre part, les SGBD relationnels classiques ne permettent que de retrouver des données stockées dans la base. C'est-à-dire, qu'ils ne sont pas capables de déduire des données à partir de données et de lois générales.

Des extensions du modèle relationnel s'avèrent donc indispensables, tout d'abord pour offrir la possibilité de gérer des types des données plus variés, et ensuite pour supporter des possibilités déductives en étendant des outils pour définir et stocker des règles.

Bon nombre de chercheurs et concepteurs entrevoient finalement le modèle relationnel comme le support de développement d'une troisième génération de SGBD intégrant des données de types divers. L'idée est par exemple de partir d'un système relationnel et de lui faire supporter des domaines beaucoup plus riches, définis par l'utilisateur sous forme de types abstraits, ceci à partir des types de base du système ou de types déjà définis. Dans cette nouvelle génération, il s'agit de permettre une utilisation plus efficace de la sémantique des objets, en considérant leur faible structuration. Les bases de données généralisées et les bases de données déductives pourront se situer dans cette nouvelle génération.

Comme premier exemple, nous avons le projet TIGRE [VELE 84], [LOPV 83], où la philosophie est d'offrir un modèle de données orienté vers la manipulation des "objets" non classiques et peu structurés, comme le texte, les images, les documents, etc. L'intégration de ces objets et des données traditionnelles (entiers, chaînes de caractères, etc) constitue le "document généralisé". Le modèle TIGRE offre les outils pour définir et manipuler la structure et la sémantique d'un "document généralisé". Une description plus détaillée du modèle est donnée dans le chapitre trois.

Un deuxième exemple est représenté par le projet SABRE [ABSG 86] lequel tend à apporter des améliorations aux SGBD relationnels. Parmi les objectifs du projet, on trouve l'extension des types de données gérés par les systèmes relationnels, l'introduction de fonctionnalités déductives, l'amélioration du temps de réponse, le contrôle de l'intégrité physique des données dans les transactions simultanées, dans les pannes du système et dans les mises à jour de la base de données.

1.2.2. Applications médicales utilisant les techniques de SGBD.

La qualité du contrôle de l'information peut, dans plusieurs cas, donner une idée de la qualité du service de soins et même avant l'avènement de l'informatique en milieu médical, les professionnels de la matière se préoccupaient de maintenir des méthodes de stockage et d'accès à l'information. L'introduction de l'informatique dans le milieu médical remonte au début des années soixante, avec l'utilisation de systèmes basés sur des fichiers indépendants. Beaucoup d'efforts ont été investis dans l'idée de créer des systèmes généraux et complets, capables de gérer aussi bien l'information administrative que l'information purement médicale. Ensuite, les techniques de bases de données font leur apparition dans le milieu médical dans la décennie des années soixante-dix et un nombre de projets donnèrent des résultats positifs. Cependant, les frustrations ont été aussi importantes, dues surtout au manque de fiabilité de la base de données, venant du mauvais contrôle de l'information qui entrait dans la base, mais aussi à la difficulté d'accès aux données elles-mêmes [WIED 81]. Ces expériences ont quand même permis d'éclaircir les besoins médicaux que devait satisfaire un système informatisé [DIAS 86], [MUNZ 86]:

- le stockage des données hétérogènes,
- l'extraction précise des données correspondant à un individu spécifique,
- la possibilité de manipuler des types de données divers (historique du patient, résultats de laboratoires, diagnostics, radiologie, etc),
- la possibilité de modifier la structure de données sans avoir à modifier les programmes d'application,
- la facilité d'obtenir des statistiques sur les données stockées,

Avec l'évolution des SGBD, de nouvelles applications se sont développées, avec des résultats de plus en plus positifs. Aujourd'hui, on peut trouver des applications de ces techniques dans, par exemple, la gestion administrative des hôpitaux, les systèmes de documentation médicale, la recherche médicale, les systèmes d'aide au diagnostic, les services de soins et de prise en charge des patients, ou bien dans l'évaluation de la qualité de traitements, chacune de ses applications pouvant être un module d'un S.I.H.

Diverses applications médicales qui utilisaient l'un des trois modèles de SGBD (hiérarchique, réseau et relationnel) ont été développées, avec une tendance aujourd'hui à privilégier le modèle relationnel. Le modèle hiérarchique est utilisé par exemple dans la construction des S.I.H. [GRIE 78] ou dans des systèmes

orientés vers la recherche médicale [FOX 78], [SATO 86]. Ce modèle paraît très adapté pour modéliser certaines données d'une façon très naturelle, mais des problèmes surgissent au niveau opérationnel, venant parfois du modèle lui-même. En ce qui concerne le modèle réseau, la plupart des applications dans le domaine médical se trouvent dans la construction des S.I.H. [DUBI 77], [OHSA 78]. Dans le milieu médical on trouve de plus en plus des applications du modèle relationnel. Un grand nombre d'entre elles correspondent à la construction des S.I.H [JERU 86], mais d'autres sont purement médicales, par exemple [KANA 80], [ASSM 86].

L'idée principale d'utiliser les bases de données dans le domaine médical est d'avoir un outil qui facilite la gestion de l'information. Par là, nous voulons dire l'acquisition, l'archivage, la transmission, la récupération, la présentation et la diffusion de l'information. Ceci sont des besoins qui se présentent à tous les niveaux de la médecine, et elles sont confirmés par la présentation non exhaustive que nous faisons ensuite des différentes catégories médicales se servant des bases de données.

1.2.2.1. Le médecin indépendant.

Pour eux, la plupart des besoins de gestion sont satisfaits par des systèmes à petite échelle, permettant de gérer la liste des patients et les données administratives, ainsi que l'aspect économique du cabinet. En ce qui concerne les données médicales, si elles sont introduites dans l'ordinateur, on le fait en gardant un format simple et facile à gérer.

1.2.2.2. Le groupe médical indépendant.

On rencontre souvent le cas où un groupe de médecins et/ou laboratoire(s) travaillent ensemble. Si les dossiers des malades doivent passer d'une personne à une autre, et que chacune y apporte son information, alors leur accès peut devenir problématique. Dans de telles conditions, un système de base de données pourrait apporter beaucoup à la gestion de l'information médicale de base, en permettant par exemple l'enregistrement des diagnostics, les prescriptions, etc. De plus, un système de facturation peut se servir des données administratives contenues dans la base, en permettant ainsi un contrôle de la comptabilité du cabinet.

1.2.2.3. Le médecin spécialiste.

Un médecin spécialiste peut largement bénéficier de tels systèmes, étant donné que la plupart de ses patients sont des personnes avec des maladies longues ou bien des maladies chroniques. Ainsi, un système qui lui permet par exemple de repérer rapidement les résultats de tests ou de traitements suivis par le malade serait d'une grande aide. A la différence de l'information manipulée par les médecins généralistes (elle est plus variée), l'information traitée par les spécialistes peut souvent être bien structurée en schémas tabulaires. Ceci représente une manipulation plus facile du point de vue informatique, comme du point de vue des médecins.

1.2.2.4. La recherche médicale.

• Une utilisation importante des bases de données médicales est le support qu'elles peuvent apporter aux recherches cliniques. Quand l'information devient considérablement volumineuse, en termes de population ou périodes de temps, il s'avère indispensable de disposer des moyens informatiques permettant de la manipuler. Dans la recherche clinique, il est souvent convenable de coupler une base de données avec d'autres systèmes informatiques (p.e. logiciels statistiques, traitement d'images, etc).

1.2.2.5. Les hôpitaux et les cliniques.

Les problèmes de gestion de l'information dans ces entités sont complètement différents de ceux présentés précédemment. Ici, le patient passe au minimum par un service, et il peut arriver qu'il passe dans plusieurs services le même jour. Une information qui vient d'être ajoutée dans son dossier, peut être immédiatement utilisée par une autre personne différente de celle qui l'a générée. Par ailleurs, son dossier restera actif dans les archives de l'hôpital pendant plusieurs années. Nous retrouvons donc les problèmes présentés dans la section antérieure, et on revient donc à l'intérêt des S.I.H.

1.2.3. Quelques exemples de S.I.H. développés en France.

En France, la plupart de S.I.H. existants sont limités aux applications administratives [DUJO 86]. Les données qui y sont traitées à propos de chaque malade sont des données classiques avec une structure bien définie; elles sont utilisées pour contrôler: l'admission, le transfert, la sortie, la communication au service centralisé de dossiers médicaux et les applications comptables. Cependant, de nouveaux projets ont comme objectifs l'utilisation des données cliniques pour des fins autres qu'administratives (fins médicales, fins de recherche); ils font l'intégration des données plus complexes telles que les images ou les textes. Un bon échantillon de ces projets peut être trouvé dans les actes du congrès EUROMED 86. Quelques projets sont brièvement mentionnés par la suite à titre d'exemple:

1.2.3.1. Le système SORGHO.

Le S.I.H SORGHO développé au C.H.R. de Montpellier [DUJO 86]. Il est utilisé à des fins administratives (admission, transfert, sortie, communication au service centralisé de dossiers médicaux et applications comptables). Certains modules du système sont centrés sur le malade, ces modules (SORGHO-GESTION DES MALADES) sont articulés autour d'une bases de données contenant les renseignements sur les patients hospitalisés ou consultants. Il existe deux bases de données, respectivement base de données "en cours" (pour les patients en cours d'hospitalisation) et base de données "archives" (pour les patients ayant déjà séjourné au C.H.R.). Récemment, un module de bureautique intégrée a été greffé au système SORGHO, dont les fonctions essentielles sont d'enrichir les bases de données avec les différents textes médicaux générés durant le séjour du patient (compte rendu de sortie, compte rendu opératoire, compte rendu d'examen radiologique, etc). Le système PASTEUR permet de réaliser le stockage et l'extraction des comptes rendus, basé sur de fonctionnalités propres au service utilisateur.

1.2.3.2. Le système DIMI.

Le système DIMI est développé pour l'hôpital de Nantes [BIZA 86]. La finalité du projet est de permettre l'archivage numérique des images et leur contexte, ainsi que la distribution de ces images à travers un réseau local. Le bénéfice escompté est une amélioration de la gestion médicale et administrative des

services d'imagerie grâce à un accès plus rapide, plus facile et plus sûr de l'information. Un gestionnaire relationnel (Empress 32) est utilisé de façon à pouvoir assurer différentes visions des données gérées. Deux niveaux logiques de gestion de données sont distingués: un niveau local qui gère les données administratives, les images et le texte relatif à un examen en cours de réalisation, et un niveau global qui concerne le système dans sa totalité, et qui reçoit les données mentionnées une fois que l'examen est déclaré fini. A ce moment, les données deviennent accessibles à l'ensemble des utilisateurs autorisés, sans possibilité de modification des données médicales.

1.2.3.3. Le système SYMPHONIE.

SYMPHONIE est un système d'accès transactionnels d'une base de données répartie, conçu, réalisé et mis en place par les C.H.R. de Tours et Metz [BERT 86], [PIET 86]. L'objectif principal est la mise à disposition des produits informatiques correspondants aux besoins (médicaux, administratifs, et techniques) des hôpitaux. Pour cela, différents modules ont été développés:

- EURIDYCE : gestion administrative des malades
- PROMETHEE : gestion du dossier médical et de secrétariats médicaux
- HERMIONE : gestion des unités de soins
- GALATEE : gestion des unités médico-techniques
- OPHELIE : gestion des ressources économiques et financières
- DAPHNE : gestion du personnel

L'ensemble des modules constitue un système d'informations cohérent et homogène, centré sur le malade. Toutes les applications ou modules sont intégrées dans une base de données unique. Un réseau de miniordinateurs est établi entre les deux centres hospitaliers et la base de données est répartie sur les différentes machines du réseau.

1.3. Le service de radiologie.

Dans notre travail, nous voulons offrir aux médecins un outil se servant des nouvelles techniques de SGBD, qui leur permette d'avoir un accès plus souple et plus convivial aux données des examens radiologiques, et en même temps, fixer les bases d'une informatisation modulaire du service de radiologie, qui pourrait se

greffer au S.I.H. Nous travaillons dans le cadre d'une application réelle avec le service de radiologie du C.H.R.U. de Grenoble. Nous nous sommes limités à un domaine de la radiologie qui correspond à la tomodensitométrie ou scanner, et à l'imagerie par résonance magnétique, mais le système proposé devrait s'appliquer à tous les autres domaines de la radio. C'est dans ce but que nous faisons d'abord une description non exhaustive des activités du service et ensuite une description des objets qui composent un examen radiologique (comptes rendus et images radiologiques) et qui font partie du dossier radiologique.

La radiologie est un service de diagnostic qui utilise les images pour détecter des maladies et pour aider la conduite thérapeutique. Ces images sont interprétées par les médecins radiologues, qui réalisent un compte rendu lequel est envoyé au médecin qui traite le patient. Le service de radiologie du C.H.R.U. de Grenoble est divisé en deux grandes unités [CHU 87]:

- * la Radiologie A,

- * la Radiologie B,

ces deux unités peuvent se subdiviser (par rapport aux techniques utilisées ou au secteur médical exploré) dans diverses zones d'activité telles que:

- * La Radiologie Générale: chaque secteur de la Radiologie Générale travaille pour un certain nombre d'unités de soins, regroupées en fonction des spécialités (ex. secteur digestif et urologie, secteur osseux et pulmonaire). Tous les examens de radiodiagnostic général sont effectués ici. Le compte rendu généré dans chaque examen est produit en deux exemplaires, l'un qui est archivé par l'accueil pendant une période d'à peu près un an, et l'autre qui est archivé par le secrétariat pendant un période de deux mois. Dans les deux cas, un classement par ordre chronologique est effectué.

- * L'Angiographie: elle comprend deux secteurs, l'angiographie conventionnelle (abdomen, membres inférieurs et supérieurs) et l'angiographie numérisée (angiographie cardiaque et angiographie thoracique). Les comptes rendus générés suivent le plus souvent la même trajectoire que ceux de la radio générale. Un archivage d'images est fait par ailleurs, sur des films pour les images de coronarographie et sur cassette magnétique pour l'angiographie numérique.

- * Le scanner (Tomodensitométrie ou TDM): un examen de tomodensitométrie passe "grosso modo" par les étapes suivantes:

- prise de rendez-vous,

- recherche du dossier radiologique précédent,
- réalisation des images,
- copies des clichés,
- réalisation du compte rendu (par le médecin radiologue),
- frappe du compte rendu en double exemplaire.

Une fois l'examen fini, le dossier radiologique retourne à son archivage avec, en plus, la lettre de demande de l'examen, le double du compte rendu, l'original du cliché et éventuellement d'autres contre types. L'archivage de dossiers se fait en suivant un ordre croissant de numéro de dossier. Les dossiers jugés intéressants pour la recherche ou pour l'enseignement sont souvent complétés par des clichés complémentaires effectués pour le même malade et ils sont répertoriés et classés par le secrétariat du chef du service. La gestion du dossier radiologique est basée sur une fiche d'identification (voir figure 1.1). Cette fiche est remplie pendant le cycle de l'examen, et elle est ensuite saisie dans un microordinateur (APPLE II). Elle permet ainsi d'éditer des listes de dossiers radiologiques triés par:

- ordre alphabétique sur le nom du malade,
- date d'examen,
- région explorée,
- code du diagnostique,
- numéro du dossier.

volumineuse et complexe des textes et des images, pose de sérieux problèmes d'archivage, d'exploitation et de diffusion dans le fonctionnement quotidien du service. Ainsi, une gestion inefficace de l'information se reflète dans une mauvaise prise en charge du patient, des examens incomplets ou insatisfaisants, des retards dans l'élaboration des comptes rendus, ainsi que dans l'impossibilité de retrouver des images qui peuvent être importantes pour l'aide à la décision médicale. De plus, la vie économique du département se voit affectée par l'augmentation du coût des opérations.

Or, il s'avère indispensable de réaliser une intégration plus précise des informations extraites à partir des images avec les autres données médicales qui caractérisent le patient et qui forment "le dossier radiologique", de telle façon que l'ensemble des informations aide le médecin à utiliser ses connaissances pour naviguer au mieux parmi elles. Vouloir informatiser le dossier radiologique procède du désir de pouvoir exploiter, diffuser et conserver ses composants d'une manière rapide et précise. Le dossier radiologique est ainsi composé de deux grands groupes d'informations:

- a) les données qui identifient le patient: données administratives telles que nom, date de naissance, etc,
- b) l'examen radiologique qui comprend à la fois les images produites (images numériques ou analogiques) d'une partie du corps humain et l'information écrite (compte rendu); celle-ci contient, d'une part, les commentaires relatifs aux indications de la réalisation des images, à la technique d'obtention et aux éventuels incidents survenus et, d'autre part, elle contient l'interprétation des images, c'est-à-dire l'apport de ces images dans la compréhension de la maladie du patient. Nous faisons ensuite une description plus détaillée des images et des comptes rendus produits par ce service de radiologie.

1.3.1. L'image radiologique.

L'imagerie médicale a ses origines dans la découverte des rayons X par Mr. Wilhelm Konrad Röntgen en 1895 [GWIL 86]. Au début, la pratique de cette technique était appelée roentgenologie, mais l'utilisation de rayons X a donné origine au terme radiologie. Diverses médias ont été utilisées pour enregistrer et stocker ces images radiologiques, parmi lesquelles nous pouvons mentionner les films, et les celluloïdes. Ces types de médias sont très coûteux et représentent un vrai problème de stockage et de gestion des images. Malgré les nouvelles techniques

utilisées pour augmenter l'efficacité d'enregistrement des rayons X, pendant les 75 années qui ont suivi leur découverte, les images radiologiques ont été exclusivement analogiques. L'invention de l'ordinateur est venu révolutionner l'imagerie médicale et l'imagerie radiologique en particulier. Au début des années 70, on voit apparaître le tomographe numérique inventé par Mr. Haunsfield et l'on passe ainsi du film à la bande magnétique et ensuite au disque numérique. Plus tard l'imagerie par résonance magnétique fait son apparition.

L'évolution rapide des techniques, l'apparition de méthodes de reconstitution et de traitement mathématique des images ont permis ces dernières années une véritable révolution dans les méthodes de diagnostic. Aujourd'hui, l'imagerie médicale exploite pratiquement toutes les ressources de la physique: rayons X, ultra sons, radioactivité, résonance magnétique nucléaire, etc. Ces dix dernières années ont permis le passage de l'observation visuelle de clichés à la quantification et au traitement d'informations numériques multidimensionnelles morphologiques, fonctionnelles et métaboliques.

Dans la stratégie du diagnostic, les images sont utilisées comme complément d'autres informations cliniques, ou obtenues par d'autres examens complémentaires. Les images sont donc elles mêmes génératrices de nouvelles informations, ces informations peuvent être d'ordre morphologique (forme, limites, silhouette, et volume d'un organe), structurelle (texture tissulaire) ou fonctionnel. Cependant des problèmes se posent dans l'obtention de cette information. D'une part, l'information contenue dans une image n'est pas toujours facile à lire, d'autre part, il est souvent nécessaire de quantifier cette information, étant donné que le diagnostic se fait, non seulement par inspection visuelle de l'image, mais à l'aide de mesures géométriques ou densitométriques sur l'image. En effet, lorsqu'on a besoin de critères objectifs sur lesquels puissent s'appuyer le diagnostic ou la thérapeutique, il est souvent indispensable de pouvoir quantifier certains critères pour suivre leur évolution dans le temps.

Le but de l'imagerie médicale, c'est l'obtention de l'image du corps humain par tous les moyens physiques autres que la lumière naturelle ou artificielle, permettant de voir les tissus, les organes, et l'organisme. Il existe deux grandes méthodes d'obtention d'images radiologiques [LAMA 86]:

-les méthodes non ionisantes:

- * les ultra-sons,
- * l'imagerie par résonance magnétique (I.R.M.),
- * la thermographie;

-les méthodes ionisantes; définies par leur action sur la cellule; elles se décomposent en deux grands groupes:

- * imagerie ionisante: elles utilisent les rayons X en technique, soit conventionnelle (radiographie sur films au bromure d'argent et techniques dérivées), soit digitalisée (numérisation à partir de capteurs ex. scannographie ou tomographie),
- * les sources non scellées: ce sont les produits radio-actifs (isotopes) dont le marquage cellulaire donne des images plus spécifiques et toujours fonctionnelles.

La numérisation d'une image est l'opération qui fait passer de l'image initiale analogique (déjà formée ou virtuelle) à un codage numérique utilisable par l'ordinateur. Conceptuellement la numérisation d'une image comprend un codage spatial et un codage en densité. Le codage spatial consiste à diviser l'image en $n \times n$ pavés appelés *pixels*. La taille commune de chaque pixel fixe la résolution spatiale de la numérisation. Le codage en densité associe à chaque pixel un nombre de bits pour lui coder. Ainsi, un pixel codé sur 1 octet autorise 2 puissance 8, niveaux différents de codage. L'image numérisée est rangée ensuite par ligne ou par colonnes.

Pour certaines techniques (ex. tomодensitométrie), la numérisation est intrinsèquement liée à l'obtention de l'image, mais, dans d'autres cas, la numérisation permet des traitements destinés à améliorer l'apparence visuelle des images et à résoudre des problèmes d'imagerie fonctionnelle. Une information peut donc être présente sur une image, mais indécélable par l'œil humain, à moins d'un traitement numérique spécifique. D'autre part, certaines images peu contrastées peuvent devenir lisibles avec l'utilisation d'algorithmes permettant de rehausser le contraste. En dehors du traitement, la numérisation des images permet leur transmission et leur archivage avec rapidité et efficacité.

Faire un archivage des images, sans prendre en compte les possibilités de récupération et d'interaction avec d'autres données, peut produire un "cimetière d'images". Leur stockage dans une base de données doit permettre l'accès rapide

nécessaire à une meilleure utilisation des images et la maîtrise d'un volume croissant d'archives. En effet, indépendamment du gain qualitatif, il faut prendre conscience du fait que l'archivage traditionnel est en train de devenir impossible à gérer et qu'une image stockée trouve sa vraie valeur seulement dans le cas où un bon système de gestion d'images la rend disponible au bon moment. La gestion des images confronte des problèmes typiques liés à ce type d'information tels que: information très volumineuse et peu structurée, différentes possibilités de représentation (taille du pixel, résolution, compression, transformation, etc); à titre d'exemple nous pouvons citer:

- * des images de tomодensitométrie codées en 512 x 512 sur une densité de 12 bits; chaque image représente environ 3 Mégabits,
- * une journée d'utilisation du scannographe représente un volume d'environ 60 Mégaoctets,
- * pour une échographie numérisée en temps réel sous forme d'images 256 x 256 avec une densité de 7 bits, 5 minutes d'examen représentent 27 Gigaoctets.

Il est donc indispensable de pouvoir sélectionner les images. Cette sélection obligatoire peut être plus ou moins restrictive selon que l'on attend un ensemble d'images qui rende compte de la totalité d'un examen ou, au contraire, une ou quelques images isolées consolidant l'interprétation d'un examen. Cette sélection doit de toute manière permettre de disposer des images indispensables. Certaines images sont indispensables au moment de l'examen parce qu'elles donnent au thérapeute des informations nécessaires et suffisantes, d'autres peuvent devenir indispensables à moyen ou long terme, car elles permettent une analyse évolutive.

Comme nous l'avons mentionné, la gestion des images comprend donc trois étapes: l'archivage, l'exploitation et la diffusion.

En ce qui concerne l'archivage, ceci est un problème délicat compte tenu de la quantité d'images créées tous les jours à l'hôpital. Il faut tout d'abord savoir pourquoi on conserve les images. Il est évident qu'elles doivent être conservées jusqu'à leur interprétation. La question est plus délicate une fois que l'image a été interprétée. Les images sont conservées pour servir de référence, par exemple, pour juger de l'évolution d'une lésion, ou pour des raisons médico-légales, ou encore pour pouvoir redresser a posteriori une erreur d'interprétation (anomalie non détectée, que ce soit à la suite d'une erreur de l'observateur ou que l'image ne

prenne sa signification pathologique que dans un contexte clinique ultérieur). Il ne semble donc pas indispensable de conserver toutes les images produites. D'autre part, certaines images méritent d'être conservées à titre documentaire, en raison de leur intérêt médical pour la communauté. Il faut alors associer à ces images des informations qui les décrivent et qui reprennent la pathologie présentée par le malade, de manière à permettre une exploitation efficace des images conservées. Il faut ensuite savoir combien de temps on conserve une image. C'est un problème qui n'avait pas tellement d'intérêt tant que le moyen de stockage unique était le film, dont le recyclage possible n'est pas financièrement très rentable. Cependant, le problème se pose différemment de nos jours, avec les moyens de stockage numérique, où la mémoire est un facteur de poids et sa récupération a une grande valeur (à l'exception pour le moment des disques optiques numériques, où la réécriture n'est pas encore mise au point [DEKE 86]). Il faut aussi savoir comment on stocke l'image, c'est-à-dire, qu'il faut choisir le matériel physique (choix du support), mais aussi il faut réfléchir sur la manière de retrouver une image dans un ensemble d'images. Ceci est souvent très lié aux objectifs retenus pour l'utilisation de l'image.

En ce qui concerne l'exploitation des images, on considère les deux aspects santé individuelle et santé publique. Pour le premier cas, le but de l'exploitation est d'extraire les informations contenues dans l'image et de les traduire en termes compréhensibles par les cliniciens demandeurs de l'examen. Dans le deuxième cas, l'on souhaite utiliser également cette information, mais à des fins statistiques (recherche clinique ou épidémiologique, gestion administrativo-médicale du service), ou pour faire évoluer les connaissances médicales, ou encore à des fins didactiques.

Les problèmes liés à la communication des images sont très proches de ceux que soulève leur stockage. Là encore, il semble nécessaire de sélectionner ce qui devra être transmis, en choisissant selon les cas de transmettre l'image brute, l'image comprimée, une image traitée ou un compte rendu. Il est donc nécessaire de préciser d'abord les besoins en matière de transmission des images. Il y a ensuite les problèmes matériels (moyens physiques de communication) et logiciels (contrôle et modalités précises de la transmission des images).

1.3.2. Les comptes rendus.

Un compte rendu médical est un document structuré qui fait une description linguistique; d'une image radiologique, il est en relation avec une certaine pathologie concernant un malade. Le compte rendu comporte plusieurs éléments (voir figure 1.2), certains concernant les données administratives (identité du patient, nom du médecin prescripteur, date de l'examen, etc) et les autres concernant les données médicales (le motif de l'examen, le type d'examen, les éventuels incidents et accidents, le commentaire des images, et enfin leur interprétation ou conclusion). Les compte rendus sont normalement structurés de la manière suivante:

- introduction: relative à la réalisation des images et à la maladie du patient; elle contient normalement les motifs de l'examen ainsi que le type d'examen,
- constatations: partie analytique qui décrit les constatations de la perception visuelle du radiologue (commentaire des images),
- conclusion: partie synthétique qui reflète l'interprétation que le radiologue fait des signes ou des combinaisons de signes détectés dans l'analyse.

Le problème qui caractérise les comptes rendus est plutôt de type bureaucratique, c'est-à-dire, il est critique surtout en ce qui concerne l'archivage et l'extraction. La difficulté de cet aspect de la gestion est relative au degré de finesse demandé au système choisi pour ce propos. En ce qui concerne l'archivage, trois supports sont normalement utilisés:

- * l'archivage manuel en utilisant des dossiers papiers,
- * l'archivage sur des supports magnétiques en utilisant des systèmes informatisés,
- * l'archivage sur microfiches qui peut être un système manuel, ou un système informatisé ou bien un mélange de ces deux approches.

L'accès aux comptes rendus peut aller des moyens simples et primitifs comme l'accès manuel à travers l'identification du patient (ex. nom, No. de sécurité sociale) aux moyens plus sophistiqués comme l'accès au compte rendu en langue naturelle. Entre ces deux extrêmes, il existe toute une série de moyens d'accès parmi lesquels nous pouvons citer: l'accès par données administratives, l'accès en texte libre (cf. chapitre 4.1.2.1), l'accès en utilisant des mots clés, l'accès à travers un compte rendu type et dirigé.

Comme c'était le cas pour les images, un problème existe dans l'estimation du temps pendant lequel il faut conserver les comptes rendus, le même type de conséquences s'appliquant pour cette information. L'étude d'une solution devra alors prendre en compte les deux entités (comptes rendus et images), et le résultat de cette étude pourra bien entendu être différent dans chaque cas.

Un état de l'art des systèmes ayant étudié les différents problèmes présentés au long de cette section est présenté dans le chapitre suivant.

05.03.85

Docteur TOURNEUF

Monsieur DUPONT Pierre - 9eme II C

Tomodensitométrie en urgence:

Exploration de l'anévrisme de l'aorte thoracique descendante. Homme de 35 ans. Coarctation de l'aorte isthmique opérée en 1961 selon l'intervention de CRAWFORD. Récidive en 1975 amenant à faire un pontage utilisant l'artère sous-clavière gauche. Apparition récente et brutale d'une hémoptysie grave de 800cm³ avec encombrement post-hémoptoïque. Radiographie thoracique: opacité alvéolaire en projection du lobe inférieur. Fibroscopie: pas d'anomalie muqueuse, mais traces de sang dans l'arbre bronchique gauche et surtout au niveau de la nelson et de la postero-basale.

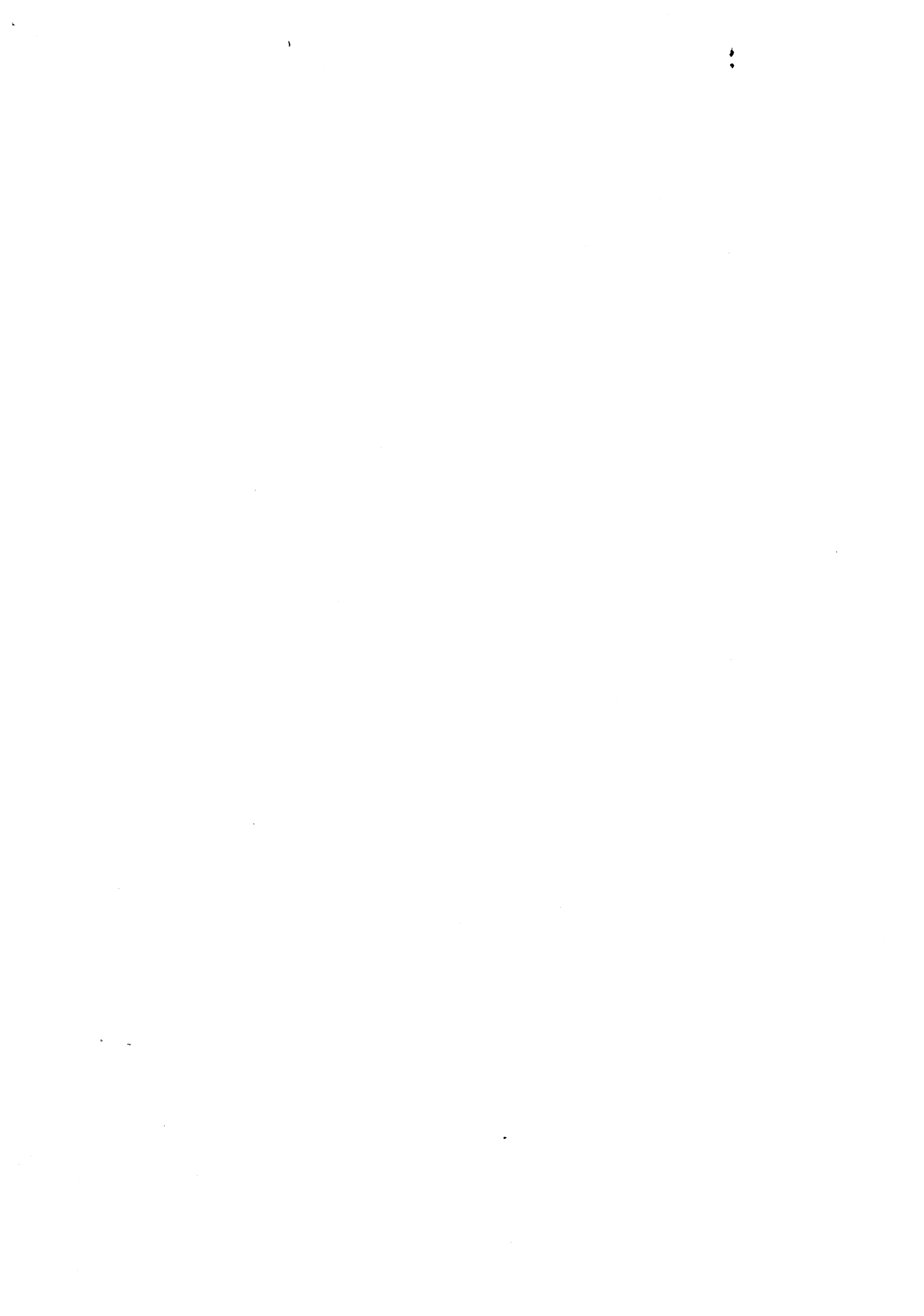
Les constatations sont les suivantes:

Augmentation modérée de calibre de l'aorte ascendante et de l'aorte thoracique descendante en dehors de la zone anévrysmale. importante dilatation de l'origine du tronc artériel brachiocéphalique. bonne visualisation du pontage aorto-aortique utilisant l'artère sous-clavière gauche. L'anévrisme proprement dit apparaît étroitement en rapport avec la bronche segmentaire apical du lobe inférieur, depuis son origine jusqu'à sa division. Il existe par ailleurs une opacité systématisée correspondant au segment apicale du lobe inférieur comblement alvéolaire ou hypoventilation. Sur la coupe no11, on retrouve la structure du bord gauche de l'anévrisme avec une image en téton isolée du sac anévrysmal.

figure 1.2

CHAPITRE II

ETAT DE L'ART



CHAPITRE 2

2. ETAT DE L'ART DES SYSTEMES DE GESTION D'IMAGES RADIOLOGIQUES ET DES SYSTEMES DE GESTION DE COMPTES RENDUS.

Traditionnellement, le concept de "système de gestion de base de données" (SGBD) fait référence à la manipulation de données alphanumériques structurées. Plus récemment, on parle de "système de gestion de base de données généralisées", pour les bases qui gèrent des données volumineuses et éventuellement à structure interne complexe, telles que: textes, images, graphiques et sons. Ce type d'information est aussi appelée données multi-média. Contrairement aux données alphanumériques, elles ne sont pas atomiques et, de plus, leur taille importante pose de nouveaux problèmes de stockage et d'accès. Nous présentons ici une synthèse des travaux qui proposent des solutions aux problèmes de gestion de données multimédia telles que les images et les comptes rendus. La description de ces travaux n'est pas exhaustive, parce que notre but principal est de donner une idée globale des approches qui ont été suivies pour répondre aux exigences de gestion de ce genre d'information.

Etant donné que, dans la plupart de la littérature consultée, il n'y a pas une intégration précise des deux données multimédia auxquelles nous nous intéressons (images radiologiques et comptes rendus), nous avons fait une séparation des systèmes gérant principalement les images, avec ceux gérant les comptes rendus. Pour ce qui concerne les images, nous nous sommes étendus sur des images autres que les images radiologiques, afin de montrer leur gestion dans d'autres domaines tels que la cartographie, les images de satellite, etc.. Finalement, nous faisons une présentation de notre approche pour gérer les dossiers radiologiques.

2.1. Etat de l'art des travaux réalisés sur les systèmes de gestion d'images.

Les problèmes d'efficacité dans le stockage et la récupération précise des images sont devenus plus importants avec l'avènement des nouvelles technologies de systèmes d'obtention d'images, tels que la tomographie, le scanner, les satellites, etc. S'il existe une demande pour une gestion d'images, c'est surtout pour améliorer leur exploitation, qui consiste à pouvoir récupérer, parmi un grand nombre

d'images, une image précise (ou un ensemble précis d'images). L'intérêt de récupérer un objet image se manifeste de différentes façons:

- * affichage pour l'observation,
- * traitement, avec transformation de l'image,
- * transmission,
- * "hard-copy".

Dans cette section, nous effectuons une synthèse des travaux réalisés sur le thème de "bases de données d'images", qui peuvent être considérées comme un sous-ensemble des bases de données généralisées. L'objectif principal de ce type de systèmes est de gérer le stockage et la manipulation des images d'une façon efficace. L'un des plus grands problèmes à résoudre est lié à la structure complexe et volumineuse qui caractérise ce type de données. Cependant, d'autres critères [MATR 86], [ROGA 85] interviennent dans la conception d'une base de données images, tels que:

1.- leur provenance :

- scanner,
- digitaliseur,
- photographie aérienne,
- capteur,
- etc.

2.- leur représentation au stockage :

- format vecteur (ou segment),
- format matriciel (ou grille),
- compression,

Le format vecteur consiste à représenter des points, des lignes et des polygones par leurs coordonnées. Le format matriciel consiste à associer à chaque point de l'image une valeur numérique donnant sa couleur. Les points sont organisés en lignes et en colonnes. Le choix entre les deux formats de stockage est principalement fonction de la complexité du contenu de l'image et des traitements qu'on veut lui appliquer.

3.- les types de données à gérer:

- des images (matrices, etc),
- des descripteurs des images (taille, type d'image, etc),
- des objets symboliques (routes, villes, rivières, etc),
- des primitives graphiques (point, ligne, polygone, etc).

4.- les opérateurs qu'on leur applique :

- opérations ensemblistes (superposition par union, intersection, etc),
- opérations géométriques (calculs de distance et de surface),
- opérations topologiques (appartenance d'un point à une région, etc),
- opérations de type traitement d'images (filtrage, zoom, agrégation, détection de contour, etc).

Les opérations appliquées sur les images sont liées à un mode de représentation physique; par exemple, le calcul du point d'intersection entre deux lignes n'est facilement réalisable qu'en format vectoriel.

5.- le moyen de stockage :

- disque optique numérique,
- disque magnétique,
- bande magnétique,
- film,
- stockage analogique,

6.- les différents types d'application:

- application médicale,
- application géographique,
- applications graphiques,
- etc.

Nous avons fait une classification assez générale des systèmes de gestion d'images, en prenant en compte leur orientation vers un domaine spécifique d'application. Un cadre schématique contenant tous les systèmes décrits est donné à la fin du chapitre.

2.1.1. Les applications médicales.

L'archivage et la gestion des images est un problème qui préoccupe le milieu médical; dans un hôpital de taille moyenne, on peut produire jusqu'à un million d'images de types divers par an [TAMU 80]; elles peuvent aller des images analogiques produites par la radio conventionnelle, jusqu'aux images numériques du scanner ou de l'imagerie par résonance magnétique. D'une manière idéale, toutes les images devraient être intégrées dans un seul système de gestion, mais

pour le moment on se contente de construire un système spécifique pour chaque type d'image. Dans [JAME 86] on trouve une analyse des aspects que doit couvrir un système de gestion d'imagerie médicale, parmi lesquels nous pouvons mentionner:

- * améliorer et standardiser le cycle d'un examen,
- * considérer les aspects économiques et humains de l'examen,
- * les images doivent être stockées de manière à pouvoir être reproduites et/ou affichées quand on en a besoin,
- * des informations doivent être associées aux images qui ont été modifiées, en indiquant les opérations appliquées aux images originales,
- * la vitesse de récupération des images doit être au moins similaire à celle obtenu dans un système manuel.

Souvent, le médecin ne s'intéresse pas aux images dites "normales", mais plutôt aux images dites "pathologiques" quand elles représentent clairement une maladie ou les conditions d'une pathologie. Diverses approches ont été prises dans le développement des systèmes de gestion d'images destinés à satisfaire ces besoins. On trouve par exemple des extensions des modèles de données existants, comme c'est le cas dans [HACH 86] où le modèle relationnel a été étendu avec un nouveau type de données (type "picture"). Dans la plupart des systèmes, on fait un traitement abstrait des images en les divisant en images "physiques" et images "logiques". L'image "physique" correspondant à l'image numérique ou l'image analogique, tandis que l'image "logique" représente des informations (les plus souvent alphanumériques) qui peuvent ne pas faire partie de l'image même, mais qui sont en étroite relation avec elle et servent à l'identifier d'une manière unique. Ainsi, les images sont en quelque sorte indexées par ces caractéristiques et c'est sur elles que l'utilisateur va filtrer. Ces systèmes ont un grand intérêt pour rendre invisible à l'utilisateur les formes de représentation physique des images. Divers travaux ont été présentés dans ce contexte et nous allons en détailler quelques uns.

2.1.1.1. Le système IQS.

(Image Query System) [HACH 86].

Le système a été développé à l'université de Kyoto (Japon), pour être utilisé principalement dans le domaine médical, mais étant donné sa structure assez générale, il pourrait être utilisé dans autres domaines. Les données sont divisées en deux groupes, les données image, équivalent des images "physiques", et les données non-image qui comprennent les images "logiques", plus les autres données alphanumériques. Les données images sont stockées séparément des autres données qui sont gérées par un SGBD relationnel. L'objectif du système est d'offrir un mécanisme qui permet de récupérer les images stockées dans une base, en utilisant trois critères principaux, à savoir:

- * une image clé : obtention de données non-image à partir de l'image affichée,
- * contenu de l'image : obtention des données (non-image ou image) qui sont en relation avec les images qui ont une certaine caractéristique,
- * similarité : récupération des autres images qui sont "similaires" à l'image affichée.

L'obtention des images "physiques" se fait grâce à l'utilisation de pointeurs existants entre les images "logiques" et "physiques".

Le système de traitement d'images ILIAD est utilisé, d'une part, pour extraire les caractéristiques des images qui seront intégrées dans la base de données et exploitables directement par le langage de requêtes du système. D'autre part, dans le cas d'une requête sur des conditions de similarité, il sert à comparer les images et déterminer si elles sont similaires ou pas. L'utilisateur peut définir des fonctions de traitement d'images (ex. détection de tumeurs, détection de zones denses, etc.), qui sont intégrées dans le langage de requêtes et exécutées par ILIAD. Un processeur de requêtes a été conçu pour contrôler le flux de données. Il fait une analyse et une décomposition des requêtes et envoi au SGBD et au système ILIAD leur requête correspondante. Ensuite, il récupère les réponses de ces deux modules et contrôle l'affichage des données.

L'interrogation est faite en utilisant un langage non-procédural similaire à SQL. GET est la commande principale du langage. Elle peut contenir des appels aux fonctions propres au système de traitement d'images. L'argument de ces fonctions

est un constituant de type "picture", il détermine les images sur lesquelles la fonction sera exécutée. Par exemple, la requête suivante contient la fonction AVEGRAY préalablement définie par le langage du système ILIAD. Cette fonction fait le calcul de la moyenne des niveaux de gris, et prend comme argument les images représentées dans le constituant INSPECT.IMAGE. La requête rend les images qui ont une moyenne de niveau de gris supérieur à 50.

```
GET INSPECT.IMAGE WHERE AVEGRAY(INSPECT.IMAGE) GT 50;
```

2.1.1.2. Le système IDBMS.

(Image Data-Base Management System) [ASSM 86], [ASSM 84], [ASSM 83].

Un prototype du système a été développé à l'hôpital de l'Université d'Hamburg (R.F.A.), en prenant comme application pilote un secteur de la radiologie, mais normalement le système doit être applicable dans tous les secteurs de la radio. Le but du système est d'offrir aux médecins un outil informatique, leur permettant de travailler naturellement et dans de meilleures conditions avec les images. L'aspect principal du projet était la gestion de données images, sans regarder beaucoup les aspects traitement et interprétation automatique. Les données images (image "physique") sont stockées dans des fichiers standards, séparément des données qui les décrivent, afin de permettre une plus grande flexibilité dans les types de représentation des images. Pour sa part, les données "logiques" sont gérées par un SGBD relationnel (ORACLE), et elles sont divisées en deux classes:

- * les données utilisateur.- comprenant les données structurées concernant le patient (nom, date de naissance, no.dossier, etc), ainsi que les données "logiques" qui sont en relation avec la sémantique des images (organe, source, diagnostic, etc),
- * les données dictionnaire.- comprenant toutes les informations concernant la structure et le stockage des images.

Cette division est faite afin d'éviter à l'utilisateur toute manipulation concernant le stockage des images "physiques". A cet effet, un interpréteur (ISQL) a été construit et sa fonction est d'offrir un accès cohérent aux images "physiques" et aux images "logiques". L'utilisateur a un contact direct avec les données qui lui sont destinées, alors que l'interpréteur ISQL utilise le dictionnaire pour répondre aux interrogations.

Le langage de manipulation SQL est étendu avec les mots clés "IMAGE" et "DISPLAY". Le mot clé "IMAGE" sert à manipuler les images, il peut être utilisé avec les commandes SELECT, INSERT et DELETE. Quand l'interpréteur identifie ce mot clé, il utilise les données du dictionnaire pour compléter la requête de l'utilisateur. Pour sa part, le mot clé "DISPLAY" ne peut être utilisé qu'avec la commande SELECT et sert à spécifier le format d'affichage des images.

L'utilisateur communique avec le système dans un contexte "image". Il peut demander par exemple:

- * obtenir une résolution différente pour une région spécifique de l'image affichée,
- * faire connaître au système un diagnostic. Par exemple, le médecin signale (à l'aide de la souris, etc) une région dans l'image affichée et l'identifie comme une tumeur. Le diagnostic va être alors stocké dans la base avec la référence à la région qu'il a signalée et aussi avec les données concernant le malade. Ainsi, la prochaine fois qu'il récupérera cette image, il trouvera la région marquée et identifiée comme une tumeur.
- * couvrir pour d'autres images qu'il signale sur l'écran, le même contour que celui qu'il a marqué dans une autre image.

2.1.1.3. Le système GRAIN.

(Graphics-Oriented Relational Algebraic Interpreter) [CHAN 77], [CHAN 79], [CHAN 80].

Le système GRAIN fait partie du S.I.H. développé par l'université d'Illinois (U.S.A.), il est utilisé par la section de vue et d'audition. Les images analogiques et les images numériques sont gérées à travers les concepts d'images "logique" et "physique". L'images "logique" est définie comme étant une collection d'objets qui sont stockés dans une base de données relationnelle à l'aide de trois tables:

- * POT = table d'objets.- contient le nom de l'objet (il doit être unique dans la relation), l'identificateur de son propriétaire (dans le cas où l'objet est un composant d'un autre objet), ses coordonnées de commencement (coin inférieur gauche), son angle par rapport au propriétaire et des attributs qui le caractérisent.

* PCT = table de contours.- contient le nom de l'objet, le nom du fichier où est stockée l'image physique, le format de coordonnées de contour de l'objet, ainsi que les coordonnées de contour elles-mêmes. Chaque n-uplet de la relation POT a un n-uplet associé dans la relation PCT.

* PPT = table de pages.- contient le nom de l'objet, le nombre de pages nécessaires pour stocker l'objet, la taille de chaque page, la localisation de chaque page, ainsi que la position de l'objet dans la page.

Les images "physiques" sont stockées séparément dans des fichiers standards en utilisant une technique de stockage par pages. Un système de gestion de stockage (ISMS) a été construit pour récupérer les images une fois qu'elles sont identifiées par les images "logiques".

La manipulation de données est faite en utilisant RAIN (Relational Algebraic INterpreter) qui est en fait un langage relationnel traditionnel, avec en plus d'autres commandes du système GRAIN qui servent à tracer l'image (pour les images analogiques) ou à l'afficher (pour les images numériques). Une fois que les images ont été indentifiées grâce aux images "logiques" et à l'utilisation de RAIN, l'utilisateur peut réaliser les opérations suivantes :

- * tracer la figure ou la forme (commandes SKETCH, PLOT),
- * faire l'affichage (commande PAINT).

L'introduction d'un nouvel objet est fait en utilisant les commandes DRAW qui permet de désigner l'objet à l'aide d'un éditeur graphique, et la commande PDEFINE qui sert à faire la description du nouvel objet.

2.1.1.4. Le système EL.

[YOKO 82].

Le système a été développé au Laboratoire d'Electrotechnique d'Ibaraki (Japon), pour être utilisé dans le cytodagnostic. Il gère une collection d'images de cellules pathologiques et microscopiques, qui sont considérées comme étant des spécimens typiques. Il a deux objectifs, le premier est d'offrir aux cytologistes un accès rapide aux images et aux diagnostics, le second est de faciliter l'accès aux images pour les systèmes d'évaluation d'algorithmes d'analyse d'images cytologiques. On distingue trois types de données, à savoir :

- les données alphanumériques (description de patients),
- les données textuelles (diagnostics des médecins),
- les données images (images digitalisées de cellules sous microscope).

Ces données sont logiquement représentées dans une base de données relationnelle. Les données textuelles et les données images sont stockées hors de la base, qui connaît le chemin d'accès à ces données ainsi que leurs descripteurs. L'accès aux images se fait à travers leur nom ou leur numéro, tandis que l'accès aux diagnostics se fait en utilisant le numéro qui les identifie. Pour obtenir ces numéros, il faut utiliser des données structurées telles que le nom du patient, la date de l'image, le nom de l'image, etc. Le système de traitement d'images CUPID (Conversation-mode Utilities for Processing Image Data) est utilisé pour réaliser certaines opérations de traitement d'images ainsi que leur affichage. Le langage de commandes propre à CUPID a été intégré dans le langage de commandes du système.

2.1.1.5. Le système SIRENE.

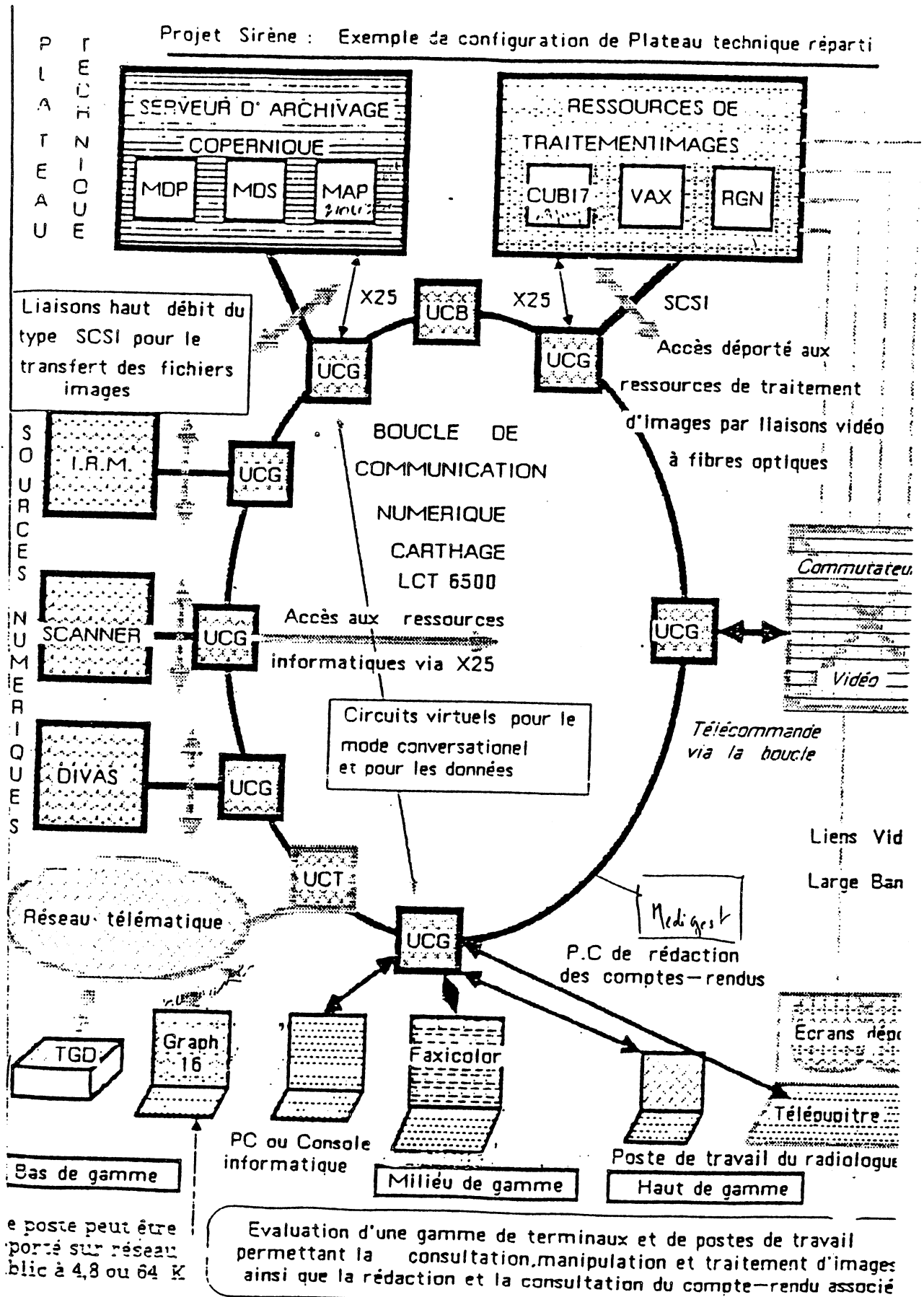
Serveur d'Imagerie médicale accessible via un REseau Numérique Experimental. [SCAR 83].

SIRENE est un projet d'imagerie médicale mené ensemble par le CHR de Rennes, le Centre Commun d'Etudes de Télédiffusion et Télécommunications (CCETT), l'Ecole Supérieure d'Electricité (ESE), l'université de Rennes 1, ainsi que des industriels associés (CGR, COPERNIQUE, LCT, TELEMAT, TITN, et XCOM). Les objectifs principaux du projet sont:

- démontrer l'intérêt médical et économique d'un système d'archivage et distribution numérique d'images médicales,
- utiliser des briques technologiques existantes ou en cours de développement, en vue de la réalisation d'un "PACS" (Picture Archiving and Communication System).

Le serveur d'imagerie médicale comprend l'archivage et la transmission des images, des textes associés et de la voix, ainsi que la manipulation simultanée de ces données. Le système est articulé autour de 6 composantes essentielles (la figure 2.1 représente l'architecture du projet SIRENE) :

figure 2.1



1) Les sources de diagnostic, fournissant des images numériques, à savoir un Scanner corps entier, un système d'Imagerie par Résonance Magnétique (IRM) et un Angiographe Numérique par soustraction type DIVAS.

2) Un serveur d'archivage. Le serveur 32, qui est une machine de gestion de bases de données développée par COPERNIQUE [COPERN], assure en temps réel le rangement et la restitution d'informations telles que le texte, l'image, et la parole. Il permet de stocker non seulement les images, mais aussi les autres éléments du dossier du patient, à savoir les comptes rendus d'examens radiologiques et autres documents. Le serveur remplit aussi des fonctions pour satisfaire des recherches multicritères particulières. Le serveur 32 est constitué de trois machines communiquant par un bus intermachine:

- * la machine primaire, qui gère les disques magnétiques ou les disques optiques numériques sur lesquels sont enregistrés les objets primaires tels que les images, les textes et les sons. Les images et commentaires associés appartenant à un même document ou dossier sont regroupés dans une "séquence" formant un ensemble adressable qui peut être créé ou consulté à partir d'un poste de travail connecté au réseau local. Ainsi, une "séquence" est l'ensemble des images et des paragraphes constituant une unité de sélection. Un "article de présentation" permet de localiser dans la "séquence" les différents objets primaires. Cette machine possède des éléments d'accélération de la lecture de disques et, grâce à sa modularité, plusieurs disques optiques ou magnétiques peuvent être placés ensemble,

- * la machine secondaire, elle supporte le système de gestion de base de données qui peut être de type réseau ou de type relationnel ou les deux simultanément. A chaque "séquence" de la machine primaire est associé un ensemble d'informations rangées dans la machine secondaire; ainsi, cette machine est chargée de la recherche des "articles de présentation" ou descripteurs d'objets primaires répondant aux requêtes multicritères. La sélection de "séquences" est assurée par un langage interactif de consultation qui peut être utilisé en temps réel ou pour la réalisation de programmes d'exploitation. Cette machine possède également un système de messagerie,

- * la machine frontale, elle assure la connexion de l'ensemble du serveur multimédia au réseau local, en dirigeant les informations vers le mode de transfert approprié.

Si la taille de l'application est telle que la machine secondaire est insuffisante pour tenir la charge, alors il est possible d'ajouter au serveur une machine d'application. Une telle machine, intégrée au serveur, assure les fonctions du dialogue d'accès avec les usagers, ainsi que les fonctions de messagerie permettant les transferts de dossiers entre radiologues et cliniciens.

3) Une ressource de traitement d'images constituée d'un VAX 750 qui joue le rôle de calculateur hôte par rapport aux deux machines dédiées au traitement d'images en 2 et 3 dimensions. Cette ressource est accessible depuis des postes de travail répartis autour du réseau. Un commutateur vidéo permet le déport des ressources des traitements sophistiqués sur des moniteurs vidéo connectés via des liaisons "Large Bande". Les fonctions de télécommande sont assurées par un télépupitre.

4) Une gamme de terminaux ou de stations de travail parmi lesquels se trouvent :

- * la station de travail haut gamme qui permet d'établir des diagnostics à l'aide des outils de traitement d'images et de rédiger les comptes rendus d'exams radiologiques en associant le texte à l'image,

- * le terminal alphaphotographique qui permet la consultation à distance d'images comprimées ainsi que des comptes rendus associés,

- * la station de consultation du dossier des patients en mode mixte (texte + image) qui permet la consultation, la manipulation et le traitement d'images conventionnels.

5) Un réseau multiservice qui connecte entre eux les modules déjà mentionnés. Il est également connecté au réseau informatique du système d'information de l'hôpital de Rennes. Ceci permet la création des dossiers électroniques des patients, archivables et consultables à distance. En plus, ce réseau est aussi connecté au réseau public Transpac.

6) Un logiciel intégrateur réunit l'ensemble des machines et des applications au-delà des mécanismes de communication pour offrir aux usagers, grâce à des menus appropriés, une vision globale du système d'informations hospitalières incluant les aspects images du serveur SIRENE.

L'intégration des systèmes de traitement d'images permet de réaliser des opérations très sophistiquées sur les images. Ainsi, en plus de la sélection et l'affichage des images, il est possible de choisir la résolution spatiale et la profondeur de l'image et obtenir des fenêtres de contraste ou de "zoom" sur une

partie de l'image. Il est également possible de visualiser plusieurs images (ou portions d'images) afin de les comparer, voir de composer plusieurs images mixtes. Un certain nombre d'évaluations quantitatives est aussi possible, par exemple, le calcul de surfaces, de distances et de volumes. Des traitements plus sophistiqués peuvent aussi être envisagés tels que les convolutions, la soustraction d'images, l'analyse d'histogrammes, etc.

2.1.1.6. Un système de gestion d'images radiologiques.

[TORI 80].

Le système a été développé à l'université de Nagoya (Japon). Son objectif est de gérer une collection d'images radiologiques préparées pour évaluer la performance du système d'interprétation d'images AISCN_V3. L'originalité de ce système est l'utilisation du "sketch" comme moyen d'accès aux images radiologiques. Un "sketch" consiste en un dessin du contour des organes et autres éléments qui sont suspects d'être anormaux dans une image radiologique. La plus grande partie du "sketch" est faite automatiquement par un système de traitement d'images, et il est complété par les médecins. Le système est composé de trois grands fichiers :

- * le fichier "master": il est formé par une collection de fichiers contenant les images digitalisées qui sont stockées sur une bande magnétique. Chaque fichier a un en-tête qui décrit les images qu'il contient. En plus des données proprement dites de chaque image, il y a un enregistrement qui contient des informations plus spécifiques concernant l'image qui suit,
- * le fichier "directory": contient les "sketchs" et les informations générales et spécifiques des images. Il est stocké sur disque magnétique et, grâce à ce fichier, on peut obtenir le nom du fichier d'une certaine image, les caractéristiques de l'image ou bien des données concernant le patient.
- * le fichier "temporaire": c'est un fichier de travail qui contient, à un moment donné, un sous-ensemble du fichier "master".

La manipulation de ces fichiers est faite par un ensemble de programmes écrits en FORTRAN qui permet de réaliser des opérations telles que:

- l'identification des "sketchs" qui satisfont certaines caractéristiques,
- l'interprétation des descriptions contenues dans les "sketchs", et l'affichage des images qui correspondent à cette description,
- la récupération des "sketchs" qui sont similaires à un "sketch" tracé ou à un "sketch" donné.

Note: chaque fois qu'un "sketch" a été sélectionné, l'affichage de l'image est possible si l'utilisateur le désire.

2.1.2. Les applications géographiques.

Les SGBD géographiques aussi appelés SGBD cartographiques, sont orientés vers la gestion d'images, la gestion de cartes et les fonctions de traitement de l'espace. Les principaux constituants des objets traités par ces SGBD sont les points, les lignes, les segments et les polygones. A partir de ces objets, on construit des objets plus complexes (ex. une route, une ville, une rivière). La base de données contient l'information structurée qui décrit l'objet (souvent les données représentant ses coordonnées) et, dans certains cas, des opérations arithmétiques et géométriques sont effectuées sur ces données. Nous citons quelques opérations à titre d'exemple:

étant donné un polygone et un point :

- est-ce que le point est à l'intérieur du polygone ?

étant donné un ensemble de polygones dont l'un contient un point :

- quel est le polygone qui contient le point ?

étant donné un ensemble de polygones :

- quel est le polygone voisin d'un polygone donné ?

étant donné un ensemble de segments et un point :

- quel est le segment qui passe le plus près du point?

Ils offrent aussi d'autres opérations plus spécifiques, basées sur des associations entre les objets et qui sont plutôt des opérations du domaine de traitement d'images, telles que la superposition, la coloriage, la rotation, le zooming, etc, plus la sélection de régions, et des fonctions de génération, de gestion et de dessin de

lignes. De nos jours, on a grande tendance à coupler les systèmes géographiques avec les systèmes de traitement d'images, à cause des opérations mentionnées et aussi pour l'utilisation d'images de satellites pour tracer des cartes. Un exemple de cette tendance se trouve dans le système IBIS [ZOBR 80] décrit ci-après.

2.1.2.1. Le système IBIS.

(Image-Based Information System) [ZOBR 80].

Le système a été développé au laboratoire de "Jet Propulsion" de l'institut technologique de Pasadena, California (U.S.A.). Le but du système est d'établir une communication directe entre les informations obtenues par les capteurs et les processus qui réalisent les traitements de ces données. C'est-à-dire qu'il offre un outil pour faciliter le transfert de données aux processus qui en ont besoin. Deux facteurs deviennent cruciaux: la manipulation d'une grande quantité de données, et les opérations de traitement d'images. Le système IBIS a été couplé avec le système de traitement d'images VICAR, qui supporte les deux types de format d'images (matriciel et vectoriel). Dans le système IBIS, trois types de données sont utilisées: (1) les données tabulaires, (2) les données graphiques et (3) les données images; chaque type de données forme un fichier. Les données tabulaires et les données images sont générées à partir des données graphiques, grâce à un module formé par un ensemble de sous-routines écrites en FORTRAN. Ce module transforme les données graphiques (polygones) en images des régions ou images des surfaces digitalisées, en créant ainsi les données images, ensuite il réalise une série de mesures sur ces dernières et génère les données tabulaires. Les images créées sont reçues par le système VICAR qui fait une série de calculs sur elles et produit un fichier où les données sont stockées par colonne. Ce fichier sert d'interface entre les images et les données tabulaires utilisées pour générer des rapports.

2.1.2.2. Le système EORID.

(Earth Observation Relational Image Database) [ROGA 85].

Le système a été développé au Centre Scientifique IBM (France); il est orienté vers la gestion et le traitement d'images de satellite. L'objectif principal est de simplifier la gestion des images pour l'utilisateur et de lui permettre l'accès à partir de paramètres caractérisant son contenu. Le contenu de l'image est en grande partie défini par ses modalités d'obtention et en particulier par les traitements

effectués. On distingue cinq classes de données: (1) les données d'acquisition de l'image, elles se rapportent au mode physique de génération de l'image, (2) les données sur la géographie telles que les lieux, les zones, les polygones et les coordonnées des points, (3) les données sur les transformations géométriques, concernant le type du modèle (ex. polynome, grille, etc) et sa zone de validité. Elle permettront la déformation des images pour les rendre superposables ou bien pour effectuer la navigation sur les images, (4) les données sur les traitements effectués, elles donnent l'histoire des images qui ont été obtenues à partir de traitements effectués sur d'autres images, et (5) les données sur le stockage des images, elles sont utilisées pour retrouver l'image physique.

Le formalisme du "modèle individuel", défini par H.Tardieu, qui utilise deux éléments structurants, "l'individu" et "la relation", a été adopté pour modéliser les données. Ce formalisme est très proche du modèle "Entité-Relation"; un "individu" (entité) définit un objet stable que l'on désire gérer, tandis qu'une "relation" (association) définit une association entre les "individus". L'implantation de ce modèle a été faite en utilisant le SGBD relationnel SQL/DS. Un ensemble de procédures qui définissent un environnement de traitement d'images et d'accès à la base, facilitent la gestion des images à l'intérieur des programmes. La base de données contient une structure liant les données et les occurrences des traitements, qui sont les résultats de programmes de transformation géométrique (et d'autres). Les images ainsi produites à chacune des étapes sont gardées dans la base et représentent les images virtuelles (images connues de la base de données, mais qui n'ont pas de support physique, parce qu'elles sont le résultat de traitements sur d'autres images). Toutes les images créées ou utilisées par un traitement sont enregistrées dans la base, avec les caractéristiques du traitement en question.

Etant donné qu'une image est essentiellement une matrice de données, [ROGA 85] propose d'introduire les images dans la base de données, au lieu de les gérer comme des fichiers externes. Ceci implique des extensions dans le langage de manipulation, et il propose l'introduction d'un opérateur de type "ARRAY". Ce type devrait permettre la définition de vecteurs et de matrices bi, ou tri dimensionnelles. L'accès aux données se fait à travers des langages de programmations tels que PL/I, COBOL, Assembler, FORTRAN, et APL2, ou bien d'une manière interactive grâce aux langages non-procéduraux comme ISQL ou QMF.

2.1.2.3. Le système IMAID.

(Integrated Image Analysis and Image Database Management System)
[CHNG 80], [CHNG 81].

Le système a été développé à l'université de Purdue (U.S.A.); il est composé d'un SGBD relationnel (REDI) et d'un système d'analyse d'images. Le système de traitement d'images permet d'extraire les caractéristiques de l'image (coordonnées, centre, etc), qui seront stockées dans la base et gérées par REDI. Les images physiques sont stockées hors de la base de données. Les requêtes concernant les images se font à partir des caractéristiques extraites. Si cette information n'est pas suffisante pour répondre aux questions de l'utilisateur, alors on sélectionne toutes les images qui satisfont un certain critère et on les envoie au système de traitement, afin d'en extraire d'autres caractéristiques qui viendront enrichir la base. Pour extraire les descriptions des images, un ensemble de fonctions élémentaires est construit (ex. mesure de texture, filtrage, segmentation, détection des bords, etc).

La manipulation de données se fait à travers le langage QPE (Query-by Pictorial-Exemple) qui est un langage base sur QBE, mais avec les extensions suivantes :

* trois entités graphiques ont été définies où l'unité de base est le segment:

point = segment avec coïncidence de ses deux extrémités,

ligne = ensemble de segments,

région = ensemble de segments groupés de façon à former les limites d'une surface.

* des opérateurs graphiques ont été introduits; ils utilisent comme argument deux unités graphiques qui donnent comme résultat une autre entité graphique :

intersection,

union,

négation,

soustraction,

constructeur de lignes et constructeur des points,

extracteur de lignes.

* des opérations géométriques réalisées sur un ensemble de segments ont été définis :

longueur,

périmètre,
surface.

* d'autres opérations orientées vers la manipulation d'objets graphiques ont été définies :

- sketch : trace le dessin d'un objet,
- set : fait la conversion d'un dessin en segments,
- sim : effectue des opérations de similarité dans le sens de valeurs d'attributs les plus proches.

Ce langage permet de formuler des questions du type:

- donner les noms de routes contenues dans le même plan que la ville X,
- donner la portion de l'autoroute A6 qui est incluse dans la ville X,
- donner le polygone formé par la ville X,
- etc.

2.1.2.4. Le système EIDES.

(ETL Image Database for Experimental Studies) [TAMU 80].

Le système a été développé au laboratoire Ibaragi (Japon); son objectif est de construire une base de données qui intègre plusieurs images ayant le format standard SIDBA (Standar Image Data-Base), afin de servir comme support de recherche dans des méthodes de traitement d'images et de reconnaissance de formes. Il est conçu pour réaliser des modifications, des insertions et l'affichage des images et des données associées. Les images sont identifiées dans la base de données par leur nom, le code du disque et par le numéro de séquence dans le fichier où elles sont stockées.

Les outils que le système offre à l'utilisateur peuvent être résumés comme suit:

- 1.- sous-routines d'accès aux fichiers images : l'accès aux fichiers SFDI est effectué à travers un ensemble de sous-routines écrites en FORTRAN, qui font la lecture ou l'écriture de l'image, ou d'une partie de l'image ("frame"). Elles utilisent l'information contenue dans l'en-tête de chaque fichier pour travailler avec l'image signalée.

2.- Transformation ou standardisation du format pour le traitement d'images : deux systèmes de traitement d'images sont utilisés (APAX et 2DDS) pour effectuer les opérations que les utilisateurs appliquent sur les images. Cependant, ces deux systèmes ont leur propre format d'images, qui sont différents du format de fichiers SFDI. A ce propos, quatre modules ont été construits qui servent à transformer le format de l'image dans le sens SFDI-->système de traitement d'images, et vice-versa.

3.- système interactif pour la manipulation de fichiers CREID: ensemble de cinq sous-systèmes, chacun composé d'une série de commandes :

- * sous-système HEADER, utilisé pour la gestion et l'édition des en-têtes d'images, où est concentrée toute l'information concernant le stockage des images,

- * sous-système FEDIT, utilisé pour l'édition de fichier image, cette édition étant faite par "frames",

- * sous-système FALTER, utilisé pour changer le format d'un "subframe" ou pour réordonner les pixels comme conséquence d'un changement dans l'en-tête de l'image,

- * sous-système PALTER, utilisé pour connaître ou pour changer la valeur d'un pixel dans une petite région,

- * sous-système PLOT, utilisé pour connaître les propriétés des images, pour tracer l'histogramme des images, ou bien pour éditer l'image avec différentes tonalités de gris.

4.- gestion de mémoire pour travailler avec de grandes images, comprenant des algorithmes pour le traitement d'images et des primitives pour partitionner les grandes images quand elles ne peuvent pas tenir en mémoire principale (gestion d'échanges entre la mémoire principale et le stockage des images).

2.1.3. Les applications plus générales.

Dans ce groupe, nous avons classé les systèmes qui s'orientent vers la bureautique, ou bien ceux qui n'ont pas pris une orientation spécifique. En d'autres termes, les systèmes classés dans ce groupe offrent un outil pour gérer les images sans prendre en compte une application précise. Normalement, dans ce type de système, le terme image a une signification plus générale, elle est vue comme étant une projection bi-dimensionnelle d'une scène tridimensionnelle, cette dernière

étant un ensemble d'objets physiques. Ainsi, une image comprend des objets numérisés tels que : les photographies, les radiographies, les cartes géographiques, les graphiques, etc. De façon simpliste, une image digitalisée peut être considérée comme une matrice de points (ou pixels) codés chacun sur un certain nombre de bits.

2.1.3.1. Le système MMS.

(Message Management System) [ECON 83].

Le système a été développé dans le laboratoire d'informatique de l'université de Toronto (Canada), il est orienté vers des applications bureautiques parmi lesquelles on peut citer: organisation de collections dans les galeries d'art et musées, contrôle des archives de police (photographies de criminels, photographies de scènes concernant un certain cas, etc). MMS est fait pour gérer des images de complexité et de diversité variée, il prend en compte la sémantique de l'image pour réaliser sa gestion. Une image est considérée comme un ensemble d'objets (qui ne sont autre chose que les éléments qui la composent) et des relations entre les objets. Par exemple, PERSONNE est une image composée des éléments *tête, tronc et jambes*. Un objet peut être composé d'autres objets et ainsi successivement, jusqu'à arriver à un niveau primitif qui est déterminé par l'utilisateur. Par exemple, l'objet *tête* qui est un élément de PERSONNE, peut être décomposé en d'autres objets tels que: *nez, yeux, bouche*. Cette décomposition peut être représentée dans un hypergraphe, dans lequel les nœuds correspondent aux objets et les liens correspondent aux relations entre les objets.

Divers termes sont introduits et interviennent dans la définition du schéma de la base de données:

type d'image.- est une entité générique qui représente les modèles des images dans lesquelles existent des objets (extraits d'un ensemble de types d'objets) et des relations entre les objets (extraites d'un ensemble de types de relations). L'information concernant un type d'image est donnée par la description de l'hypergraphe de l'image.

instanciation d'image.- est une instantiation d'un type d'image représentée physiquement par une collection ordonnée de pixels.

type d'objet. - est une entité générique identifiée par le nom du type d'objet. Il représente tous les objets qui ont des propriétés similaires. Chaque type d'objet a des attributs associés qui le caractérisent.

instanciation d'objet. - est un objet spécifique appartenant à un type d'objet. Il est identifié d'une manière unique par un identificateur et par le type de l'objet.

type de relation. - est une entité générique qui fait une abstraction de l'espace perceptif (e.g. en face, derrière, etc) entre les objets.

instanciation de relation. - est considérée comme étant une paire ordonnée d'instances d'objets qui ont une relation entre eux.

Le schéma de la base contient ainsi toute l'information concernant les types d'images, les types d'objets et les types de relations permises entre les objets. La base de données représente les instanciations de chacun des types décrits.

La manipulation de la base de données (interrogation, définition de nouveaux types objets et définition de nouveaux types de relations) se fait à travers trois types de formulaires, un formulaire étant un moyen pour balayer les données de la base, à partir d'un document qui a un format fixe et qui est rempli par l'utilisateur avec les données qui concernent sa requête.

1.- Formulaire Spécifique d'Objets (OSF).- un formulaire est associé à chaque type d'objet; il peut être utilisé pour définir un nouveau type d'objet ou bien pour interroger la base, afin de récupérer les types d'objets qui correspondent aux attributs signalés ou à une icône donnée. Il comprend les éléments suivants:

- nom,
- attribut1, attribut2, etc,
- décomposé-en,
- partie-de,
- icône.

2.- Formulaire Spécifique de Relations (RSF).- un formulaire est associé à chaque type de relation ou d'instance de relation, il est utilisé pour spécifier les relations entre deux objets. Il comprend les éléments suivants:

- nom,
- deux zones d'affichage des icônes participant à la relation,
- le nom des deux icônes.

3.- Formulaire Spécifique d'images (ISF).- un formulaire est associé à chaque type d'image ou instanciation d'image. Si le formulaire est vide, alors il est utilisé pour définir un nouveau type d'image; dans le cas contraire, il correspond à une instanciation d'image et il contiendra les instanciations des objets et des relations qui composent l'image. Il comprend les éléments suivants:

- nom du type d'image,
- une zone de définition de type d'objets (nom, nombre minimum et maximum),
- une zone de description de type relations (nom, objets participants).

Si l'on veut définir un type d'image, la zone d'objets contient une entrée pour chaque type d'objet et pour chaque type de relation contenu dans l'image.

D'autres commandes telles que COPY, EDIT, RETRIEVE, et MOVE sont utilisées pour manipuler l'information contenue dans les formulaires mentionnés.

2.1.3.2. Le système MINOS.

Références : [CHRI 86a], [CHRI 86b], [CHRI 86c].

MINOS est un système d'information multimédia orienté objet, développé conjointement par les universités canadiennes de Waterloo et du Maryland. Il exploite les possibilités des stations de travail modernes pour offrir à l'utilisateur les facilités pour créer et manipuler les objets multimédia et éventuellement complexes. Le système fournit une structure de document multimédia qui est assez riche et puissante, utilisable pour un grand nombre d'applications ayant ce type de besoins, par exemple: les systèmes d'information médicale, les systèmes de documentation technique, les bases de données graphiques, les systèmes de bureautique, etc.

Un document multimédia peut être composé de zéro ou plusieurs attributs, d'images, de sections de voix, de sections de texte et d'annotations. Un document est affiché par pages, où une page est composée par l'ensemble des informations qui est présentée en même temps.

Les attributs ont un nom et un type de données associé comme dans les systèmes de bases de données classiques. Les images peuvent être des objets assez complexes, par exemple, elles peuvent être des images sous-forme matricielle ("raster") ou sous-forme vectorielle (images graphiques ou statistiques). Les sections de voix sont formées par zéro ou plusieurs segments de voix ou narrations. Les segments de voix

sont associés aux objets tels que les paragraphes et les images et ils seront diffusés quand les utilisateurs en feront la sélection. Les narrations sont des segments de voix qui sont associés à une page de présentation et ils sont diffusés automatiquement quand la page est affichée. En ce qui concerne les sections de texte, chacune est formée d'un titre, un résumé, zéro ou plusieurs chapitres, et une liste de références. Finalement, les annotations établissent des relations entre différents documents multimédia.

Les fonctions offertes par MINOS sur les documents comprennent les fonctions de présentation, l'extraction de l'information, le partage de l'information, les fonctions d'édition et le formatage de documents.

Les fonctions de présentation représentent les facilités fournies par MINOS pour afficher et manipuler l'information récupérée, elles consistent en :

- * **BROWSING** .- déplacement à travers les pages d'un document. Divers possibilités sont offertes, par exemple : *aller à la page suivante, aller à la page numéro n, aller à l'objet X suivant, aller à la première page contenant l'objet X, etc.*
- * **ZOOMING** .- agrandissement d'une zone sélectionnée (à l'aide d'un souris) dans une page. La zone peut être partie d'une image ou d'une section textuelle.
- * **NARRATION** .- activation ou désactivation de l'exécution automatique du segment de voix qui est associé à la page.
- * **TRANSPARENCES** .- superposition de pages d'un document. Ainsi, quand la page suivante arrive, elle est placée sur l'information qui existait déjà sur l'écran. Ceci est très utile dans les applications médicales, par exemple: dans l'observation d'une série d'images radiologiques, on peut faire la superposition progressive des clichés, pour vérifier différents aspects de l'image.
- * **SIMULATION DE PROCESSUS** .- consiste à faire tourner automatiquement un certain nombre de pages du document. De cette façon, l'impression de mouvement ou de continuité est donnée.
- * **SEGMENTS DE VOIX** .- si un segment de voix a été associé à l'information de la page, alors un indicateur de voix est affiché. Pour diffuser le segment, l'utilisateur sélectionne cet indicateur à l'aide d'une souris.

* **ANNOTATIONS** .- sont des explications, des notes ou des descriptions détaillées qui sont associées à l'information. Elles représentent des liens avec d'autres documents. Quand l'annotation est sélectionnée, le document associé devient le document courant.

* **TEXTE LOGIQUE** .- C'est une information textuelle qui est associée à une ou plusieurs images. Elle s'affiche dans la même page d'affichage de l'image. C'est très utile aussi en médecine, où les images radiologiques peuvent être accompagnées des commentaires du radiologue.

* **VERSIONS** .- un document peut avoir plusieurs versions qui peuvent former une hiérarchie. La dernière version du document est la racine de la hiérarchie. Un mécanisme permet d'accéder aux différentes versions.

Extraction et partage de l'information.

Une interface permet d'extraire tout ou partie d'un document multimédia. Cette information peut être imprimée, envoyée par courrier électronique ou utilisée pour créer un nouveau document. Dans ce dernier cas, si l'information n'est pas modifiée, alors elle sera dupliquée quand le nouveau document sera stocké dans la base de données. Le système offre donc un mécanisme pour déterminer les cas où l'information est dupliquée ou partagée. Le plus souvent, les objets volumineux, qui ne changent pas, sont donc partagés par les documents.

Fonctions d'édition.

Trois éditeurs sont disponibles pour créer et manipuler l'information interactivement. Un éditeur de textes qui permet de générer ou d'extraire l'information d'un document et la modifier. Un éditeur de voix qui permet de générer, supprimer, extraire et concaténer la voix des segments. Et finalement un éditeur d'images qui permet de générer, extraire, afficher et transformer les images. C'est dans cet éditeur que l'utilisateur pourra associer aux images les annotations et le texte logique.

Formatage de documents.

Le formatage d'un document est interactif et déclaratif, c'est-à-dire que l'utilisateur peut spécifier le format de son document en utilisant des commandes de haut niveau et, grâce aux miniatures, il peut tout de suite apprécier les résultats.

Une miniature est une réduction d'un document qui s'affiche dans un coin de l'écran au moment du formatage.

Un prototype du système comprenant toutes les fonctions décrites est opérationnel. L'implémentation a été faite sur des stations de travail SUN-3 qui sont liées entre elles par un réseau. Actuellement, les données sont stockées sur des disques magnétiques, mais des études sont en cours pour utiliser le disque optique [CHRI 86c].

2.1.3.3. Le système IDMS.

(Integrated Database Management System) [TANG 81].

Le système a été développé à l'université de New York, Buffalo (U.S.A), il a une orientation générale, parce qu'il est capable de travailler avec n'importe quelle image numérisée (ex. photographie, radiographie, signature, carte géographique, etc). L'une des caractéristiques principales du système, est qu'il traite également les données alphanumériques et les données "images". Pour cela, il fait une extension au modèle relationnel, en ajoutant deux nouveaux types de données, à savoir le type "picture" et le type "device". Les attributs ayant comme domaine le type "picture", contiennent la matrice de pixels de l'image. Ils sont définis par trois paramètres m , n , b , où $m \times n$ définit la taille de la matrice, et b définit le nombre de bits nécessaires pour stocker chaque élément de la matrice. Des opérations d'égalité, de différence, d'inclusion, et de négation sont possibles entre des attributs de type "picture" en faisant une comparaison entre les pixels de chaque matrice participant à l'opération. De son côté, le type "device" sert à spécifier le type d'outil nécessaire pour afficher une donnée déterminée. Un attribut ayant comme domaine le type "device" prend ses valeurs parmi les entrées et sorties reconnues par le système. Le but du type "device" est d'offrir plus de simplicité à la manipulation des données en assurant une compatibilité entre les données et les entrées et sorties du système.

L'interrogation est faite à travers le langage SEQUEL. Celui-ci a été modifié afin de permettre la manipulation de nouveaux types de données et les opérations réalisées au niveau des matrices. D'autres extensions ont été faites au langage, afin de faire face aux problèmes de stockage, surtout en ce qui concerne les données de type "picture". L'utilisation des méthodes de compression des données et de migration de données (affectation précise du stockage de données, en indiquant le dispositif qui doit recevoir la donnée) est offerte. Pour cela, trois clauses ont été

introduites: STORE PICTURE, STORE RELATION et CODE OF IS. STORE est utilisé pour indiquer le dispositif qui va stocker les données, et CODE indique la méthode pour coder les données avant de les stocker.

2.1.3.4. Le système ELF.

(Extended relational model for Large, Flexible picture database) [YAMA 80].

Le système a été développé à l'université de Tokyo (Japon). Dans le système ELF, les images sont considérées comme un ensemble d'objets contenant une structure sémantique, qui est la description des relations entre les objets qui composent une image. Leur objectif principal est de traiter les structures sémantiques des images comme étant des faits ou des occurrences d'une base de données. Les objets ont des propriétés qui sont dérivées de la connaissance physique de l'objet ou de l'image même (ex. mesures, nombre, etc); ils sont classés dans des unités logiques qui représentent le monde réel et forment les classes génériques appelées aussi ensemble d'objets. Par exemple, *Jack, Pierre, Anne, etc* sont des objets appartenant à la classe générique *humain*. Le processus de description d'une image consiste à associer ces éléments (objets) avec des classes génériques et à déterminer les relations qui existent entre eux. Les structures sémantiques contiennent une description des relations qui sont valables entre les objets. Le modèle d'images consiste donc en une énumération de l'ensemble d'objets, de l'ensemble de relations, et la définition des cadres sémantiques dans lesquels l'ensemble des relations est valide.

Le modèle relationnel sert de base pour représenter le modèle d'images décrit. Deux types de relations sont utilisés pour établir une correspondance entre ces deux modèles.

(1) Les relations OBJETS, qui décrivent les propriétés des objets qui composent les images. Le nom de la relation a un rapport direct avec l'objet en question. Un identificateur est affecté à chaque tuple dans chaque relation, pour identifier l'image à laquelle appartient l'objet.

(2) Les relations ASSOCIATIONS, qui décrivent les relations qui existent entre les objets. Il existe une relation ASSOCIATION pour chaque ensemble de relations défini.

La structure sémantique d'objets est exprimée à travers des contraintes d'intégrité sur ces deux types de relations.

Une originalité du système ELF est de permettre que le nom d'une relation fasse partie du domaine spécifique d'un attribut. Ceci implique une extension du modèle relationnel, pour pouvoir accepter des noms de relations comme des valeurs d'attributs. En conséquence, le calcul relationnel doit être aussi étendu, afin de pouvoir manipuler les noms des relations indifféremment avec les autres valeurs de la base. L'interrogation se fait à travers le langage QBE-LIKE, qui est basé sur le langage QBE, avec les extensions mentionnées ci-dessus.

2.1.3.5. Le système ADM.

(Aggregate Data Manager) [TAKO 80].

Le système a été développé au Centre Scientifique IBM (Japon); l'idée principale est d'offrir un outil pour permettre une manipulation intégrée et homogène de données "images" et de données conventionnelles. Le modèle relationnel est utilisé en faisant une extension sur les types de données à gérer. Le type de données agrégat est introduit et il représente un terme général pour désigner toutes les données non-structurées (ex. images, graphiques, textes, ensembles etc.). Les données de type agrégat sont stockées d'abord dans un espace de travail et ils passent ensuite dans la base de données, en même temps qu'elles sont associées à des données atomiques (entiers ou caractères).

Pour le moment, le type agrégat concerne uniquement les images et les ensembles, mais sa portée peut être facilement étendue aux autres données non-structurées (ex. les textes, les graphiques, etc). Les images sont divisées en deux classes, les images binaires et les images avec des niveaux de gris. Pour sa part, un ensemble est considéré comme un groupe non ordonné de données atomiques qui servent à décrire les données de type agrégat (ex. un ensemble de mots utilisés pour décrire une image). Les images binaires sont comprimées avant d'être stockées, tandis que les images avec des niveau de gris sont stockées sans compression; ainsi, chaque tuple est représenté par une séquence de longueur variable de mots codés, plus un code numérique qui contient le numéro de tuple et le nombre d'octets occupés par les mots. En ce qui concerne les ensembles, ils sont stockés dans une relation binaire qui contient, dans chaque tuple, l'identification interne de l'ensemble et un élément de celui-ci.

ADM est composé de quatre sous-systèmes:

(1) interaction.- contrôle l'interaction homme-machine, analyse les commandes, contrôle l'affichage des données images et de données classiques, et déclenche le sous-système suivant,

(2) éditeur/processeur.- pour chaque type de données agrégat, il existe un éditeur qui, en plus de l'affichage des images, accomplit les fonctions de base de traitement des images, plus d'autres fonctions définies par l'utilisateur. Parmi les fonctions offertes par l'éditeur se trouvent : l'extraction, la suppression, la superposition, l'inversion, l'agrandissement, et la réduction d'images.

(3) espace de travail.- collection de données du type agrégat qui sont retenues temporairement pendant la session du travail soit pour des insertions ou pour des manipulations dans le cas où elles existaient déjà.

(4) base de données.- contrôle l'accès aux données, et constitue une interface pour accéder aux données atomiques et aux données de type agrégat, d'une manière intégrée. Du point de vue de l'utilisateur, la base de données est considérée comme une collection de tables et les données de type agrégat sont manipulées de la même façon que les types de données conventionnelles. Cependant, le SGBD, fait une gestion interne différente due à la différence de nature des données.

Pour l'interrogation de données, ADM offre deux possibilités. La première est une interface orientée commandes, classées en quatre groupes: (1) les commandes "scroll", utilisées pour lister les tables, ou pour choisir les images à afficher, (2) les commandes utilisées pour gérer un espace du travail contenant les données de type agrégat. Quand une donnée de ce type est concernée, le nom de l'espace de travail qui la contient doit être spécifié, (3) les commandes d'édition, qui sont appelées avec la commande EDIT suivie du nom de l'espace de travail qui contient l'objet à éditer. Le type de données (texte, graphique, image, etc), détermine la classe d'éditeur qui sera appelé. (4) les commandes de la base de données, qui sont celles du langage SEQUEL sans modification.

Les opérateurs de comparaison qui peuvent se trouver dans une requête sur les images sont l'égalité et la non-égalité; ils agissent tous les deux, sur l'identification interne de l'image. Pour ce qui concerne les ensembles, les opérateurs de comparaison sont :

égal

différente

contient complètement

ne contient pas complètement

contient une partie

ne contient aucune partie

est contenu par

n'est pas contenu par

Les requêtes SEQUEL contenant des opérations de comparaison sur les ensembles, ont besoin d'une transformation avant d'être exécutées, ceci à cause de la façon de stocker les ensembles dans la base.

La deuxième possibilité est une interface au travers de formulaires.

2.2. Etat de l'art des travaux réalisés sur les systèmes de gestion de comptes rendus.

La gestion de comptes rendus médicaux est plus liée aux aspects de recherche documentaire. Dans les systèmes de gestion de comptes rendus plus traditionnels, normalement leur archivage et leur récupération se fait manuellement à partir d'un numéro d'identification, qui peut être celui du malade, ou bien un code préétabli. Plusieurs problèmes sont attachés à ce type de méthodes, on trouve par exemple le cas présenté dans [SMIT 84] où un système de codage est utilisé. A chaque partie du corps et à chaque maladie correspond respectivement un numéro. L'ensemble de ces deux numéros forme le code du diagnostic. Un problème va se présenter quand plusieurs parties du corps ou plusieurs maladies sont concernées dans un même patient. D'abord, quel est le code que le médecin va assigner? Si un seul code lui est assigné, alors le cas ne sera pas récupéré quand la recherche concernera les autres aspects, ceci implique donc une perte d'information. Si, par contre, plusieurs codes sont assignés au même cas, on se trouve alors face à un problème de classification et de recherche, c'est-à-dire, lequel de ces deux codes sera utilisé pour archiver et retrouver l'information? Une autre option serait de dupliquer l'information autant de fois que de codes existants, et donner ensuite un code à chaque copie, mais ceci conduit finalement à une dispersion de l'information et à un gaspillage de temps, des efforts et du matériel.

De plus en plus, des systèmes informatisés de gestion de comptes rendus font leur apparition. Il existe même plusieurs systèmes qui se servent de méthodes de codage [SMIT 84], [LEMM 79], [GELL 76], mais leur utilisation est la plupart du temps automatique ou semi-automatique, et réglée par une grammaire ou une syntaxe interne. D'autres systèmes vont plus loin en essayant de faire une analyse du langage naturel (réduit au domaine médical) comme c'est le cas du système REMEDE [HEAU 80], [HEAU 79], qui fait une combinaison des méthodes de codage, et une manipulation du texte libre.

Même s'il y a différentes approches pour la gestion des comptes rendus, leurs objectifs se retrouvent le plus souvent identiques et comprennent généralement la génération automatique de comptes rendus, le contrôle de leur archivage et sa récupération à partir de différents critères et pour diverses fins. Nous présenterons par la suite une synthèse des différents travaux que nous avons trouvés intéressants et en rapport avec notre application spécifique.

2.2.1. Le système REMEDE

[HEAU 79], [HEAU 80], [BRAM 85], [DACH 82].

Une première version du système REMEDE a été développée en 1972 à l'hôpital Pitié Salpêtrière (FRANCE), par le Prof. M. de HEAULME; son but était la mise en place d'un système de traitement de résumés médicaux par ordinateur, ceci pour satisfaire trois familles de besoins se rapportant au traitement des comptes rendus et de résumés bio-médicaux [HEAU 80]:

- * les besoins des médecins en information quotidienne (ex. écriture simple de comptes rendus, consultation des informations figurant dans le dossier du malade, etc),
- * les besoins des secrétariats médicaux (ex. identification, classification et communication correcte de documents, etc),
- * les besoins liés au problème général de l'analyse d'activité (ex. analyse comptable, analyse du flux d'informations médicales, etc).

REMEDE a été conçu pour archiver sur ordinateur des résumés ou comptes rendus médicaux de tous types, pour les interroger en vue d'en extraire les données qui répondent aux besoins spécifiques de l'utilisateur. Il a pour principe général l'utilisation d'un langage médical contrôlé au double sens, lexical et syntaxique, qui

serve à l'indexation, au stockage et à l'interrogation de l'information [BRAM 85]. Ce langage utilise un vocabulaire qui est défini par spécialité, l'ensemble des termes médicaux contenus dans une spécialité formant un thésaurus, lequel est organisé en classes indépendantes telles que: symptômes, topographie, examens, maladies, traitement, etc. Le thésaurus comporte des termes possédant une organisation hiérarchique, par exemple: membre supérieur: *épaule/bras/coude/avant-bras/poignet/main/doigt*. Les classes sont transparentes pour l'utilisateur et le rattachement du mot à la classe est fait automatiquement par le thésaurus. Enfin, des commentaires libres peuvent accompagner chaque terme, pour compléter l'information importante qui aurait pu échapper à la description formelle.

Le but de REMEDE est de fabriquer des phrases descriptives. A cet effet, les mots sont reliés entre eux par des opérateurs spécifiques permettant de créer des relations non ambiguës entre ceux-ci. La combinatoire entre les termes et les opérateurs est grande, cependant il existe une syntaxe qui sert à limiter les possibilités, le contrôle étant effectué sur la nature des classes des mots co-présents dans la chaîne. Les principaux opérateurs utilisés par REMEDE sont [DACH 82]:

<----- implique une relation topographique.

Ex: douleur de genou

douleur <-----genou

-----> implique une relation causale.

Ex: douleur du genou due à l'arthrose

douleur<-----genou---->arthrose

= , + indiquent un résultat et un attribut respectivement.

Ex: calcémie augmentée

calcémie=+

* indique un traitement.

Ex: traitement anti-inflamatoire du genou contre l'arthrose

arthrose<-----genou * anti-inflammatoire

L'intention de REMEDE est d'utiliser l'éclatement des idéogrammes, jusqu'à l'obtention de concepts unidimensionnels. L'intérêt de cet éclatement est de fournir et de fabriquer des concepts complexes représentés non par un mot, mais par une phrase, et ceci à l'aide d'un nombre de mots limité. Il est à noter que la description de nouveaux concepts ne se fera qu'au prix d'une inflation du thésaurus.

Les résumés d'hospitalisation sont indexés en langage REMEDE par des responsables du service. Après validation de l'information, seuls les résumés définitivement acceptés sont stockés dans un fichier général dans le langage artificiel de REMEDE. L'interrogation s'effectue avec le même genre de "phrase". Elle consiste à analyser les textes codés en REMEDE, et à vérifier s'ils correspondent à la question posée. La recherche sur mot fait appel aux règles d'inclusion du thésaurus (ex. l'appel de "membre supérieur" fera chercher automatiquement les membres descendants), tandis que la recherche sur phrase permet de travailler sur les relations entre les mots. Les questions portent d'abord sur la totalité du fichier général, puis sur des sous-fichiers issus des questions précédentes selon un processus d'affinement "en cascade" [HEAU 80].

Bien que le système permette une absence d'ambiguïté dans la description de termes et qu'il offre de puissantes procédures de recherche documentaire, grâce à la possibilité de jouer sur la combinatoire mots-opérateurs, il présente aussi des inconvénients parmi lesquels on peut citer:

- la nécessité d'une initiation préalable du médecin qui utilise le système et des médecins susceptibles de recevoir un communiqué en texte REMEDE,
- l'absolue nécessité d'employer des mots contenus dans le thésaurus,
- la classification hiérarchique qui entraîne le problème de termes qui peuvent figurer dans plusieurs classifications simultanément.

Deux langages ont été dérivés du langage REMEDE, le langage LRFM et le langage INTERMED. Leur principe de base est le même que celui de REMEDE, mais il y a eu une évolution principalement au niveau de l'analyse sémantique.

2.2.2. Le langage LRFM.

Le langage de représentation des faits médicaux [DACH 82].

L'analyse d'une phrase médicale comprend deux étapes successives, qui ont pour but de transformer les concepts médicaux exprimés en langage source en un mode d'expression unique. La première étape consiste en une transformation sur une phrase entrée en langage quasi naturel (LQN) pour l'amener en une forme intermédiaire (FI) du LRFM. La seconde étape fait appel à des règles de transformation qui font intervenir des connaissances médicales, pour traduire la FI dans une forme canonique (FC). Cette dernière sera normalement une phrase complète, sans redondance ni ambiguïté. Considérons par exemple les phrases suivantes:

- hépatomégalie
- gros foie
- hypertrophie hépatique
- augmentation du volume du foie

Elles font toutes référence au concept "le foie a augmenté de volume"; ainsi, des phrases différentes, mais ayant un même contenu conceptuel, doivent aboutir à une expression unique qui exprime le concept sémantique. Les liaisons sémantiques des expressions dans les formes canoniques sont assurées par des connecteurs similaires aux opérateurs de REMEDE.

2.2.3. Le langage INTERMED

[BRAM 85].

C'est un langage permettant la réalisation et la communication avec une base de faits médicaux. Il est capable d'interpréter des phrases du langage naturel décrivant des faits médicaux, pour les utiliser ensuite dans l'indexation ou dans l'interrogation. Il est considéré comme une évolution du LRFM, et comprend aussi la présence de connecteurs entre les mots. Deux dictionnaires sont utilisés, l'un contenant tous les termes avec des inférences (dictionnaire de référence de base) et l'autre contenant des synonymes.

Les étapes qui permettent l'indexation et la recherche sont résumées ci-dessous.

* analyse morphologique et lexicale: le langage naturel subi un analyse morphologique, où il sera affecté automatiquement une classe sémantique à chaque terme repéré.

* interprétation des faits médicaux: on supprime les mots non pertinents et, ensuite, on utilise une grammaire sémantique pour interpréter les faits comme étant pathologiques ou normaux. La phrase peut suivre après d'autres transformations, par rapport à des règles préalablement définies.

* indexation: cette étape comprend une boucle qui commence par l'indexation de la phrase qui vient d'être générée, laquelle est ensuite décomposée en d'autres phrases plus courtes qui seront aussi indexées. La boucle fait appel aux inférences qui ont été définies au moment d'entrer un terme dans le dictionnaire de références; elles servent à utiliser d'autres termes pour indexer la même phrase. Par exemple, si l'on a "*cancer bronchique*", par les inférences définies, on peut générer "*tumeur maligne de la bronche*".

* recherche: cette étape comprend une série de transformations de la phrase générée par les étapes précédentes, afin de trouver toutes les phrases qui indexent la phrase d'origine.

INTERMED n'est pas seulement le moyen de fabriquer des bases intelligentes de faits médicaux, mais est aussi une interface entre le langage médical et d'autres systèmes. Ses applications sont donc multiples par exemple:

- * dossier médical,
- * accès à des systèmes experts,
- * etc.

2.2.4. Un système de gestion d'examens de tomодensitométrie destinés à l'enseignement.

[SMIT 84].

Un prototype pour gérer les cas intéressants en tomодensitométrie a été développé par le collège médical de l'université du Wisconsin et de l'université de Washington (U.S.A.). Leur but principal est de pouvoir accéder d'une manière

précise et rapide aux cas intéressants susceptibles d'être utilisés dans une conférence, dans une publication ou bien dans l'enseignement.

L'information est archivée dans des dossiers qui sont classés en accord avec un code anatomo-pathologique composé de quatre chiffres, basé sur les codes fixés par le Collège Américain de Radiologie pour le diagnostic radiologique. Ces codes sont assignés manuellement par le radiologue et représentent l'organe et la maladie du cas traité. Chaque dossier contient les données qui identifient le patient, la date de l'examen, le numéro d'examen, une brève description du cas, les constatations pertinentes, ainsi que les images radiologiques.

Toutes ces données (à l'exception des images) ainsi que d'autres données qui concernent principalement les caractéristiques de réalisation de l'examen (ex. type de scanner (tête ou corps), utilisation ou absence de produit de contraste, etc), sont introduites dans une base de données et gérées par un SGBD relationnel. Presque toutes les données stockées dans la base peuvent servir de critères de recherche, étant donné qu'elles sont des données atomiques. Cependant, la recherche d'une donnée anatomo-pathologique doit se faire en utilisant le code du diagnostic, la description et les constatations étant des chaînes de caractères sans indexation autre que les codes. Ceci implique que, pour arriver à des résultats satisfaisants, l'utilisateur doit être familiarisé avec cette codification. Par ailleurs, la technique de codification a des problèmes en soi, étant donné que les médecins peuvent assigner des codes différents pour des cas similaires. Dans ces circonstances, on rencontre beaucoup de silence ou d'ambiguïtés.

2.2.5. Le système CLIP.

(Coded Language Information Processing) [HEAU 80].

Le système CLIP a été développé par le département de Radiologie de l'école Médicale de Harvard (U.S.A.). Il est orienté vers la génération automatique de comptes rendus. Le système fait un codage des mots et des phrases contenus dans les comptes rendus, en utilisant la méthode de classification de Simon-Leeming, dans laquelle on classe le terme dans une structure hiérarchique et on lui assigne un code. Ainsi, CLIP contient à peu près 28000 termes médicaux, classés en trois sections principales: anatomie, étiologie et sémiologie. Chacune de ces sections est représentée au niveau de codage par un symbole où, " : " représente l'anatomie, " ; " représente l'étiologie, et " / " représente la sémiologie. Les sections sont divisées

en catégories, chacune identifiée par sa lettre initiale (voir figure 2.2), et chaque catégorie est divisée numériquement en disposition décimale, (ex. "A" est le code pour ANATOMIC AREA, "A6" est le code pour THE LOWER LIMB, et "A61" est le code pour THE THIGH). Par ailleurs, il existe une autre section qui contient des termes non médicaux utilisés pour construire les phrases complètes qui seront insérées dans les comptes rendus.

Pour rédiger un compte rendu, le médecin travaille avec cette classification; il choisit, parmi une liste qui lui est proposée à l'écran, les termes pertinents pour le cas à rédiger et, à chaque fois qu'un terme est sélectionné, il est ajouté automatiquement à la rédaction du compte rendu. En même temps, le code correspondant est concaténé à une chaîne de codes qui est associée au compte rendu et qui servira comme index du document.

Une première version du système était opérationnelle déjà en 1974 à l'hôpital Beth Israël de Boston, mais il était très peu utilisé, étant donné que la recherche de termes était très fatigante pour le médecin, surtout dans les cas de mots rares, où il était obligé de parcourir la liste pour constater à la fin que le mot n'existait pas et qu'il fallait donc l'introduire comme texte libre. Ainsi, les améliorations suivantes ont été rajoutées au système:

- * recherche automatique de mots à partir d'un nombre de lettres initiales; le système affiche une liste de mots commençant avec les lettres données par l'utilisateur et il peut s'apercevoir tout de suite si le terme cherché existe ou pas,
- * entrée rapide de termes ou de compte rendus complets; le système repère automatiquement chaque terme, fait le codage et la rédaction narrative du compte rendu sans l'intervention du médecin,
- * variations grammaticales telles que passage du pluriel au singulier ou vice-versa, utilisation de mots de négation (absence, aucune, etc), utilisation d'adjectifs, etc. Toutes les variations sont exprimées à l'aide de symboles spéciaux, par exemple le signe "# " change un mot de nombre (pluriel <---> singulier), le tiret " - " est utilisé pour indiquer la présence d'une négation,
- * pré-assemblage de phrases standards pour les utiliser dans les constatations qui se répètent souvent. Un nombre de phrases pré-établies est affichée, quand le code d'un organe ou d'une région sont introduits et le médecin fait son choix en tapant le numéro qui correspond à la phrase,

* suppression des termes non médicaux qui sont redondants; ceci s'applique aux comptes rendus qui sont rédigés en utilisant les phrases pré-établies. Ainsi, quand deux phrases ou plus vont être assemblées, les termes non médicaux répétitifs sont enlevés.

La même structure hiérarchique de classification de termes sert à la recherche d'information. Cette structure permet l'utilisation de synonymes, des formes adjectivales. Grâce à la subdivision de termes, un terme peut être récupéré sans être mentionné dans la requête de l'utilisateur, par exemple, le terme *carcinoma* sera codé comme " ; N21 " avec " ; " pour étiologie, " N " pour Neoplasme, " 2 " pour maligne, et " 1 " pour carcinoma. En cherchant le terme *tumeur maligne*, on pourra trouver *carcinoma*, *cancer malin*, *processus mitotic*, etc, parce que tous ces termes sont codés dans la même branche hiérarchique.

Système CLIP

figure 2.2

:ANATOMY		/FINDINGS		:ETIOLOGY	
A	ANATOMIC AREA	A	ALL NORMAL	A	ALL NORMAL
B	BONE	B	BREACH OF STRUCTURE	B	BACTERIAL OR OTHER BUG
C	CARDIAC	C	CALCULUS OR FOREIGN BODY	C	CONGENITAL OR DEVELOPMENTAL
D	DIGESTIVE	D	DEVITALIZED OR DEAD TISSUE. ULCER	D	DEGENERATIVE
E	ENDOCRINE	E	—EDGE CHARACTERISTIC	E	ENDOCRINE DISORDER
F	FUNCTIONAL BODY SPACE	F	FREE FLUID OR GAS	F	FEEDING DISORDER
G	GENITAL	G ⁺	—GRADE OF SEVERITY OR STAGE	G ⁺	—GRADE OF SEVERITY OR STAGE
H	HEMATIC	H [*]	HYDRODYNAMIC DISTURBANCE	H	HEMATOLOGIC DISORDER
I	INTEGUMENTARY & BREAST	I	INFILTRATION OR CONSOLIDATION	I	IMMUNOLOGIC OR INFLAMMATORY
J	JOINT	J	—JUDGE FINDINGS	J	—JUDGE DIAGNOSIS
K	— KEY PHRASE OR WORD	K	—KEY PHRASE OR WORD	K	—KEY PHRASE OR WORD
L	LYMPHATIC	L	LUMEN DEFORMITY	L	—LIBRARY, TEACHING, RESEARCH
M	MUSCULAR	M	MASS LESION	M	METABOLIC DISORDER
N	NERVOUS	N [*]	NEUROMUSCULAR DYSFUNCTION	N	NEOPLASTIC OR DYSPLASTIC
P	PREGNANCY	P	—POSITION OR DISPLACEMENT	P	PSYCHIATRIC
Q ⁺	—QUANTITY	Q ⁺	—QUANTITY: SIZE, NUMBER, RATE	Q ⁺	—QUANTITY
R	RESPIRATORY	R	REPAIR OR FIBROSIS	R	REACTION: DRUG, DUST, CHEMICAL
S	—SIDE SIGHT & SOUND	S	—SHAPE OR DEFORMATION	S	SURGICAL
T	TEETH	T	—TEXTURE OR TISSUE DERANGEMENT	T	TRAUMATIC
U	URINARY	U	—UPDATED FINDINGS	U	UNKNOWN, BUT DIAGNOSTIC
V	VASCULAR	V [*]	VISCERAL DYSFUNCTION	V	VASCULAR DISORDER
W	—WHEREABOUT?	W	—WHEREABOUT?	W	—WHEREABOUT?
X	—X-RAY PROCEDURE	X	—X-RAY SHADOW	X	—X-RAY SHADOW
Y	—YOUR PATIENT DATA	Y	—YOUR PATIENT DATA	Y	—YOUR PATIENT DATA
Z	—ZANY SHADOW	Z	—ZANY SHADOW: ARTIFACT, COMPOSITE	Z	—ZANY SHADOW

- + Exact numbers, measurements or staging may be coded in parenthesis.
- * Denotes functional as distinct from structural changes.
- Denotes a qualifying category.

The major categories of the ANATOMY, FINDINGS and ETIOLOGY sections of the Simon/Leeming classification are coded by easily memorized alphabetic initial letters.

2.2.6. Le système ROM.

(Radiology Operations Management), [AREN 79].

ROM est un système conçu pour gérer les activités du service de radiologie de l'hôpital de l'université de Pennsylvanie (U.S.A.). Il comprend plusieurs modules dont un concerne la génération de comptes rendus en permettant trois modes de rédaction.

Dans le premier mode, les comptes rendus sont tapés au terminal à l'aide d'un traitement de textes, les données concernant l'identification du patient sont automatiquement affichées, provenant du module associé à la réception du patient dans le service. Ce travail est normalement fait par des secrétaires.

Dans le second mode, c'est surtout le radiologue qui génère le compte rendu lui-même. Chaque radiologue possède un livret de comptes rendus types ayant des barres de codes associées, il sélectionne et fait lire, à l'aide d'un crayon optique, les codes qui correspondent à l'examen qu'il est en train de faire. Une fois que tous les codes nécessaires ont été introduits, le système rédige le compte rendu d'une manière complètement automatique, parce qu'aucune intervention n'est plus nécessaire. La possibilité d'introduire du texte libre n'est pas permise dans ce mode.

Dans le troisième mode, ce sont aussi les radiologues qui génèrent les comptes rendus; ils utilisent un outil spécial appelé "diagnostic reporting terminal", qui consiste essentiellement en trois écrans qui contiennent des mots, des phrases, et des dessins anatomiques avec lesquels le radiologue va construire le compte rendu. En touchant sur l'écran avec un crayon optique, il sélectionne le texte, les causes et les régions anatomiques qu'il veut insérer dans le compte rendu. Ce mode accepte l'entrée d'un texte libre.

Les compte rendus qui ont été générés sont accessibles en utilisant le nom du patient, le numéro de compte rendu, ou bien le numéro du dossier assigné par le département de radiologie. Des extensions sont envisagées pour utiliser le code du diagnostic du Collège Américain de Radiologie comme moyen d'indexation et de récupération de l'information, de façon analogue à l'utilisation des codes dans le système développé à l'université du Wisconsin [SMIT 84].

2.2.7. Le système RAPORT.

[SELT 77].

RAPORT est un système commercialisé pour générer automatiquement des comptes rendus radiographiques. La rédaction d'un compte rendu se fait en utilisant un formulaire de marquages sensitifs sur lequel sont contenus des symboles, des graphiques, et des termes médicaux (voir figure 2.3). Le radiologue désigne à l'aide d'un crayon optique les éléments qui correspondent à l'examen en question, ensuite le compte rendu est automatiquement rédigé en se servant des éléments touchés. A la fin de la rédaction, si le radiologue le désire, le compte rendu peut être complété avec des phrases tapées au clavier et considérées comme du texte libre.

Etant donné que le compte rendu est entièrement écrit par le système, on élimine les erreurs de frappe qui sont fréquentes dans les comptes rendus tapés à la main. De plus, un compte rendu plus précis est obtenu, parce que les phrases contradictoires ou ambiguës sont aussi éliminées. Cependant, l'utilisation de ce système implique que le radiologue apprenne à manipuler de nouveaux symboles et formulaires, ce dont il n'a pas en général l'habitude.

Les comptes rendus ne sont stockés et gérés par le système que pendant le temps de séjour du patient à l'hôpital. Des mécanismes de recherche de l'information ne sont pas mentionnés, à l'exception d'une liste produite par le système contenant des cas intéressants qui ont été signalés par le radiologue.

Système RAPORT

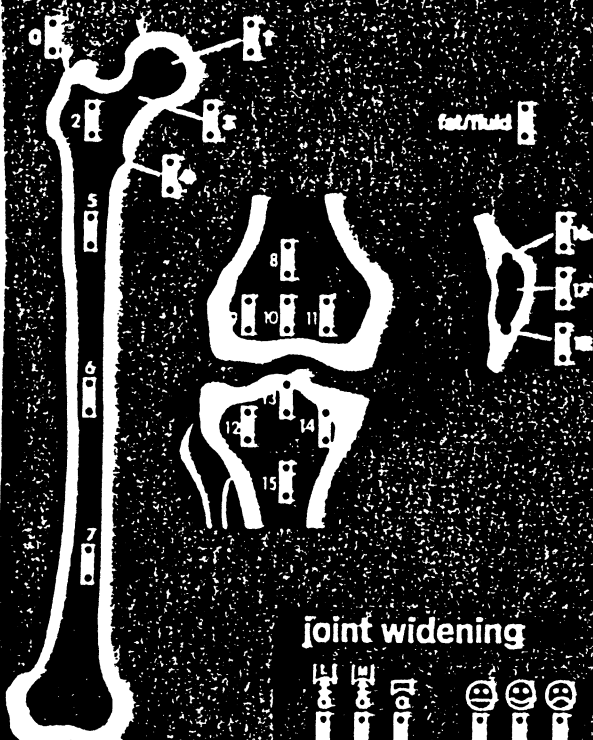
figure 2.3

femur-knee GENERAL ELECTRIC RADIOGRAPHIC REPORT SYSTEM **GE RAPORT™**

exam: R L femur knee patella AP lat obt stress wt. brg. port OR post reduc. cast: in out

day number: 0 1 2 3 4 5 6 7 8 9
 0 1 2 3 4 5 6 7 8 9
 0 1 2 3 4 5 6 7 8 9

comp. film date: month 1 2 3 4 5 6 7 8 9 10 11 12
 day 10 20 30 1 2 3 4 5 6 7 8 9
 year 65 66 67 68 69 70 71 72 73 74 N.A.



fat/muld

joint widening:

degenerative change:

joint narrowing: L M I II III

general: I II III

trauma 1

incidental:

fracture type:

position:

alignment:

healing:

trauma 2

incidental:

fracture type:

position:

alignment:

healing:

patella: bipart. post-surg. abserice frag.

effusion: none I II III

loose bodies

A P 1st

A P 2nd

osteocond. dissecans osgood-schlatter

calcification

meniscus:

jt. cart.:

osteocondromatosis fabella

fract. disloc.

osteo-myel. foreign body

N r/c O

R: 1 2 3 4 5 6 7 8 9

2.2.8. Le système JOHNS HOPKINS.

[WHEE 76].

Ce système a été développé par les départements de radiologie et de biochimie des institutions médicales Johns Hopkins, Baltimore (U.S.A.). Leur objectif principal est la rédaction automatique de comptes rendus, mais ils considèrent aussi leur stockage et leur accès.

Pour réaliser un compte rendu, l'utilisateur rentre d'abord toutes les données qui servent à identifier le patient; ensuite, il fait afficher sur l'écran deux types de "frames" ou cadres. Le premier cadre contient tous les mots constants utilisés dans le diagnostic (ex. mesures, degré de certitude, adjectifs, etc), ainsi que le lexique de l'anatomo-pathologie. En ce qui concerne le deuxième cadre, appelé "cadre principal", il existe un cadre associé pour chaque type d'examen. Généralement, il contient des termes et des phrases pathologiques, anatomiques et descriptives suffisantes pour rédiger la plupart de cas normaux. Les termes anatomiques sont présentés avec un diagramme permettant ainsi un repérage plus facile de la part des médecins. Des options pour afficher d'autres cadres existent dans les cas de la rédaction de comptes rendus plus complexes. A l'aide d'un crayon optique, le médecin choisit sa terminologie parmi les options qui lui sont présentées, les termes choisis sont automatiquement insérés dans le compte rendu en même temps qu'il sont codés dans un format de quatre chiffres. Les cas qui sont complètement nouveaux ou qui doivent être décrits avec un détail inhabituel, sont tapés directement dans le clavier.

Des programmes sont disponibles pour réaliser la recherche d'information, laquelle peut se faire sur différents critères, à savoir:

- type de maladie,
- section anatomique,
- caractéristiques du patient.

Les compte rendus entrés à l'aide du clavier, ne peuvent pas intervenir dans les conditions de recherche, autres que les caractéristiques du patient, parce que leur texte n'est pas indexé.

2.2.9. Le système AURA.

[GELL 76].

Le système a été développé à l'université d'Austin (U.S.A.). Son objectif était d'offrir aux médecins un outil qui leur permettrait un accès facile aux comptes rendus, sans pourtant modifier leurs habitudes de rédaction; il réalise donc une indexation du texte original du radiologue.

Le système comprend trois modules, à savoir:

- 1.- Saisie de données: cette étape comprend la rédaction du compte rendu à l'aide d'un éditeur de textes, dans les formes habituelles des médecins.
- 2.- Analyse et stockage: le compte rendu passe par deux étapes d'analyse, la première consiste à identifier les données administratives contenues dans le document telles que le nom du patient, la date de l'examen, etc. La seconde étape est plus complexe; en termes généraux, elle se charge de l'indexation du corps du compte rendu, qui comprend: le diagnostic du médecin demandeur, le type et la description de l'examen et finalement le diagnostic du radiologue. Chaque compte rendu est comparé à une liste de mots qui contient tous les termes médicaux utilisés dans le diagnostic radiologique, ainsi que des codes associés. Le but de ces codes est de pouvoir identifier les synonymes en permettant ainsi un accès à l'information par différentes facettes. Chaque mot du corps est comparé avec cette liste pour lui assigner un ou plusieurs codes, qui permettront de lui donner sa signification précise et sa position dans un système de classification.
- 3.- Récupération de l'information: les recherches se font dans la terminologie habituelle des médecins, qui peut être utilisée en combinaison avec les opérateurs logiques AND, OR, et NOT. Cependant, il est conseillé de ne pas utiliser de requêtes très restrictives quand l'on veut faire une recherche exhaustive, parce que l'on risque d'avoir beaucoup de silence (documents pertinents qui ne sont pas récupérés), ceci dû aux différents styles de diagnostic trouvés chez les médecins. On voit donc que les classifications hiérarchisées permettent de retrouver aisément les informations élémentaires, mais par contre on manque beaucoup de diagnostics lorsqu'on désire faire une recherche plus précise.

2.3. Discussion

Dans ce chapitre, nous avons surtout cherché à faire une présentation non exhaustive de systèmes de gestion des images et des comptes rendus associés. Tout d'abord, nous avons fait une séparation des systèmes qui gèrent ces deux objets, parce que la plupart d'entre eux se concentrent sur un objet ou sur un autre, et rares sont ceux qui font une intégration homogène des deux. Ensuite, pour ce qui concerne la gestion des images, nous avons voulu présenter le procédé pour des domaines autres que la médecine, pour essayer de montrer les niveaux de gestion que chacun demande.

Ainsi, nous avons pu constater qu'un système de gestion d'images unique pour toutes les applications est presque impossible, parce que chaque domaine a plus ou moins des besoins propres à son application. Nous pouvons le voir par exemple avec les opérations géométriques de construction de polygones à partir de lignes qui sont plutôt utilisables dans les applications géographiques, tandis que la détection de contours serait une opération plus spécifique au domaine médical. Ceci n'empêche pas qu'il existe des opérations qui peuvent être communes à toutes les applications, comme c'est le cas du zooming, du changement d'échelle, de la rotation. Parallèlement, les opérations de périmètre, surface, distance et mesures, trouvées souvent dans le domaine géographique, peuvent être fort utiles aux SGBD médicaux, qui, dans la plupart des cas, utilisent un système de traitement d'images et de programmes "ad-hoc" pour les réaliser.

Un autre aspect important que nous avons pu constater, est que le stockage des images est très lié au niveau de granularité avec lequel elles sont considérées dans l'application, c'est-à-dire, que l'image "physique" n'est stockée dans la base que lorsqu'il existe un intérêt à travailler directement avec les unités minimales de l'image (c.a.d. les pixels) ou des segments. Ce sont surtout les applications géographiques qui suivent cette tendance. Ceci n'est pas le cas pour d'autres applications telles que les applications médicales ou bureautiques, où l'on travaille surtout avec les images complètes, qui sont souvent stockées hors de la base. Et en ce qui concerne les dispositifs de stockage, ce sont les disques et les bandes magnétiques qui dominent. L'utilisation du disque optique numérique commence à être envisagée, mais aucun des systèmes présentés (à l'exception du système SIRENE), ne s'en sert d'une manière opérationnelle.

Tout paraît indiquer que l'informatisation de la gestion des images suppose des systèmes plus coûteux que ceux qui s'attachent à l'informatisation des comptes rendus, étant donné que la quantité de mémoire demandée par les premiers est plus importante et, en plus, leur manipulation demande parfois de créer des opérations propres au système.

Pour sa part, une grande partie des systèmes concernant l'informatisation de comptes rendus s'intéressent notamment à leur conception automatique; d'autres, moins nombreux s'intéressent à leur analyse et leur stockage et, ils sont encore moins nombreux, ceux qui s'orientent vers une intégration homogène des images et des comptes rendus. Ces systèmes seront considérablement améliorés si la gestion des comptes rendus s'accompagne d'une gestion corrélée des images radiologiques, en considérant ainsi l'ensemble des informations multimédia comme parties intégrantes du dossier radiologique.

On trouvera à la fin du chapitre un tableau comparatif qui synthétise les descriptions des systèmes étudiés.

2.4. Notre système.

Notre approche a des aspects communs avec les systèmes présentés précédemment. Notre objectif principal est d'offrir un outil pour gérer d'une façon homogène et intégrée les images radiologiques et les comptes rendus associés. Pour ceci, on se propose d'intégrer les images "logiques", les données administratives et les comptes rendus dans une seule base de données, tandis que les images "physiques" seront stockées dans des fichiers externes à cette base. Dans la mesure du possible, nous essayons de ne pas changer les habitudes de travail et donc de rédaction utilisées actuellement par les médecins.

Nous rappelons que notre travail se situe dans le cadre d'une application réelle menée avec le service de radiologie du C.H.R.U. de Grenoble. Dans ce service, les comptes rendus sont rédigés à la main ou dictés au magnétophone par les médecins radiologues. Ensuite, les secrétaires se chargent de les taper sur un micro-ordinateur à l'aide d'un traitement de texte. Les comptes rendus sont ensuite analysés, pour séparer les données administratives du contenu textuel. Celui-ci contient une brève description du cas, les constatations du radiologue au moment de l'observation des images, ainsi que sa conclusion. L'information textuelle est indexée à l'aide d'une méthode d'indexation (décrite dans le chapitre 4) qui

consiste à faire une indexation des chaînes de caractères composant le texte source, sans faire aucune analyse morphologique, ni syntaxique sur le texte. Une fois l'indexation réalisée, les données administratives et le corps des comptes rendus vont s'intégrer dans la base de données, tandis que les index vont former un fichier externe à la base.

En ce qui concerne les images, nous nous limitons aux images digitalisées générées soit par le scanner, soit par l'imagerie par résonance magnétique. Chaque fichier contenant la séquence des images qui composent un examen, possède une en-tête qui décrit les images contenues dans le fichier. Une partie de cette information est représentée dans la base et constitue l'image "logique".

Pour réaliser l'interrogation, l'utilisateur pourra utiliser les données administratives, ou bien des chaînes de caractères (sans limite de taille) contenant les termes médicaux qui sont utilisés comme conditions de recherche. Il pourra utiliser les opérateurs de l'algèbre relationnelle en combinant ces deux ensembles de critères de recherche, mais, à l'intérieur de la chaîne de termes médicaux, il pourra utiliser seulement les opérateurs logiques AND, OR, et ADJ (pour adjacent). Il faut remarquer que, quand nous parlons de termes médicaux, nous ne faisons pas référence à des mots clés spécifiques, mais à des termes médicaux en général.

L'utilisateur pourra demander de voir les images et/ou les comptes rendus (complets ou partiels) qui répondent aux conditions données dans sa requête. Quand une image est affichée, il s'affiche en même temps un ensemble de données concernant l'image "logique", afin de donner le plus d'information possible à l'utilisateur. L'image est affichée sur un système de traitement d'images, grâce auquel l'utilisateur pourra transformer l'image, s'il le désire. Dans ce cas (mais aussi pour les images que l'utilisateur choisit) l'image est intégrée dans la base comme étant une nouvelle image. L'utilisateur ajoute un commentaire (toujours en texte libre), pour décrire l'intérêt qu'a la nouvelle image, ainsi que les transformations possibles qui ont été appliquées sur l'image source. Par ailleurs, le système récupère les données de l'image "logique" qui est à l'origine de cette nouvelle image et les associe à la description entrée par le médecin. Par la suite, ces images pourront être manipulées directement en utilisant les commentaires, ou bien indirectement à travers l'image d'origine tant qu'elle existera dans la base de données.

Notre approche présente l'avantage de donner une liberté complète aux médecins dans la rédaction de leurs comptes rendus. Ainsi, la récupération des images est possible, entre autres, grâce à l'information contenue dans ces docu-

ments. Par ailleurs, le système offre aussi la possibilité de séparer les images du compte rendu associé, en permettant aux usagers d'isoler et d'identifier les images par elles-mêmes.

Pour réaliser un tel système, nous avons utilisé un SGBD relationnel ainsi que certains des concepts définis dans le cadre du projet TIGRE sur les bases de données généralisées. Ces expériences sont décrites en détail dans les chapitres suivants. Nous nous sommes surtout consacrés à l'aspect base de données multimédia pour l'intégration, le stockage et l'accès aux données à partir des données classiques, des données textuelles et des images (logiques). En ce qui concerne l'interrogation des données textuelles, notre solution n'est que partielle et expérimentale; elle n'a pas pris en compte les aspects syntaxiques et sémantiques du texte des comptes rendus, qui relèvent plutôt du traitement de la langue naturelle. De ce fait, la pertinence de l'information récupérée peut varier beaucoup et n'être pas satisfaisante dans certains cas.

Afin de compléter le système sur cet aspect, on pourrait appliquer aux données textuelles un traitement analogue à ce qui est fait dans REMEDE, où l'on transformerait les phrases originales dans une forme canonique. Ceci est un problème de traduction d'un langage source en un langage cible, dont les phrases seront interprétées en fonction de connaissances préalables. Des travaux sur cet aspect sont actuellement en cours [YAN 86], [CINQ 87]. Les principaux outils utilisés sont un dictionnaire sémantique (classes sémantiques hiérarchisées en niveaux, plus des opérateurs sémantiques exprimant les relations entre les concepts), un analyseur syntaxique et des règles de réécriture.

Tableau comparatif.

NOM DU SYSTEME	DEVELOPPE PAR	AUTEURS ET REFERENCES	ORIENTATION DU SYSTEME	MATERIEL UTILISE
IQS (Image Query Systems)	Université de Kyoto Japon	K. HACHIMURA [HACH 86]	médicale	VAX 11/70 SGBD relationnel MIMER syst. de traitements d'images ILIAD
IDBMS (Image Data-Base Management System)	Université d'Hamburg R.F.A.	A. ASSMANN R. VENEMA [ASSM 83] [ASSM 84] [ASSM 86]	médicale	VAX 11/780 COMTAL (image display) SGBD relationnel ORACLE système connecté au CA-1 (computer angiography 1)
GRAIN (Graphics oriented Relational Algebric INterpreter)	Université d'Illinois Chicago U.S.A.	S.K. CHANG J. REUSS [CHAN 77] [CHAN 79] [CHAN 80]	médicale	PDP 11/40 UNIX, C ISMS (Image Storage Management Systems)
EL	Laboratoire Electrotechnique Japon	N. YOKOYA H. TAMURA [YOKO 82]	médicale	PRIME 450 FORTRAN CUPID (Conversation Utilities for Processing Image Data)
Un système de gestion d'images radiologiques	Université de Nagoya Japon	J. TORIWAKE J. HASEGAWA T. FUKUMURA Y. TAKAGI [TORI 80]	médicale	FORTRAN

Tableau comparatif.

DESCRIPTION	OPERATIONS SUR LES IMAGES
<p>Les images sont gérées à travers les deux concepts d'images physique et image logique, ces dernières sont gérées par un SGBD relationnel. Le type "picture" est introduit et il est utilisé pour représenter virtuellement les images dans la base de données. Des fonctions de traitement d'images peuvent être intégrées dans les requêtes de l'utilisateur, ces fonctions prennent comme argument un constituant de type "picture" pour déterminer le domaine d'action de la fonction.</p>	<p>fonctions de traitement d'images définies par l'utilisateur</p>
<p>Le modèle relationnel est étendu à deux types de données: le type "image" et le type "display". Ils sont utilisés respectivement pour récupérer et afficher les images, qui sont représentées dans la base de données, mais stockées dans des fichiers externes à la base. Les données sont divisées en données du dictionnaire et données de l'utilisateur. Un interpréteur de requêtes analyse et transforme (dans le cas nécessaire), les demandes de l'utilisateur, en utilisant pour ceci les données du dictionnaire.</p>	<p>possibilité d'intégrer les fonctions de traitement d'images, dans le langage d'interrogation.</p>
<p>Les concepts d'image logique et d'image physique sont utilisés pour gérer les images. Les premières sont décrites en trois relations (contours, pages et objets) dans une base de données relationnelle. Des techniques de stockage par page sont utilisées pour archiver les images. Un système de gestion et de stockage d'images sert à la manipulation des fichiers.</p>	<p>dessin des contours sketch dessin de l'image</p>
<p>Prototype utilisé pour gérer une collection de cellules microscopiques ainsi que l'information symbolique associée aux images. La base de données est composée de trois fichiers principaux: le fichier de textes qui contient les diagnostics des médecins, le fichier image et le fichier relationnel; ce dernier est formé par un ensemble de tables qui contiennent les informations structurées concernant les images et les textes. Le système peut récupérer, afficher et analyser les images et faire l'affichage de textes.</p>	<p>affichage et projection, dessin du contours et tracé de l'histogramme</p>
<p>Un "sketch" est utilisé comme moyen d'accès aux images gérées par un système composé de trois types de fichiers: le fichier "master" qui contient les images digitalisées, le fichier "directory" qui contient les "sketchs" et les informations textuelles concernant les images, et le fichier du "travail" qui garde un sous-ensemble du fichier "master" pendant une session de travail. Un "sketch" est un dessin des contours des organes et autres éléments suspects d'être anormaux dans une image radiologique. La communication entre l'utilisateur et le système se fait à travers une série des programmes écrits en FORTRAN.</p>	<p>récupération d'images avec certaines caractéristiques</p>

NOM DU SYSTEME	DEVELOPPE PAR	AUTEURS	ORIENTATION DU SYSTEME	MATERIEL UTILISE
IBIS (Image-Based Information System)	Laboratoire Jet Propulsion. Institut Technologique de Pasadena Claifomie U.S.A.	A. ZOBRIST N. BRAYANT [ZOBR 80]	géographique	FORTRAN VICAR (Video-Image Communication and Retrieval)
EORID (Earth Observation Relational Image Database)	IBM France	L. ASFAR J.P. ROGALA [ROGO 85]	géographique	SGBD relationnel SQL/DS
IMAID (Integrated Image Analysis and Image Database Management System)	Université de Purdue U.S.A.	N.S. CHANG K.S. FU [CHAN 80] [CHAN 81]	géographique	PDP 11/45 UNIX, C SGBD relationnel
EIDES (ETL Image Database for Experimental studies)	Laboratoire Electrotechnique Japon	H. TAMURA [TAMU 80]	géographique	TOS-BAC syst. de traitement d'images APAX et 2DDS
MMS (Message Management System)	Université de Toronto Canada	P. ECONOMOPOULOS F.H. LOCHOVSKY [ECON 83]	général	poste de travail SUN UNIX , C SGBD relationnel MISTRESS
IDMS (Integrated Database Management System)	Université de l'état de New York Buffalo U.S.A.	G.Y. TANG [TANG 81]	général	non-implémenté

DESCRIPTION	OPERATIONS SUR LES IMAGES
<p>Système chargé de faire la gestion de données obtenues par les capteurs, il réalise aussi certaines opérations pour transformer ces données dans d'autres objets tels que les images. Le type de données "raster" est introduit pour pouvoir représenter les images dans une base de données relationnelle. En général, le système est capable de faire la récupération et l'analyse de l'information, ainsi que la génération de rapports.</p>	<p>rotation, projection, changement d'échelle, copy d'une image vers l'autre</p>
<p>Une base de données relationnelle et d'autres utilitaires facilitent à l'utilisateur la gestion des images de satellite. Les utilitaires prennent en charge les requêtes de l'utilisateur afin d'accéder aux images et restituent les informations relatives à la description des images, à leur contenu et à leurs modalités d'obtention.</p>	<p>recherche, lecture et écriture des images, calcul des coordonnées, définition d'image virtuelle à partir d'une table de transcodage, changement de référentiel</p>
<p>Interfaçage entre un SGBD relationnel et un système de traitement d'images. Des caractéristiques des images (centre, coordonnées, etc) sont extraites à l'aide du système de traitement d'images et stockées dans la base de données, tandis que les images numérisées restent en dehors de la base. Les caractéristiques sont utilisées par l'utilisateur pour filtrer l'information, et récupérer les images cherchées. Si l'information contenue dans la base n'est pas suffisante, d'autres caractéristiques seront extraites pour compléter l'information.</p>	<p>distance entre deux points surface d'une région, périmètre d'une région, similarité d'entités, intersection, union, négation et soustraction de points et de segments</p>
<p>La base d'images EIDES contient un nombre considérable d'images avec un format standard. Elles sont utilisées pour tester des algorithmes de reconnaissance des formes et de traitement des images. Le logiciel construit pour manipuler la base EIDES comprend trois aspects : l'entrée et la récupération des images, un système interactif (CREID) pour la manipulation des fichiers images et un système de traitement d'images.</p>	<p>transformation de l'entête reformatage, montrer ou changer les valeurs de pixels dans une zone réduite</p>
<p>Une image est considérée comme un ensemble d'objets, et sa gestion consiste à gérer ces objets ainsi que les relations qui peuvent exister entre eux. Une base de données relationnelle est utilisée pour représenter le schéma de données, qui est formé par trois types d'entités (entité de type image, entité de type relation, et entité de type objet) et des instances de chacune. L'accès au contenu de la base se fait en utilisant trois types de formulaires correspondant à chaque type d'entité.</p>	<p>en utilisant les formulaires, l'utilisateur peut interroger sur les objets ou les relations entre eux</p>
<p>Les données structurées et les données multi-média sont traitées d'une manière uniforme, et stockées dans la même base de données. Le modèle relationnel est étendu avec l'introduction de deux nouveaux types de données: le type "picture" et le type "device". Le type picture est caractérisé par trois nombres: M, N et B, où $M \times N$ définit la taille de la matrice, et B définit le nombre de bits nécessaires pour stocker chaque élément de la matrice. Le type "device" prend ses valeurs parmi les entrées et sorties reconnues par le système.</p>	<p>similarité, contenu de l'image, changement des niveaux de gris</p>

NOM DU SYSTEME	DEVELOPPE PAR	AUTEURS	ORIENTATION DU SYSTEME	MATERIEL UTILISE
ELF (Extended relational model for Large Flexible database)	Université de Tokio Japon	K. YAMAGUCHI [YAMA 8a]	général	non-implémenté
ADM (Aggregate Data Manager)	IBM Japon Centre Scientific de Tokio	Y. TAKAO S. ITOH J. IISAKA [TAKA 80]	général	IBM 370 en PL/1, Assembleur, SGBD relationnel SYSTEM-R
SIRENE Serveur d'Imagerie médicale accessible via un REseau Numérique Experimental	CHR Rennes CCETT ESE Université de Rennes	JM. SCARABIN et al.	médicale	serveur d'archivage COPERNIQUE terminaux divers (haut de gamme, milieu de gamme et bas de gamme)
MINOS	Université de Waterloo Université de Maryland CANADA	S. CHRISTODULAKIS C. FALOUTSOS M. THEODORIDOU F. HO [CHRI 86 a] [CHRI 86 b] [CHRI 86 c]	général bien adapté au médicale	SUN-3

DESCRIPTION	OPERATIONS SUR LES IMAGES
<p>Le modèle relationnel est étendu en permettant que le nom d'une relation soit compris dans le domaine spécifique d'un constituant. Les objets et les relations entre eux sont décrits dans une base de données relationnelle. Les structures sémantiques des objets font partie de la base, et elles sont traitées comme des contraintes d'intégrité.</p>	<p>obtention de caractéristiques, position, couleur, égalité et similarité</p>
<p>Un SGBD relationnel est utilisé pour manipuler d'une manière intégrée les données multi-média qui sont entièrement stockées dans la base de données. Le type de données "aggregate" est ajouté au modèle relationnel afin de manipuler toutes les données non-atomiques. Le prototype existant est capable de gérer des ensembles et des images (images binaires et images avec des niveaux de gris). Pour manipuler et afficher les données multi-média, on utilise un éditeur qui est en accord avec la nature de la donnée</p>	<p>égalité, non-égalité, plus les fonctions de l'éditeur: * extraction, * agrandissement, * rotation, * translation, * recouvrement</p>
<p>Archivage et transmission d'images, des textes associés et de la voix. Recherches multicritères sur les données textuelles. Création de dossiers médicaux électroniques. Puissante manipulation des données multimédia grâce à l'utilisation de terminaux de haut gamme et de stations de travail.</p>	<p>sélection sur la profondeur sélection sur la résolution affichage, visualisation simultanée de plusieurs images, comparaison, évaluations quantitatives (ex : calcul de distances, de surface, etc), convolution, soustraction, analyse d'histogrammes</p>
<p>Système multimédia orienté objet. Fonctions de présentation pour les documents. Extraction de tout type d'information contenue dans un document. Trois éditeurs spécialisés pour créer et manipuler les textes, la voix et les images</p>	<p>générations, affichage, extraction d'une partie de l'image, translation, rotation, transformation de présentation (ex: graphique à tableau)</p>



CHAPITRE III

BASES DE DONNEES GENERALISEES



CHAPITRE 3

3. LES BASES DE DONNEES GENERALISEES.

L'aspect modélisation et manipulation de données généralisées ou multimédia est considéré comme une orientation de recherche qu'il faut privilégier. Les modèles de données résultant de ces dernières recherches ont donné lieu aux bases de données généralisées. Comme nous l'avons déjà mentionné, cette génération de base de données est orientée vers la mise en œuvre des applications où les données sont moins structurées et éventuellement plus complexes que les données gérées par les bases de données classiques. Les objets complexes peuvent être caractérisés par des attributs simples, par des attributs composés ou même par des attributs qui représentent un autre objet.

La principale motivation de ces recherches est d'offrir une bonne performance de l'exécution des opérations qui ont pour but de récupérer et de manipuler les objets complexes. Un modèle de gestion d'objets complexes doit pouvoir supporter efficacement des requêtes qui visent à extraire l'objet complet ou ses composants. Les modèles de données généralisées existants offrent les concepts de base pour manipuler les objets, effectuer des opérations entre eux et exprimer certaines contraintes d'intégrité. Mais ils diffèrent dans le formalisme pour représenter les objets, ainsi que dans des détails spécifiques associés aux concepts de base. Souvent, ces modèles sont orientés pour être utilisés dans des applications spécifiques, telles que la bureautique, la conception de circuits VLSI, etc.

Les différents modèles de cette nouvelle génération peuvent être classés comme suit [LOPE 86], [BROD 81]:

A) les modèles sémantiques : le but principal lors de la conception d'un modèle de données est d'améliorer les possibilités d'accroissement de la sémantique des applications au niveau conceptuel (description de l'organisation) et de la sémantique au niveau logique (organisation des données). Les modèles sémantiques peuvent être classés en cinq groupes :

- * extensions directes des modèles classiques,
- * modèles mathématiques,
- * modèles irréductibles,
- * modèles statiques à hiérarchie sémantique,

* modèles dynamiques à hiérarchie sémantique,

B) les modèles dédiés : ces modèles sont des adaptations ou des spécialisations de modèles sémantiques pour prendre en compte les besoins d'applications particulières. Ces modèles fournissent des concepts de haut niveau propres à un domaine donné, permettant ainsi une modélisation plus aisée et plus précise des applications,

C) les modèles orienté objet : ces modèles consistent en la définition d'objets, où chaque objet possède des informations décrivant son état et des méthodes pour traiter ses informations. Les objets communiquent entre eux par l'intermédiaire de messages.

Plusieurs approches ont été suivies dans ces recherches, parmi elles nous pouvons mentionner les extensions au modèle de données relationnel ou la combinaison d'un modèle existant avec des systèmes indépendants manipulant des types de données généralisées. Dans le premier groupe on peut trouver par exemple les extensions faites dans le système R [HASK 82] et dans POSTGRES [STON 86], et dans le deuxième groupe, on trouve par exemple le modèle TIGRE.

Dans ce qui suit, nous faisons une brève description des extensions faites au modèle relationnel dans le système R et dans POSTGRES. Le modèle TIGRE est décrit plus en détail dans la section suivante.

En ce qui concerne le système R, de nouvelles fonctionnalités ont été ajoutées au modèle relationnel, permettant de :

- * stocker les objets complexes et les données non formatées dans des champs de type *long*. Il est possible de sélectionner, insérer ou modifier ce type de champ "par morceaux",

- * traiter des données structurées mais hétérogènes comme une unité. Un objet est considéré comme un agrégat de n-uplets, pouvant être réparti dans différentes relations,

- * rendre les transactions interactives, c'est-à-dire que l'utilisateur n'est pas forcé d'attendre qu'une transaction soit finie, pour réutiliser l'objet concerné. Le verrouillage se fait donc sur des parties de l'objet et non sur l'objet complet.

Les extensions faites dans POSTGRES au modèle relationnel peuvent être ainsi résumées :

* Trois types de données ont été ajoutés dans le but de gérer les objets complexes : *le type tableaux*, *le type procédure* et *le type POSTQUEL*. Le premier type représente des tableaux de dimension n-aire et de longueur variable. Un cas spécial est le type *texte* qui est un tableau unidimensionnel de caractères. Le type POSTQUEL sert à définir des constituants qui auront comme valeurs une série de commandes de manipulation de la base de données; par exemple, nous pouvons définir une relation dont l'un des constituants est de type POSTQUEL.

FORME (nom = string(8), objet = POSTQUEL);

Un exemple de cette relation est :

nom	obj
carré	retrieve (POLYGONE.all) where POLYGONE.id = 10

POLYGONE est une relation définie préalablement.

Les constituants du type procédure contiennent des instructions écrites dans un langage hôte (ex. C, PASCAL, etc), autorisant des commandes de manipulation de la base de données. Par exemple, une procédure pourrait être définie pour afficher un objet spécifique. Dans ce cas, elle contiendra les commandes qui vont sélectionner l'objet dans la base, ainsi que les instructions nécessaires pour l'afficher.

* Gestion de versions et d'historiques des données. Dans POSTGRES, les relations ne sont jamais détruites ou modifiées (à moins que l'utilisateur ne signale le contraire). Chaque fois qu'une opération d'"insert", de "delete" ou d'"update" intervient, une nouvelle version de la relation est créée. L'utilisateur a un accès par défaut à la dernière version de la relation, mais des commandes de manipulation lui permettent d'accéder aux versions antérieures.

* Implémentation d'itération automatique de commandes de manipulation, d'*alerteurs*, et de *triggers*. Une commande peut être exécutée itérativement, jusqu'à ce qu'elle n'ait plus d'effet. Un *alerteur* est utilisé pour signaler certains états spécifiques de la base de données, par exemple, quand un constituant a une valeur déterminée. Les *triggers* sont des actions (append,

delete, ou replace) qui sont exécutées automatiquement quand une condition est accomplie.

3.1. Le modèle de données TIGRE.

Le modèle de données TIGRE [VELE 84] est conçu dans le but d'offrir un système de gestion de base de données offrant des fonctionnalités nouvelles, pour essayer de satisfaire les besoins en matière de gestion de données généralisées ou multimédia. Dans TIGRE, les **données généralisées** sont gérées d'une manière intégrée par un même langage appelé LAMBDA [FER 84]. Une importance particulière est accordée à la bureautique, où les objets les plus fréquemment manipulés sont les documents. Les documents possèdent une structure interne et une sémantique qui leur sont propres, couramment ils comportent du texte, des photographies, des schémas géométriques et éventuellement ils comportent aussi des commentaires écrits ou oraux.

L'intégration de tous ces éléments au sein d'une seule entité est appelé "**document généralisé**". Plus précisément, un "**document généralisé**" est un objet pouvant contenir des données alphanumériques, des images, des formules mathématiques, des graphiques, etc., au sein d'une seule entité.

On a voulu intégrer le modèle de "**document généralisé**" à un modèle de données alphanumériques, dans le but de pouvoir décrire les documents dans les environnements des applications où ils évoluent. Pour cela, TIGRE utilise le modèle "Entité-Association" afin de réaliser l'interprétation de la réalité perceptible, en lui intégrant le concept de **document généralisé** comme un constructeur de types. Les attributs d'une entité ou d'une association peuvent être définis sur un type *document*. Cette extension permet la prise en compte de données complexes à partir de constructeurs de type adaptés. Le schéma conceptuel est considéré comme une collection de types de données définis au moyen de types de base et de constructeurs particuliers. LAMBDA est le langage de manipulation associé au modèle TIGRE, il permet de retrouver des documents en fonction de leur contenu et de leurs liaisons sémantiques avec le reste d'objets de la base.

Le modèle TIGRE fait une extension du modèle Entité-Association, en ajoutant des abstractions telles que la généralisation et l'agrégation. La généralisation est obtenue par trois opérateurs: *la spécialisation, l'union et l'intersection.*

Pour sa part, deux formes d'agrégation sont offertes: *l'agrégation d'entités et l'agrégation d'associations.*

Le SGBD TIGRE est construit sur un SGBD relationnel et sur des modules indépendants appelés "modules de gros objets complexes", qui sont utilisés pour la gestion des documents généralisés. Dans la réalisation actuelle, le prototype TIGRE s'appuie sur le SGBD ORACLE.

3.2. Les types de données du modèle TIGRE.

Les types de données du modèle TIGRE sont une extension de certains types rencontrés dans le langage PASCAL. Les types de données sont définis en utilisant une collection de structures de données, qui correspondent à des catégories de types de données. Les types de données ainsi obtenus peuvent être combinés à l'aide d'opérateurs de types, pour produire de nouveaux types.

Il existent trois structures de données dans le modèle TIGRE, celle correspondant aux types de base, aux types construits et aux types-classe. Plus récemment, des extensions au modèle TIGRE ont été réalisées pour ajouter les types concernant le temps [BUI 86]. Ainsi, cinq types ont été ajoutés : type de base temps simple, type de base temps restreints, type intervalle de temps, type durée et type période.

3.2.1. Les types de base.

Les types de base sont définis par : les types simples et les types restreints.

3.2.1.1. Les types simples.

Les types simples comprennent les types *entier, réel, booléen et chaîne de caractères*. Ces types sont les composantes à partir desquelles tous les autres types sont construits.

3.2.1.2. Les types restreints.

Les types restreints comprennent les types *scalaires et intervalles*. On les appelle restreints, parce qu'ils spécifient implicitement une contrainte d'intégrité par rapport aux types de base simples sur lesquels ils sont définis. Par exemple, une valeur du type *intervalle* est un entier ou un réel qui doit être compris entre une limite inférieure et une limite supérieure données.

3.2.2. Les types construits

Les types construits sont obtenus par application des constructeurs suivants: "*renommage*", "*enregistrement*", "*tableau*" et "*document*".

3.2.2.1. Le constructeur "*renommage*".

Le constructeur *renommage* permet de renommer un type de base. Ainsi, la distinction entre des types sémantiquement différents définis sur le même type de base, est rendue possible en utilisant ce constructeur.

Exemple :

```
type nom : string (30) ;  
type temperature : real ;
```

Une fois que les types renommés sont définis, ils peuvent être utilisés comme types des constituants. Par exemple le constituant *nom_patient* peut être défini sur le type *nom*:

```
nom_patient : nom;
```

3.2.2.2. Le constructeur "*enregistrement*".

Le constructeur *enregistrement* est similaire au "*record*" du PASCAL, mais il n'est défini que sur des types de *base* ou *renommés*. Ce type est très utile pour représenter des objets définis à partir d'autres objets.

Exemple :

```
type adresse : record
    no          : integer ;
    rue         : integer (25) ;
    code_postal : integer ;
    ville       : nom_de_ville ;
end ;
```

3.2.2.3. Le constructeur "tableau".

Le constructeur *tableau* est similaire au type "array" (à une dimension) de PASCAL. Ce constructeur produit des types dont les réalisations sont des listes d'un même type de base ou renommé. Le type *tableau* est très utile pour modéliser des dépendances multivaluées. Par exemple, si, pour l'entité *ANTIBIOTIQUES*, la connaissance du nom de l'antibiotique détermine l'ensemble de contre-indications et d'effets indésirables, nous pourrions représenter cette dépendance par:

```
Type ANTIBIOTIQUE : entity
    key nom : string (30) end_key ;
    contre-indications : array (10) of string (50) ;
    effets-indésirables : array (10) of string (50) ;
End;
```

3.2.2.4. Le constructeur "document".

Le constructeur *document* est utilisé pour la définition de ce que l'on entend par "document généralisé" dans le modèle TIGRE. Le type *document* spécifie une structure hiérarchique qui doit être satisfaite par toutes les occurrences du type. Cette structure sert à construire les *documents*, à les parcourir, à les présenter, ou encore à faire référence à d'autres parties dans le même *document*.

La structure hiérarchique de types *documents* est construite à l'aide d'éléments composés qui sont les nœuds non-terminaux dans la hiérarchie. Ces éléments composés sont obtenus par des opérateurs de composition à partir des objets déjà définis. Ces derniers peuvent être des composants élémentaires appelés unités, des éléments composés ou des structures hiérarchiques de *documents* déjà définis. Le terme unité recouvre tous les éléments terminaux acceptés par la grammaire de *document*. Une unité possède l'une des natures suivantes: texte, graphique, image,

formule, tableau, programme, vocal, ou référence à un élément associé. Les constructeurs du type *document* sont:

- mise en séquence d'objets (agrégat) : `begin end` ;
- liste d'objets d'une même classe : `liste (min, max) of` *min* et *max* contraignent la cardinalité de la liste,
- choix entre plusieurs objets : `case s of V1:..., Vn:..` ; *s* est appelé le sélecteur et *V1, ..., Vn* les valeurs du sélecteur.
- affectation des valeurs constantes : `":= nom-constante"`,
la constante peut être une unité ou un élément associé déclaré dans une partie réservée aux constantes.

Le constructeur *document* permet l'utilisation de certains éléments, appelés éléments associés. Chaque élément associé appartient à une classe d'éléments associés. On distingue les classes d'éléments associés suivantes : illustration, note, énoncé bibliographique, marque interne, commentaire, paramètre et fonction. Les paramètres et les fonctions ont une importance particulière, parce qu'ils assurent la liaison entre les *documents* et les autres types de données. Dans un *document*, un paramètre désigne une partie variable, accessible directement par son nom, sans passer par la structure hiérarchique du *document*. Une fonction désigne une règle de calcul et un domaine d'évaluation, par exemple des calculs arithmétiques. Les paramètres définis dans un document peuvent être les paramètres formels des fonctions des documents. Dans ce cas, quand la fonction sera évaluée, elle prendra les valeurs courantes des paramètres utilisés.

Un exemple de l'utilisation du constructeur *document* est la description du corps d'un compte rendu médical. Ce document contient deux paramètres accédés indépendamment de la structure du document :

```
Type COMPTE_RENDU : document
  structure
    Begin
      section : list (1, *) of sect_structure : begin
        nom-section : text;
        texte-section : list (1, *) of text;
      end;
      signature : image;
    end;
  constantes
    paramètre No-ex : Integer;
    date-cr : date;
End;
```

3.2.3. Les types-classe.

L'idée de base est de définir des ensembles d'objets (entités) et des associations entre ces ensembles. Ainsi, les types-classe peuvent être des types entités ou des types-associations.

3.2.3.1. Le type-entité.

Un type-entité est une agrégation des types qui représentent les propriétés (attributs) des entités qui appartiennent à une classe d'entité. Un attribut est défini comme une fonction entre un ensemble d'entités (ou un ensemble d'associations) et un ensemble de valeurs.

A chacun de ses éléments, correspond un ensemble de faits. Un fait regroupe une entité ou une association avec ses valeurs et ses attributs. La notion de fait correspond donc à celle de n-uplet dans le modèle relationnel.

Exemple : *Définir une entité IMAGE ayant comme attributs l'identificateur de l'image, l'organe, la source et la date de l'image.*

Type IMAGE : Entity

```
key iid : Integer end_key ;
  organe : string (15) ;
  source : string (15) ;
  date-img : date ;
```

End ;

3.2.3.2. Le type-association.

Les types-associations représentent un lien sémantique entre les entités. Chaque occurrence d'une association lie une ou plusieurs occurrences de chaque classe d'entités participant à l'association. Les propriétés d'une association sont représentées par des attributs, de la même façon que pour les entités. Chaque fait d'un type-association lie un fait de chaque type-entité qui participe à chaque association.

Le schéma de la base de données est composé par une collection de types-classe. Les définitions de types de *base* et *construit* appartiennent au schéma seulement à travers les types-classe.

Exemple : *Définir une association EXAMEN entre l'entité IMAGE et l'entité COMPTE-RENDU, ayant comme attributs propres la date de la demande de l'examen, ainsi que le nom du médecin demandeur.*

```
Type EXAMEN : Relationship between IMAGE : produit (1,1) and
COMPTE-RENDU : élaboré (1,1);
  date-dem : date;
  med-demandeur : nom ;
```

End ;

Les rôles que jouent les entités par rapport à l'association, sont représentés par les mots *produit* et *élaboré* respectivement. Les numéros contenus entre parenthèses représentent des contraintes d'intégrité sémantiques. Ils signifient que, pour chaque image, il doit exister un et un seul compte rendu qui lui est associé. Idem pour l'entité COMPTE-RENDU.

3.2.4. Les types-temps.

Nous faisons une énumération rapide des types-temps, ajoutés au modèle TIGRE pour inclure la notion de temps (logique), venant ainsi enrichir la sémantique du modèle. En effet, le temps est un aspect qui est largement utilisé dans les applications, pour associer aux objets le temps de création, le temps de vie, etc. Le lecteur intéressé pourra se rapporter à [BUI 86, ADBU 86] pour plus de détail.

Le premier type de temps est le type de base "time". Il est défini par les périodes du calendrier grégorien auxquelles on ajoute l'heure, sous la forme d'heures, minutes et secondes. Une valeur de type "time" est:

1987/04/11 12:29:15

Il y a ensuite les types de base restreints; ils permettent de définir des calendriers avec une granularité plus grande que celle définie par le type précédent. Par exemple :

type date : *time* > *hour*

Cette déclaration spécifie un calendrier du type (année, mois, jour), qui peut prendre comme valeur 1987/04/11.

Le troisième type est le type intervalle. Il définit un intervalle de temps en utilisant deux dates de même précision, dont la première est inférieure ou égale à la deuxième. Exemple :

type annee_scolaire : (1986/10/01-1987/09/30) ;

Le type suivant est le type durée, qui est très utilisé pour spécifier la notion de durée dans une application. Il est représenté par une chaîne de caractères où l'on spécifie la quantité et l'unité de la durée. Un exemple est :

temps_exposition : 30second ;

Le dernier type est le type périodique, il peut servir pour décrire des activités qui se répètent périodiquement. A chaque valeur de ce type, il faut associer une valeur de référence où cette période se déroule. Son constructeur est le mot *periodic* et ses variables associées peuvent prendre comme valeurs par exemple :

month := 12, ou bien *day* := 10-*day* := 20. La première valeur définit le mois de

décembre comme période, tandis que la deuxième définit un période comprise entre le 10 et le 20 de chaque mois.

3.2.5. Opérateurs de types.

Les opérateurs de type TIGRE fournissent une capacité d'abstraction. Ces opérateurs s'appliquent sur des types-classe et produisent d'autres types classe dérivés. Les types-classe dérivés héritent des attributs et peuvent avoir des attributs propres.

Deux formes d'abstraction sont traitées: la généralisation et l'agrégation. Les opérateurs de généralisation sont: la spécialisation, l'union et l'intersection, tandis que l'agrégation opère sur des entités ou sur des associations. Les opérateurs de généralisation sont un outil pour modéliser le fait qu'une classe d'objets est plus spécifique ou plus générale qu'une autre. On exprime implicitement une dépendance existentielle et un héritage d'attributs. Un prédicat de restriction peut être donné pour déterminer quels faits de l'entité parente vont appartenir à l'entité dérivée.

Avec les opérateurs de généralisation, il est possible de former ce que nous appelons une hiérarchie de généralisations, où les nœuds sont des dérivations (par généralisation) de types parents.

De son côté, l'agrégation a un objectif double. Le premier est de modéliser les objets qui sont composés par d'autres objets, et le second est de permettre une vision agrégée d'une association. Deux types d'opérateurs d'agrégation sont disponibles, l'opérateur d'agrégation d'entités et l'opérateur d'agrégation associative. Le premier produit un type-entité à partir d'un ensemble de types-entités arguments. Le deuxième opérateur produit un type-entité qui permet de voir un type-association et les types-entités liés comme un tout.

Exemple : Définir l'entité SCANNER comme étant une spécialisation de l'entité IMAGE, avec le numéro de coupes et le temps d'acquisition comme constituants propres.

Type SCANNER : Specialisation of IMAGE

Where source : 'scanner' ;

No-coupes : integer ;

temps-acquisition : string (6) ;

Le prédicat de restriction " *Where source : 'scanner'* " permet de spécifier que seulement les faits de l'entité IMAGE qui ont *scanner* comme valeur du constituant source, seront hérités par l'entité SCANNER.

3.3. Le modèle TIGRE et l'application médicale.

Pour le service de radiologie décrit dans les chapitres précédents, il est extrêmement important d'avoir un système qui offre une bonne gestion de l'information. Les objets fondamentaux intéressants sont donc: les images radiologiques, les comptes rendus, les patients, les lettres de demande d'examen ainsi que les médecins participants. Parmi tous ces objets, il existe une interaction étroite, guidée par la notion d'examen et de dossier.

La notion d'examen commence avec la lettre de demande qui donne les premières informations. Elle contient une description simple de la maladie du patient et signale le type d'examen que celui-ci doit subir, en indiquant la zone de l'organisme à explorer. Cette lettre contient aussi la date de la demande, le nom du médecin demandeur, son adresse et son numéro de téléphone (la demande peut venir de l'extérieur ou de l'intérieur de l'hôpital). Ensuite, le dossier concernant le patient est récupéré (ou créé si c'est la première fois que le patient passe par le service). Chaque dossier contient un identificateur unique affecté par le service. Les données minimales que le dossier contient (c'est-à-dire, avant la prise d'images du patient) sont les données administratives qui identifient le patient, telles que le nom, le sexe, le numéro de sécurité sociale, la date de naissance, l'adresse personnelle, le numéro d'entrée au CHU (donné par le bureau des entrées), etc.. Toutes ces données sont remplis manuellement par le personnel du service.

Avec la lettre de demande et le dossier du patient, on réalise ensuite la prise des images radiologiques. A cette étape, d'autres informations doivent être générées, telles que la technique utilisée (ex. tomодensitométrie, IRMN, etc), l'utilisation de produit de contraste (dans certains cas), le nombre de coupes réalisées, l'heure de début et l'heure de fin de chaque coupe, ainsi que d'autres informations, dont plusieurs sont liées à l'image physique même (ex: coordonnées, fréquence de démodulation, taille du pixel, etc).

Une fois les images générées, elles sont examinées par le médecin radiologue qui réalise un compte rendu de ses observations. Le compte rendu est numérisé grâce à un système de traitement de textes et il reste en mémoire numérique pour

des utilisations ultérieures. Ces comptes rendus sont récupérables à partir des données alphanumériques comme par exemple le nom du patient ou le numéro du dossier. Un exemplaire "papier" du compte rendu est envoyé au médecin demandeur de l'examen et un autre est archivé dans le dossier du patient. On archive en même temps l'impression de certaines coupes (choisies par le radiologue) et la lettre de demande. Actuellement, pour des raisons d'espace mémoire, les images numérisées obtenues du scanner ou de l'IRMN ne sont pas conservées, mais l'idée à future est d'utiliser un disque optique pour conserver numériquement des clichés sélectionnés dans l'examen ainsi que l'entête du fichier qui décrit ces clichés. De cette façon, chaque examen sera représenté par un fichier contenant les clichés censés être représentatifs de l'examen. L'examen se termine à cette étape et le dossier est archivé (l'archivage de ce dossier est physique et les dossiers ne sont récupérables qu'à partir du numéro qui leur est associé).

Cependant, l'exploitation de l'information continue à être utile dans des autres travaux par exemple :

*** dans les recherches scientifiques ou dans l'enseignement.** On a intérêt à récupérer toutes les images d'une telle pathologie, ou bien, à des fins statistiques, prendre des échantillons et compter les images présentant le cas en étude. Dans certaines occasions, le médecin aura besoin de disposer en même temps de l'image et du compte rendu associé. D'autres fois, il pourra avoir besoin de l'un de ces deux objets seulement. Et dans d'autres cas encore, il n'aura besoin que d'une portion de l'objet (ex. conclusion du compte rendu). En résumé, les différents besoins d'information sont nombreux et concernent tous les objets du dossier radiologique.

Les images sont étroitement liées aux autres informations du dossier, principalement aux comptes rendus, mais il existe un intérêt scientifique, pour rendre accessible les images par des voies autres que le dossier radiologique. Par exemple, il peut arriver que l'image observée par le médecin à un moment donné, soit de grand intérêt pour l'étude d'un cas spécifique. Dans une telle situation, le médecin voudrait rendre indépendante cette image du dossier auquel elle appartient, et pouvoir la récupérer par la suite à partir d'autres critères. Un autre exemple se présente quand on applique des fonctions de traitement d'images à une image originale et que pour des raisons diverses, l'image originale ne soit pas conservée. Il faudra donc prévoir que le schéma d'accès à l'image dérivée ne soit pas dépendant de l'existence de l'image source.

* **dans la recherche rétrospective des cas d'un patient.** On peut avoir besoin de récupérer les informations qui satisfont certaines caractéristiques. Par exemple la récupération d'un examen spécifique subi par le patient, ou bien la récupération de tous les examens du patient X réalisés dans une période de temps donnée.

* **dans le contrôle sur les activités du service.** On a besoin, par exemple, de savoir combien d'examens de tomodensitométrie ont été effectués à une date précise, ou bien d'avoir un contrôle périodique des examens réalisés avec chacune des techniques utilisées dans le service, ou encore le nombre de comptes rendus élaborés, etc.

3.4. Modélisation des objets du dossier radiologique.

Dans cette section, nous présentons la modélisation d'un système qui permettrait de répondre aux besoins présentés précédemment. Tout d'abord, nous allons prendre chacun des objets composant le dossier radiologique, pour faire une association avec le modèle TIGRE et déterminer la nature de chaque objet. La définition du schéma conceptuel est donnée dans l'annexe 1.

Description des entités.

Nous présentons d'abord les entités qui ont été formées avec les objets en question.

3.4.1. L'entité PATIENT.

L'objet patient est composé des données administratives qui permettent d'identifier un patient. Ces données ont une nature classique et peuvent être intégrées dans un type-entité que nous avons appelée PATIENT. Elle contient les attributs suivants :

- * pid -> identificateur du malade (par exemple le numéro INSEEE),
- * nom -> nom du malade
- * sexe,
- * date_nai -> date de naissance du malade,
- * adress_p -> adresse personnelle du malade.

3.4.2. L'entité IMAGES.

L'image radiologique est un objet à structure volumineuse et complexe. Dans la modélisation et réalisation de notre application, nous avons suivi le principe utilisé dans la plupart des systèmes présentés dans l'état de l'art donné dans le chapitre deux. C'est-à-dire, nous avons divisé les images en images "logiques" et "physiques". De cette façon, elles seront toujours accessibles au travers des données textuelles ou alphanumériques. Dans les images radiologiques, il existe des éléments qui sont communs à toutes les images et d'autres qui sont spécifiques aux images produites avec une certaine technique. Une image est formée par un ensemble de coupes ou de clichés. Nous avons donc défini une entité que nous avons appelée IMAGES pour représenter les éléments communs aux images. Et nous avons utilisé l'opérateur de spécialisation (cf. Opérateurs de types) pour exprimer les différences liées aux techniques.

Ainsi, l'entité IMAGES comprend les attributs suivants :

- * n_ex -> numéro d'examen,
- * pid -> identificateur du patient,
- * source -> technique utilisée pour produire l'image (ex. scanner, IRMN, etc),
- * organe -> organe à explorer,
- * date_ex -> date de réalisation de l'examen,
- * nombre_coupes -> nombre de coupes réalisées,
- * épaisseur -> épaisseur en millimètres des coupes (est la même pour toutes les coupes).

Les trois constituants suivants concernent l'image "physique" dont :

- * taille -> concerne la taille des coupes (est la même pour toutes les coupes formant l'image),
- * unité -> nombre d'octets sur lesquels est codé chaque pixel,
- * coupes -> ce constituant est défini comme type document (cf. Le document CLICHES). Ce document contient des données concernant chaque coupe, telles que le numéro de coupe, l'heure de début, l'heure de fin, et les pixels.

Si on ne désire pas stocker les images "physiques" dans la base de données, mais dans des fichiers externes, alors, il faudra enlever de la structure liée au constituant *coupes* l'attribut qui contient les pixels. Il faudra aussi ajouter quatre autres constituants: un qui contiendra le nom de fichier où est stockée l'image, un autre qui nous signale le nombre de blocs occupés par chaque coupe, un autre qui

nous dit combien de blocs libres existent entre une coupe et la suivante, et finalement un dernier qui indique la position du bloc où commencent les données de l'image proprement dite.

Pour chaque source différente d'images (scanner, IRMN, etc), une entité de spécialisation est créée. Comme nous l'avions mentionné au début de ce document, notre application se limite pour le moment à la tomodensitométrie et à l'imagerie par résonance magnétique (IRMN), ainsi, nous avons créé deux spécialisations, l'entité SCANNER et l'entité IRMN.

3.4.3. L'entité SCANNER.

Comme son nom l'indique, l'entité SCANNER concerne les images générées en utilisant la technique du scanner. Cette entité hérite donc de tous les attributs de l'entité IMAGES, mais, en plus, elle comprend les attributs propres suivants :

- * manipulateur -> nom de la personne chargée de manipuler le scanner au moment de réaliser la prise des images,
- * nbr_coup_ss_inj -> nombre de coupes sans injection de produit de contraste,
- * nbr_coup_ac_inj -> nombre de coupes avec injection de produit de contraste,

3.4.4. L'entité IRMN.

Cette entité contient les éléments spécifiques aux images produites en utilisant l'imagerie par résonance magnétique. En plus des attributs hérités de l'entité IMAGES, cette entité contient les attributs propres suivants :

Note: nous mettons entre parenthèses les valeurs que chaque constituant peut prendre.

- * type_collection -> (statique, dynamique, NMR 1 ECHO, NMR 2 ECHO),
- * indicateur_sequence -> (saturation, inversion, spin echo, IRSE, flux),
- * type_antenne -> (tête, corps, surface),
- * resolution -> (low, high),
- * mode_acquisition -> (champ large, champ normal, zoom),
- * orientation_plans_acquisition -> (transverse, sagittal, coronal, oblique),
- * mode_positionnement_coupes -> (standard, interleaved, parametric).

3.4.5. L'entité IMAGE_INDEP.

Cette entité sert à gérer les images que l'utilisateur rend indépendantes. Elles sont, soit une copie virtuelle des images contenues dans le dossier radiologique, soit des images dérivées en appliquant des opérations de traitement d'images. Nous n'avons pas fait une spécialisation de l'entité IMAGES pour représenter ces images, parce que l'intérêt est de pouvoir y accéder indépendamment du dossier radiologique. Cependant, nous avons établi une association entre l'entité IMAGE_REF (qui est une union des spécialisations de l'entité IMAGES) et l'entité IMAGE_INDEP, afin d'accéder aux images sources et aux autres données associées, tant qu'elles existent dans la base de données. Cette association est appelée DERIVATION et elle est décrite plus tard. L'entité IMAGE_INDEP contient donc les attributs suivants :

- * iid -> identificateur de l'image,
- * fonction_applique -> signale le type de fonction appliqué dans le cas où l'on a utilisé des fonctions de traitement d'images,
- * source -> indique le type de technique utilisé pour générer l'image source,
- * organe -> organe exploré,
- * intérêt -> ceci est un texte qui décrit l'intérêt de l'image. Le type SECTION qui est un type document et qui, comme nous le verrons plus tard, sert à décrire les corps des comptes rendus. Il est aussi utilisé pour gérer l'information contenue dans ce constituant.

3.4.6. L'entité IMAGE_REF.

Cette entité est le résultat de l'union de l'entité IMAGES et de ses entités spécialisées (SCANNER et IRMN pour le moment). L'entité IMAGE_REF hérite donc de tous les attributs des entités participant dans l'union. C'est au travers cette entité que l'on pourra déterminer l'image et le cliché exact qui ont servi comme source d'une image copiée ou d'une image dérivée (voir entité IMAGE_INDEP).

3.4.7. L'entité COMPTE_RENDU.

Le compte rendu est le rapport que le médecin radiologue fait sur ses observations d'images radiologiques. Le compte rendu contient donc le numéro d'examen qui permet d'établir une liaison unique avec l'image, des données administratives concernant le patient, le nom du médecin demandeur, le nom du radiologue et sa signature, ainsi que la date et le corps du compte rendu (voir figure 3.1).

N_EX : 12000
N_DOSS : 13000

Le 25.04.87

Dr. PARAMELLE

Monsieur DUPONT Pierre
Né le 15/05/43
No. Sec. Soc. 123456

Tomodensitométrie thorax :
(processus expansif médiastino-pulmonaire droit - étiologie non connue au moment où on pratique l'examen).

Les constatations sont les suivantes :

- processus expansif et invasif de grande dimension siégeant au niveau du médiastin,
 - . au niveau de la loge de Baréty, jusqu'au niveau de la loge pré-carénale les structures anatomiques apparaissent refoulées et comprimées, notamment la veine cave supérieure dont la paroi postérieure est envahie.
 - . loge de la bifurcation
 - . médiastin antérieur au devant du tronc veineux brachio-céphalique.
- opacités pulmonaires en continuité.

En conclusion :

Processus expansif et invasif médiastino-pulmonaire avec atteinte pleurale, dans lequel il est difficile de dissocier une tumeur primitive et des adénopathies régionales de grande dimension. L'hypothèse d'un APC à point de départ droit peut être évoqué.

Dr. J. LEGRAND

figure 3.1

Nous pouvons constater tout de suite que la nature des données contenues dans ce document est assez variée. Il représente en même temps un exemple typique d'un document généralisé du modèle TIGRE. C'est ainsi, que nous avons utilisé le constructeur document pour la modélisation de cet objet.

L'entité COMPTE_RENDU est composée des attributs suivants: date_cr, nom du demandeur, nom du radiologue, et texte. Texte est un attribut défini sur le type document SECTION, qui contient dans sa structure un agrégat composé d'un

constituant *signature* défini sur une unité image, et un constituant *corps* défini sur une liste d'un type qui est aussi un agrégat (voir le document SECTION).

3.4.8. L'entité MEDECIN.

Une entité médecin s'avère indispensable, celle-ci est censée regrouper les données administratives concernant les médecins (radiologue ou demandeur). En effet, il peut arriver que l'on ait besoin d'avoir des informations autres que le nom du médecin. Par exemple, sa spécialité, son adresse, ou bien le numéro de téléphone où on pourrait le contacter pour des urgences ou pour des questions diverses. L'entité MEDECIN définit deux types de médecins, le radiologue et le médecin demandeur. Alors, une fois de plus, l'opérateur de spécialisation nous a permis d'établir cette différence. Deux entités de spécialisation ont été définies sur cette entité, l'entité DEMANDEUR et l'entité RADIOLOGUE. Ces deux dernières entités ne comprennent pas d'attributs. Comme nous le verrons plus tard, elles trouvent leur raison d'être dans les associations que chacun des médecins a avec les autres objets. Pour sa part, l'entité MEDECIN a comme attributs le nom du médecin, son identificateur, sa spécialité, son adresse et son numéro de téléphone.

Description des associations.

Jusqu'ici, nous avons fait une description des entités nécessaires pour faire une modélisation de l'application médicale. Maintenant, nous présentons les associations qui existent entre ces entités. Dans le modèle TIGRE, les associations apportent un aspect sémantique à la modélisation, étant donné qu'elles nous permettent de matérialiser les liens sémantiques existant entre les objets. Par exemple, la notion d'examen est un concept abstrait qui concerne une demande, un patient, des images et des comptes rendus. Cette notion est représentée par une demande d'examen au service de radiologie, la présence d'un patient, la prise des images, l'interprétation des signes observés, et la rédaction d'un compte rendu.

3.4.9. L'association EXAMEN.

Nous définirons une première association que nous avons appelée EXAMEN. Elle lie les entités PATIENT, IMAGE et COMPTE_RENDU. C'est une association ternaire qui représente le fait que, pour un patient *examiné*, il y aura des images *produites*, et pour des images produites, il y aura des comptes rendus *élaborés*.

Ainsi, *examiné*, *produites* et *élaborés* sont les rôles des entités vis-à-vis de l'association. De plus, il est possible d'exprimer des contraintes de cardinalité des entités participant à l'association. Ainsi, pour un patient *examiné*, il peut exister une ou plusieurs images et un ou plusieurs comptes rendus, alors la participation de cette entité dans l'association est (1,*) où " * " représente une cardinalité n .

De même, à une image produite, il ne correspond qu'un seul compte rendu, et une image n'appartient qu'à un seul patient, donc la cardinalité de l'image vis-à-vis de l'association est de (1,1). Finalement, pour un compte rendu élaboré il existe une seule image et un seul patient qui lui sont associés, alors la cardinalité de la participation de cette entité dans l'association est (1,1).

Etant donné qu'il y a une lettre de demande qui participe aussi dans la notion d'examen, l'association aura des attributs propres qui concerneront cette demande. C'est-à-dire, la date de la demande, le nom du médecin demandeur, ainsi que l'organe à explorer seront les constituants propres de cette association.

3.4.10. L'association EN_TRAITEMENT.

Maintenant, nous passons aux associations des médecins avec les autres objets. Comme nous l'avons mentionné, les spécialisations de l'entité MEDECIN (DEMANDEUR et RADIOLOGUE) vont nous permettre d'exprimer les associations de chaque médecin par rapport à chacun des objets du dossier radiologique. Ainsi, l'entité DEMANDEUR a une association avec l'entité PATIENT, et nous la représentons par l'association EN_TRAITEMENT. Cette association illustre le fait qu'un patient "*est traité par*" un médecin (1,1), et qu'un médecin "*traite*" un ou plusieurs patients (1,*). Le rôle de chacune des entités vis-à-vis de cette association est mis en évidence par les caractères en *italique*, suivis de leurs cardinalités correspondantes.

3.4.11. L'association OBSERVATION.

L'association OBSERVATION met en relation l'entité RADIOLOGUE avec l'entité IMAGE. Cette association traduit le fait qu'un radiologue "*observe*" une ou plusieurs images (1,*) et, à son tour, une image est "*observée par*" un radiologue (1,1).

3.4.12. L'association DERIVATION.

L'association dérivation est l'association définie entre les entités IMAGE_REF et IMAGE_INDEP. Elle n'a pas d'attributs propres et elle sert à associer les images dérivées avec les images sources et, à travers ces dernières, établir une communication avec les autres données de la base. L'association exprime donc que, pour 0 ou plusieurs (0,*) images *sources*, il existe 0 ou plusieurs (0,*) images qui sont *dérivées*. Cette association sera valide tant que l'image source correspondante existe dans la base: ceci explique les cardinalités des deux entités par rapport à l'association.

3.4.13. L'entité agrégée DOSSIER.

Finalement, la notion de dossier peut être exprimée avec l'opérateur d'agrégation associative (cf. Opérateurs de classes). Rappelons qu'un dossier est composé d'un objet patient, des objets images et des objets comptes rendus et, en plus, ces objets ont des associations entre eux. Ainsi, nous utilisons les trois entités déjà définies PATIENT, IMAGE et COMPTE_RENDU, ainsi que l'association EXAMEN existant entre ces entités, et nous faisons une abstraction de toutes ces classes dans une nouvelle entité que nous appelons DOSSIER. Cette entité construite avec l'opérateur d'agrégation associative contient un attribut propre qui est le numéro de dossier, toutes les autres informations sont héritées des classes participantes.

Description des attributs définis sur un type document.

3.4.14. Le document SECTION.

Dans l'entité COMPTE_RENDU, le constituant texte a été défini sur une liste de type document appelée SECTION. En effet, la partie textuelle d'un compte rendu peut être formée au maximum de trois sections (introduction, constatations et conclusion), chacune composée d'un nombre variable de paragraphes. Pour exprimer cette hiérarchie, nous avons créé le type document SECTION. Sa structure est formée par un attribut appelé *corps*, qui est une liste de 1 à 3 de l'attribut *paragraphe*. Ce dernier attribut décrit exactement la structure d'une section. Il est défini sur un type agrégat qui contient un attribut appelé *titre*, défini sur une unité textuelle, plus un attribut appelé *contenu* qui est une liste de 1 à *n* unités textuelles. Ceci signifie qu'une section a un titre (introduction, constatations ou conclusion), et un contenu textuel.

3.4.15. Le document CLICHES.

Les entités IMAGES et IMAGE_INDEP ont un constituant *coupes* qui est défini sur le type CLICHE. Le type CLICHES est défini avec l'opérateur document comme une structure contenant un agrégat composé d'un constituant appelé *suite_coupes* qui est une liste de 1, * du constituant *cliches_struct*. Ce dernier décrit la structure d'un cliché, il est défini donc comme un agrégat contenant :

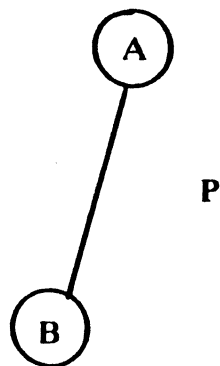
- * *n_cliché* -> un numéro de cliché,
- * *heure_commencement* -> heure à laquelle la coupe a été commencée,
- * *heure_achèvement* -> heure à laquelle la coupe a été finie,
- * *pixels* -> ce constituant est défini sur une unité *image*, et contient le cliché même.

Ce document contient une fonction qui calcule le temps d'acquisition de la coupe. Les paramètres de cette fonction sont en fait les constituants *heure_commencement* et *heure_achèvement*.

3.4.16. Diagramme du schéma conceptuel.

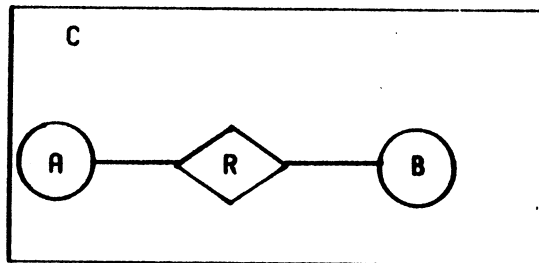
Le schéma conceptuel décrit ci-dessus est maintenant présenté sous forme de diagramme dans la figure 3.2. Pour le représenter, nous utilisons les mêmes conventions graphiques que celles qui sont utilisées dans le modèle TIGRE :

- _ les entités sont représentées par des ronds,
- _ les associations sont représentées par des losanges,
- _ l'opérateur de spécialisation est représenté par deux ronds de la façon suivante :

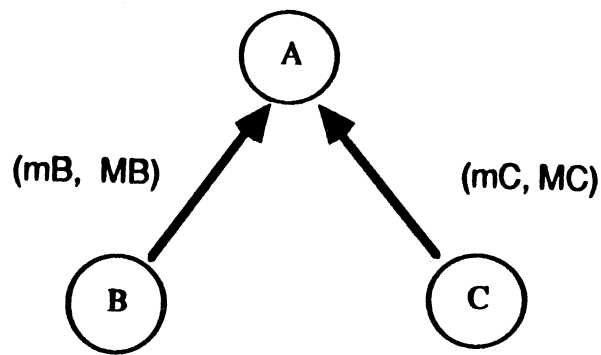


B est une spécialisation de A avec
prédicat de restriction P.

- _ l'opérateur d'agrégation associative est représenté par un rectangle qui enferme les classes participantes :

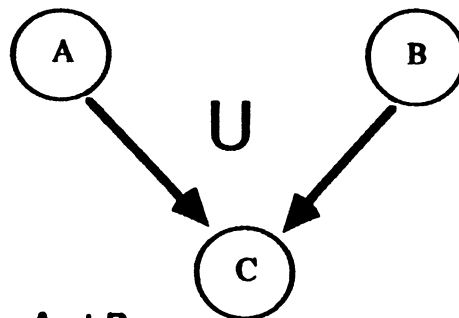


C = agrégation associative de A, B et R.



A : agrégation d'entités B et C.

(mA, MA) : cardinalité minimale et maximale imposées au nombre de faits de B qui peuvent appartenir à un fait de A.



C : union de A et B

L'opérateur d'union produit comme résultat un type-entité lequel est dérivé à partir de deux (ou plus) types-entité arguments ayant un ancêtre commun. Un fait appartenant à un type-entité dérivé par union, doit appartenir à un (tous les) type(s)-entité argument(s).

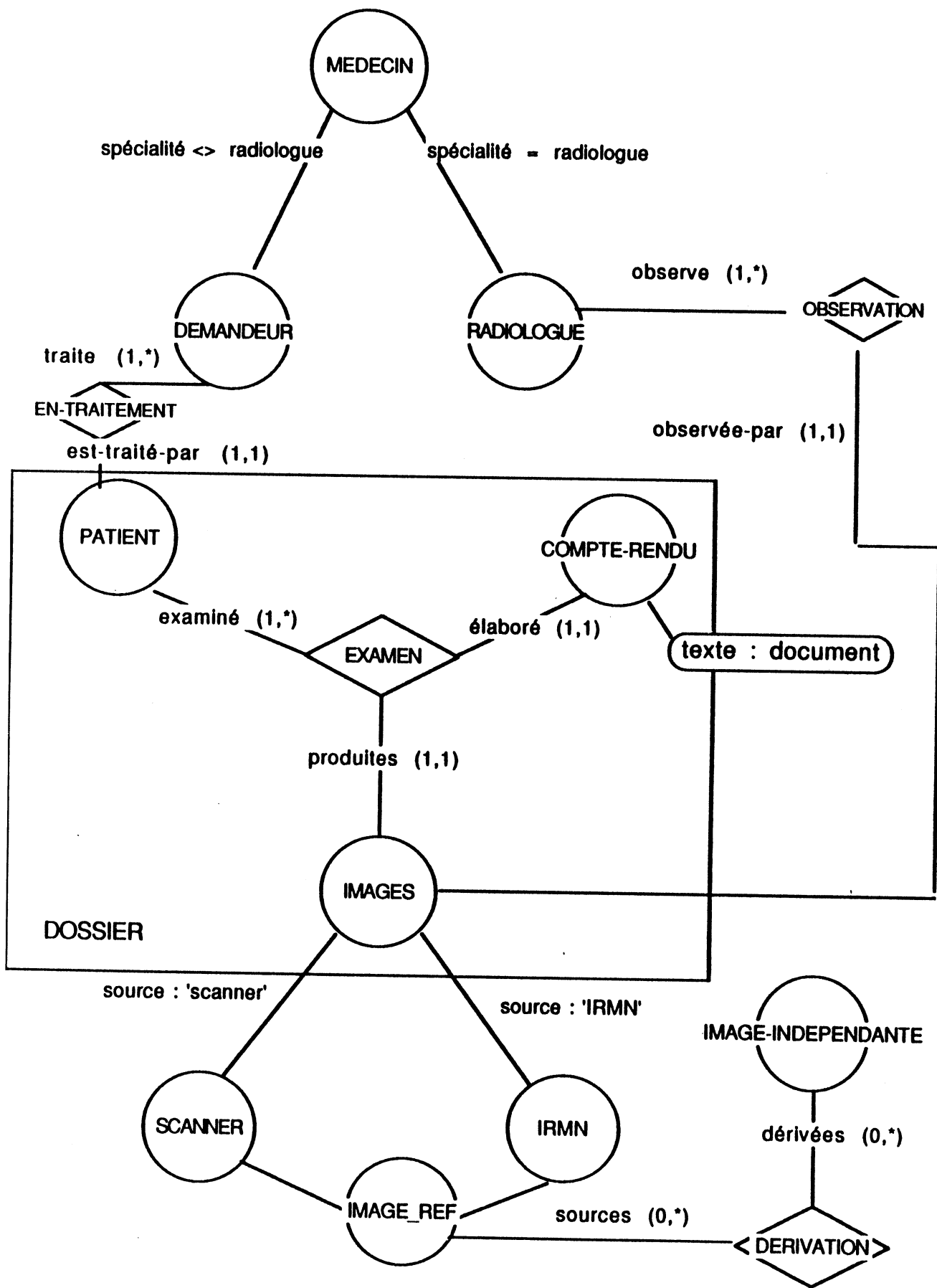


Diagramme du schéma conceptuel
figure 3.2

3.4.17. Autres schémas conceptuels possibles.

Le schéma conceptuel présenté n'est pas la seule solution pour modéliser notre application. D'autres options sont possibles, tout est fonction de la sémantique avec laquelle on considère les objets et les associations entre eux. Le modèle TIGRE nous permet donc d'exprimer ces différences.

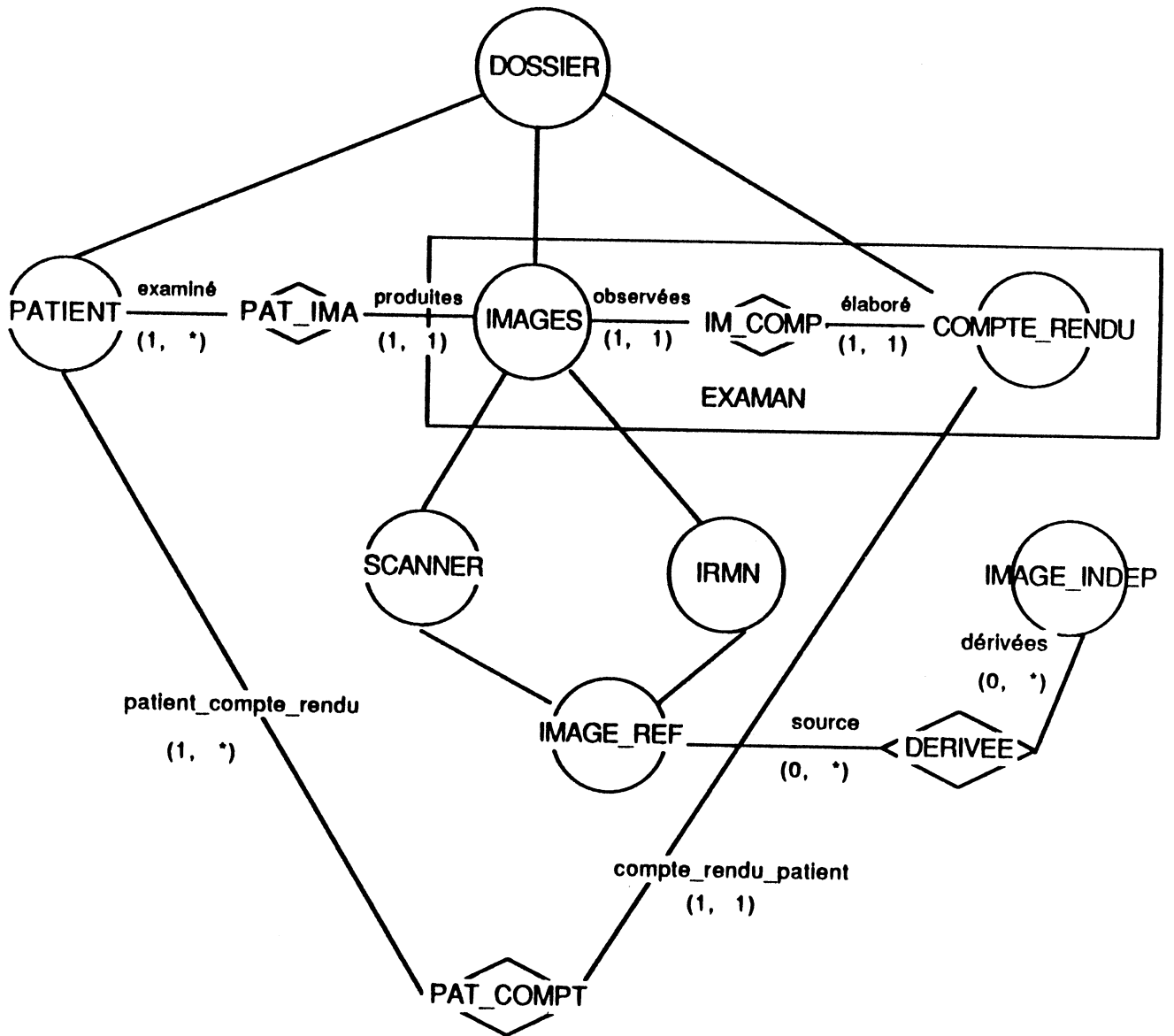
Ainsi, par exemple, en ce qui concerne l'association EXAMEN, au lieu de l'exprimer comme une association ternaire, nous aurions pu établir trois associations différentes. Chacune des associations comporterait un lien sémantique aux entités PATIENT, IMAGE et COMPTE_RENDU. Ces associations pourraient s'appeler PATIENT_IMAGE, IMAGE_COMPTE_RENDU et PATIENT_COMPTE_RENDU. Les rôles et les cardinalités des entités vis-à-vis des associations, sont les mêmes que ceux qui ont été décrits dans l'association EXAMEN. C'est-à-dire, un PATIENT *examiné* peut avoir de 1 à n IMAGES *produites* (1,*). A chaque IMAGE *observée* il ne correspond qu'un seul COMPTE_RENDU (1,1). Et chaque PATIENT peut avoir de 1 à n COMPTE_RENDU (1,*).

Dans ce nouveau schéma, la notion d'examen pourrait être exprimée en faisant l'agrégation associative des entités IMAGES et COMPTE_RENDU avec l'association IMAGE_COMPTE_RENDU (cf. figure 3.3). Cette agrégation est liée à l'entité PATIENT, étant donné que les entités agrégées le sont. Ainsi, pour lier un patient avec un examen, il faudrait faire intervenir l'entité PATIENT et l'entité agrégat EXAMEN.

Pour sa part, l'entité COMPTE_RENDU peut être aussi exprimée de façon différente. C'est-à-dire, au lieu de déclarer le texte du compte rendu comme un document, tout le compte rendu pourrait être déclaré de type document, où la hiérarchie du document serait définie de la façon suivante : le numéro d'examen et l'identificateur du patient sont donnés comme paramètres. De cette façon, ils vont nous permettre d'identifier les documents sans passer par leur structure. Pour sa part, la structure du document comprend la date du compte rendu, le nom du médecin demandeur et le nom du médecin radiologue, la signature de ce dernier (cet attribut étant défini sur une unité de type image), et le texte du compte rendu. Le *texte* du compte rendu est un constituant qui a exactement la même structure que le constituant *corps* défini dans le document SECTION. C'est-à-dire, il est composé d'une liste de 1, * d'un attribut défini sur un agrégat. Cet agrégat comprend le titre de la section et une suite d'unités texte.

La notion de dossier aurait pu être exprimée en faisant une agrégation d'entités à partir des entités PATIENT, IMAGE et COMPTE_RENDU (cf. figure 3.3). Alors l'entité DOSSIER aurait hérité de cette façon de tous les attributs des entités participantes et, comme dans le cas antérieur, un attribut propre à l'association contenant le numéro de dossier serait défini. Les cardinalités de chacune des entités participant à l'agrégation sont : Un DOSSIER ne peut contenir qu'un PATIENT (1, 1), il peut contenir de 1 à n IMAGES (1, *) et de 1 à n COMPTE_RENDU (1, *).

Les deux schémas de données présentés sont assez similaires, la différence se trouve pratiquement dans les associations et les agrégations que nous faisons entre les objets. Pour cette raison, nous n'avons pas fait une description plus détaillée de ce deuxième schéma. Toutefois, la figure 3.3 sert à l'illustrer.



Une autre option du schéma conceptuel

figure 3.3

3.5. Le langage LAMBDA et les exemples de requêtes de l'application médicale.

LAMBDA (LAngage de définition et Manipulation des Bases de Données généralisées) est le langage associé au modèle TIGRE. C'est un langage non procédural de haut niveau basé sur la notion de type et permettant la manipulation de variables. La conception du langage permet son couplage avec les langages de programmation de haut niveau (ex. C, PASCAL). Le principe de couplage est basé sur un pré-compilateur qui traduit les énoncés LAMBDA en des appels au SGBD, avant de compiler le langage de haut niveau.

LAMBDA est composé des énoncés d'interrogation et des énoncés de mise à jour (insertions, suppressions et modifications). Il offre en plus la possibilité de travailler avec des ensembles. Dans les paragraphes suivants, nous faisons une présentation générale du langage LAMBDA. Les lecteurs intéressés pourront se reporter à [VELE 84a, VELE 84b] pour une description complète de ce langage, ainsi que sa complétude et sa comparaison avec d'autres langages.

Nous allons donner une série de requêtes basées sur le diagramme du schéma conceptuel donné dans la figure 3.2. Elles sont dérivées des questions que les médecins souhaiteraient poser au système. Ces exemples ne prétendent pas être exhaustifs, ils ont pour but principal de montrer l'accès aux images et aux comptes rendus, ainsi que l'utilisation des énoncés d'interrogation LAMBDA.

3.5.1. Les énoncés d'interrogation.

Les caractéristiques les plus importantes de LAMBDA du point de vue interrogation sont :

- 1.- La capacité de retrouver des documents ou des sous-ensembles de documents à partir de leur contenu ou à partir de leurs liaisons sémantiques avec les autres informations de la base.
- 2.- L'utilisation de trajets et de variables permettant de supprimer les prédicats de jointure présents dans les langages comme SQL ou QUEL. Les trajets reflètent les liaisons sémantiques entre les données (associations, généralisations et agrégations).

3.- La capacité de construire des ensembles dans les énoncés d'interrogation, ainsi que la possibilité de leur appliquer les fonctions calculs.

La forme générale d'un énoncé d'interrogation, aussi appelée expression de sélection, est :

```
SELECT valeurs  
FROM trajets  
[WHERE condition ];
```

Les *valeurs* précisent l'information que l'on veut obtenir, les *trajets* servent à désigner où se trouve l'information demandée, tandis que les *conditions* spécifient des prédicats que doivent satisfaire les données à retrouver.

Exemple 1 : Pour tous les compte rendus rédigés après le 12/12/86 retrouver le nom du malade.

```
select c.date_cr, p.nom  
from examiné p of compte_rendu c  
where c.date_cr > '12-12-86';
```

Le trajet "examiné p of compte-rendu c" désigne le chemin à parcourir pour obtenir l'information demandée. Le trajet est composé d'un pas par association, qui va de l'entité COMPTE_RENDU (racine du trajet) à l'entité PATIENT (destination du trajet). L'entité PATIENT est mise en évidence à travers le rôle "examiné" qu'elle joue en tant que participant dans l'association EXAMEN. Les variables p et c désignent respectivement l'entité PATIENT et l'entité COMPTE_RENDU. Elles sont appelées **variables de désignation** et le trajet est appelé **trajet de désignation**.

Le pas est appelé **pas par association**, parce que le trajet va d'une entité à une autre, en passant à travers une association et en utilisant le rôle que l'entité destinataire joue par rapport à l'association. Les pas par association sont caractérisés par le mot-clé **of**. Il existe de même les **pas par agrégation de l'agrégat vers l'agrégée**, où le trajet de désignation va d'une entité de type agrégat, vers une entité de type agrégée. Ce type de pas est caractérisé par le mot-clé **part-of**. Il existe enfin les **pas par agrégation de l'agrégée vers l'agrégat**. Celui-ci est un pas équivalent au précédent, mais le trajet de désignation allant d'une entité de type agrégée, vers une entité de type agrégat, il est caractérisé par le mot-clé **containing**.

Les trajets en LAMBDA sont composés de sous-trajets qui peuvent être de deux types, linéaires ou arborescents. A leur tour, les sous-trajets peuvent être composés de plusieurs types de pas. Un trajet linéaire est une suite de pas où la destination d'un pas est le départ du pas suivant. Le trajet de l'exemple 1 est un exemple du trajet linéaire. Un trajet arborescent se divise en plusieurs branches à partir d'une entité ou d'une association. Les trajets arborescents s'utilisent dans les cas où l'on veut désigner des entités qui ne peuvent pas être liées par un trajet linéaire.

Parmi les trajets arborescents, on trouve les trajets arborescents par association et les trajets arborescents par agrégation. Dans les deux cas, le principe consiste à factoriser le mot-clé concerné.

Exemple 2 : sous-trajet arborescent par association.

Retrouver les comptes rendus des patients ayant plus de 50 ans auxquels on a réalisé un scanner du poumon.

```
select c
from (produites i, examiné p) of COMPTE_RENDU c
where p.date-nai < '30-04-37' and i.source = 'scanner' and i.organe =
'poumon' ;
```

Ici les faits de IMAGES *i*, PATIENT *p* et COMPTE_RENDU *c* sont liés par l'association EXAMEN laquelle est reconnue à travers les rôles des entités.

Note : la variable *c* suivant la clause *select* revient à sélectionner tous les faits de l'entité associée (cf. exemple 4).

Pour obtenir les informations propres aux associations, il existe les pas vers les associations. Dans ce type de trajets, la destination est donc une association, et l'on doit passer par au moins deux entités liées par cette association.

Exemple 3 : Donner le nom du médecin demandeur ainsi que la date de la demande de l'examen suivi par Mr. DUPONT le 12/12/86.

```
select e.nom_demandeur, e.date_dem
from examen e : (p, c) of examiné of IMAGES i
where 'DUPONT' = p.nom and i.date_ex = '12/12/86' ;
```

Chaque type de pas dans un trajet impose des contraintes implicites sur les variables de désignation et donc sur les faits à retrouver. Par exemple, dans l'exemple précédent la variable *e* désigne un fait d'examen, qui lie les faits

PATIENT (représenté par la variable *p*) et IMAGES (représenté par la variable *i*). Les conditions figurant dans la clause **where** sont considérées comme des contraintes explicites.

3.5.1.1. Expression de fonctions.

Les fonctions qui apparaissent après le mot-clé **SELECT** d'un énoncé **LAMBDA** doivent appartenir à quatre familles (f1..f4).

Famille f1 : choix de faits d'une classe.

Les fonctions de cette famille permettent de retrouver des faits des classes (type-entité ou type-association) désignées dans les trajets d'une expression de sélection. Les faits doivent satisfaire les contraintes implicites et les contraintes explicites notées dans les trajets et dans la clause *where* respectivement.

Exemple 4 : Retrouver les images du scanner de poumon après injection intraveineuse de produit de contraste.

```
select i  
from scanner i  
where i.organe = 'poumon' and i.nbre_coupes_ac_inj > 0 ;
```

Dans cet exemple, la fonction suivant la clause *select* est représentée par la variable *i*. Elle indique que l'on veut obtenir tous les faits de l'entité **scanner** qui satisfont les contraintes.

Exemple 5 : Donner les comptes rendus concernant les processus expansifs découverts dans la loge de Baréty.

```
select c  
from COMPTE_RENDU c  
where 'processus expansif loge Baréty' in c.texte ;
```

Il s'agit ici d'une recherche par contenu dans une unité textuelle d'un document (*c.texte*).

Famille f2 : choix de valeurs des attributs d'une classe.

Les fonctions de cette famille représentent la projection des faits d'une classe sur un attribut spécifique.

Exemple 6 : Retrouver les noms et la date de naissance des PATIENTs qui ont suivi un examen scanner le 30/03/87.

```
select p.nom, p.date_nai
from produites i of PATIENT p
where i.date_ex = '30-03-87' and i.source = 'scanner';
```

Famille f3 : choix d'un sous-ensemble des valeurs d'un attribut.

Les fonctions de cette famille s'appliquent aux attributs définis sur un type enregistrement, tableau ou document. Elles servent à choisir un élément d'un enregistrement, ou d'un tableau, ou un sous-ensemble d'un document, ou encore la valeur d'un paramètre ou d'une fonction d'un document. En ce qui concerne les fonctions retrouvant un sous-ensemble de documents, une clause **where** peut être contenue dans la fonction. Un exemple illustrant cette possibilité est donné par la suite.

Exemple 7 : Retrouver les conclusions des comptes rendus correspondant aux examens de monsieur DUPONT, et qui ont été signés par le docteur PIERROT.

```
select titre of paragraphe of corps of c.texte where 'conclusion' in titre
from élaboré c of PATIENT p
where p.nom = 'DUPONT' and c.radiologue = 'PIERROT' ;
```

La fonction **titre of paragraphe of corps of c.texte** représente le chemin dans la structure du document SECTION sur lequel est défini le constituant **texte**. Ce chemin est conditionné par la clause **where 'conclusion' in titre**.

Trois autres fonctions applicables aux documents sont contenues dans cette même famille. La fonction **COUNT** qui compte le nombre d'occurrences d'un nœud de la structure d'un document. Les deux autres fonctions portent sur les éléments associés "paramètre" et "fonction" (cf. 2.2.2 Le constructeur document). Une de ces deux fonctions permet de retrouver la valeur d'un ou plusieurs paramètres d'un document, tandis que l'autre fonction permet d'évaluer une ou plusieurs fonctions d'un document. Ces trois fonctions sont considérées comme une sous-classe de la classe de fonctions qui retrouvent un sous-ensemble du document.

Exemple 8 : Combien de thymomes a-t-on observé parmi les processus expansifs de densité tissulaire dans la loge de Baréty?

```
select count (c)  
from COMPTE_RENDU c  
where 'thymomes processus expansifs densité tissulaire loge Baréty' in  
c.texte ;
```

Cette requête compte les occurrences des comptes rendus contenant le texte cherché.

Famille f4 : fonctions d'agrégation.

Les fonctions d'agrégation s'appliquent aux ensembles que l'on peut construire en LAMBDA. On distingue les fonctions suivantes:

- COUNT compte les éléments d'un ensemble,
- SUM calcule la somme d'un ensemble,
- AVG calcule la moyenne d'un ensemble,
- MAX obtient le maximum d'un ensemble de valeurs numériques,
- MIN obtient le minimum d'un ensemble de valeurs numériques.

Les fonctions présentées dans les familles précédentes (f1..f4) s'appliquent aussi à la clause where du langage LAMBDA, avec en plus l'expression de constantes appartenant à la famille f5.

Exemple 9 : Combien d'examens de scanner a fait le docteur LEGRAND entre le 1er. Janvier 1986 et le 15 Janvier 1986.

```
select count (i)  
from observe r of IMAGES i  
where i.source = 'scanner' and r.nom like 'LEGRAND' and  
i.date_ex >= '01.01.86' and i.date_ex <= '15.01.86' ;
```

La requête compte le nombre d'examens qui satisfont les contraintes données.

Famille 5 : expression de constantes.

Cette famille regroupe les expressions constantes. Une expression constante peut être une variable temporaire ou une variable déclarée dans un programme de haut niveau (ex. C, PASCAL). Les constantes admises sont :

- * des constantes définies sur les types de base (entier, booléen, intervalle, etc.),
- * des constantes enregistrement (n-uplets des constantes définies sur un type de base simple),

- * la constante NULL,
- * des expressions régulières de chaînes de caractères de la forme :

[*] chaîne1 (* chaîne2)ⁿ [*]

où $n \geq 0$. Ces expressions sont utilisées pour effectuer des recherches par le contenu sur les unités textuelles d'un document. Le joker '*' correspond à n'importe quelle chaîne de caractères.

Exemple 10 : Donner les comptes rendus et les images où l'on a observé des anévrysmes dans le médiastin antérieur.

```
select c, i
from produites i of COMPTE_RENDU c
where ' anévrysme* médiastin antérieur ' in c.texte ;
```

La requête récupère les comptes rendus qui satisfont la condition textuelle donnée, ainsi que les images que lui sont associées.

3.5.1.2. Expression des ensembles.

LAMBDA permet de construire des ensembles dans une requête (l'équivalent d'une requête imbriquée en SQL). Un ensemble peut apparaître comme argument d'une fonction d'agrégation (c.f. Famille 4), ou dans des prédicats de la clause where qui comparent les éléments d'un ensemble à une valeur ou à un autre ensemble. Ces prédicats peuvent avoir la forme :

* <valeur> in <ensemble>.

* <ensemble> <op-ensembliste> <ensemble>.

où <op-ensembliste> := union(+), intersection (*) et différence (-). Les types des valeurs des ensembles doivent être compatibles entre eux.

3.5.2. Stockage et recherche des documents.

Le stockage et la recherche des documents sont assurés par le gestionnaire de documents, qui offre un ensemble d'opérateurs de manipulation, considérés comme une extension des opérateurs algébriques du modèle relationnel. Les opérateurs concernant la recherche de documents ou de parties de documents sont :

* projection à l'intérieur d'un attribut document :

$\text{project-doc}(R, a, \langle \text{trajetDoc} \rangle [, n])$

Cet opérateur permet de retrouver un sous-ensemble de document. Il a trois arguments obligatoires et un argument facultatif (n , qui est un nom de nœud). R est une relation, a est un attribut défini sur un type document et $\langle \text{trajetDoc} \rangle$ est la représentation d'un trajet dans la structure du document. Si l'argument facultatif est spécifié, le trajet s'applique à partir du nom de nœud donné. Dans le cas contraire, le trajet s'applique depuis la racine du document. L'opérateur produit une relation où les valeurs de a sont des listes ordonnées de couples (nom nœud, identificateur) correspondant au nœud retrouvés par $\langle \text{trajetDoc} \rangle$.

* sélection à l'intérieur d'un attribut document.

$\text{select-doc}(R, a, \langle \text{cond} \rangle [, n])$

Cet opérateur est utilisé pour effectuer de la recherche par contenu dans les unités textuelles d'un document. R est une relation, a est un attribut identifiant le document ou les nœuds du document, et cond est une condition de la forme " $K \langle \text{op} \rangle \text{trajetDoc}$ ", où K est une chaîne de caractères éventuellement partiellement spécifiée (utilisation de jocker *), $\langle \text{op} \rangle$ est un des opérateurs =, <>, in et trajet doc sont des trajets document. Le trajet doit arriver à une unité textuelle ou à un ensemble d'unités textuelles (auquel cas l'opérateur est in). Les requêtes données dans les exemples 2 et 3 contiennent un prédicat de sélection sur un document.

* Extraction des éléments d'un document.

$\text{extract}(R, a, \langle \text{trajetDoc} \rangle [, n])$

On utilise cet opérateur pour extraire des unités textuelles d'un document, dans le but d'effectuer des comparaisons ensemblistes avec d'autres ensembles. Il peut être utilisé aussi pour compter des occurrences des éléments spécifiques dans le document.

Exemple 11 : Récupérer les conclusions des comptes rendus radiologiques de monsieur LEPETIT où l'on a observé des anévrismes de la veine azygos.

```
select paragraphe r of corps of c.texte
from examiné p of COMPTE_RENDU c
where p.nom = 'LEPETIT' and
      'conclusion' in titre of r and
      'anévrisme* veine azygos' in contenu of r ;
```

Cette requête récupère une partie d'un document, celle où le texte satisfait la condition donnée et où le nom de la section est "conclusion". Nous pouvons remarquer que le trajet d'accès à la structure du document est différent de celui donné dans l'exemple 7. Ceci montre qu'il existent plusieurs options pour accéder aux données.

Exemple 12 : Donner les conclusions des comptes rendus où sont observés des kystes broncho-géniques du médiastin postérieur qui sont calcifiés.

```
select paragraphe p of corps of c.texte
from COMPTE_RENDU c
where 'conclusion' = titre of p
      and 'kyste* broncho-génique* médiastin postérieur calcifié*' in
      contenu of p ;
```

Exemple 13 : Donner toutes les images où l'on peut observer des processus expansifs de densité tissulaire dans la loge de Baréty.

Les images qui peuvent répondre à cette requête sont soit des images indépendantes ou des images associées aux comptes rendus. Etant donné que les entités concernées sont indépendantes les unes des autres, alors il nous faut deux requêtes pour répondre à cette question.

```
1. select i
   from élaboré c of IMAGES i
   where 'processus expansifs densité tissulaire loge Baréty' in c.texte ;
```

```
2. select m
   from IMAGE-INDEPENDANT m
   where 'processus expansifs densité tissulaire loge Baréty' in m.texte ;
```

Si nous n'avions pas séparé la requête en deux énoncés d'interrogation, alors un produit aurait été effectué entre les deux entités demandées (*select i, m*), ce qui n'aurait pas fourni le résultat attendu.

3.5.3. Énoncés de mise à jour.

Les énoncés de mise à jour de LAMBDA peuvent être de trois types :

- * l'insertion de faits dans les classes,
- * la suppression de faits d'une classe,
- * la modification de valeurs d'attributs d'une classe.

La commande INSERT est une commande interactive utilisée dans LAMBDA pour faire des insertions dans la base de données. Elle permet d'insérer, dans une classe, des faits provenant d'autres classes de la base.

Exemple 14 : Insertion d'un nouveau PATIENT.

```
Insert (pid = 12345, nom = 'DARTY', sexe = 'M', date_nai = '15.05.48',  
adresse = '1, Bd. Haussman, Paris') to PATIENT;
```

Les suppressions en LAMBDA s'expriment à l'aide de la commande DELETE. Pour supprimer un ensemble de faits, on le qualifie à l'aide d'une clause *where*. S'il n'y a pas de clause *where*, la suppression portera sur tous les faits de la classe.

Exemple 15 : Supprimer le patient inséré dans l'exemple antérieur.

```
Delete i  
from PATIENT p  
where p.pid = 12345;
```

Les modifications en LAMBDA sont exprimées par la commande REPLACE, qui permet de changer la valeur d'un ou plusieurs attributs d'un ensemble de faits appartenant à une seule entité ou à une seule association.

Exemple 16 : Changer la date de naissance de monsieur DARTY pour '15.05.49'.

```
Replace p.date-nai := '12.05.49'  
from PATIENT p  
where 'DARTY' = p.nom ;
```

3.6. Problèmes de mise en œuvre.

Comme nous l'avons mentionné dans l'introduction de ce travail, l'idée de départ était de modéliser et de réaliser l'application médicale en question, en utilisant le modèle et le SGBD TIGRE. Le but de cette modélisation était de réaliser une application pilote qui nous permette de juger les concepts du modèle sur des bases réelles et offrir en même temps des solutions aux besoins informatiques de l'application. Cependant, l'état de développement du prototype TIGRE n'était pas assez avancé pour réaliser directement notre application médicale avec ce SGBD. De plus, les aspects textuels, et en particulier la recherche par contenu dans un document structuré ne sont traités que superficiellement (recherche séquentielle d'une chaîne de caractères). De la même manière, la notion d'image, même si elle apparaît dans le modèle, n'est pas réellement prise en compte dans le prototype TIGRE. Ce prototype n'a été (partiellement) opérationnel que fin 86 alors que notre application avait démarrée en 84.

Par souci d'offrir un environnement à l'application médicale, et en nous inspirant toujours de la philosophie de TIGRE, nous avons voulu expérimenter l'implantation de notre base de données de dossiers médicaux multimédia en utilisant un SGBD relationnel et en développant du logiciel spécifique pour gérer les données textuelles des comptes rendus et les images, c'est-à-dire, les données multimédia. Nous avons réalisé une implémentation du schéma conceptuel à l'aide d'un modèle purement relationnel. Néanmoins, dans cette option, il nous restait toujours le problème de la manipulation des données textuelles et des images; c'est pour cela que nous avons décidé d'utiliser la technique de *signature* pour réaliser les recherches par contenu sur les unités textuelles. La technique de *signature* est une méthode d'indexation textuelle, elle utilise une chaîne de bits contenant une représentation approximative du texte associé. La méthode permet de répondre à des questions dans lesquelles on utilise des suffixes et des préfixes. Une description plus détaillée de la méthode sera donnée dans le chapitre suivant.

En ce qui concerne les images, nous avons décidé de les gérer en utilisant les concepts d'image **physique** et d'image **logique**. Les images physiques ne seront pas stockées dans la base de données, c'est seulement l'image logique qui fait partie de la base. Rappelons que l'image logique contient tous les descripteurs alphanumériques et textuels qui nous permettent d'identifier une image parmi un ensemble. Le chapitre suivant contient donc une description de la modélisation et de l'implémentation de ce schéma relationnel.



CHAPITRE IV

MODELISATION DE L'APPLICATION

MEDICALE AVEC UN SGBD RELATIONNEL



CHAPITRE 4

4. MODELISATION DE L'APPLICATION MEDICALE AVEC UN SGBD RELATIONNEL.

Dans ce chapitre, nous présentons d'abord présenter le schéma relationnel que nous avons conçu à partir du schéma TIGRE décrit précédemment. Ensuite nous faisons une description de l'implémentation de ce schéma et des possibilités de recherche par contenu à partir des données textuelles. Des requêtes similaires aux requêtes données dans le chapitre précédent sont utilisées pour montrer la procédure d'interrogation.

4.1. Schéma relationnel de l'application.

Les spécifications du schéma conceptuel réalisé dans TIGRE ont servi de base pour la réalisation du schéma qui a été implémenté à l'aide d'un SGBD relationnel. Nous avons représenté les entités par des relations et les associations par des attributs communs entre les relations. C'est ainsi que nous avons abouti au schéma suivant (les constituants écrits en **gras** sont les clés de la relation):

PATIENT (**pid**, *nom*, *date_nai*, *sexe*, *adresse_p*)

COMPTE_RENDU (**n_ex**, *pid*, *n_doss*, *date_cr*, *demandeur*, *radiologue*, *date_dem*, *organe*, *origine_dem*)

TEXTE_CR (**id**, *n_ex*, *paragraphe*, *texte*)

IMAGE_INF (**n_ex**, *iid*, *pid*, *n_doss*, *date_ex*, *source*, *organe*, *nbre_coupes*, *épaisseur*, *image_mère*, *taille*, *unité*, *nom_fichier*)

IMAGE_INDEP (**id**, *iid*, *fonction_appliqué*, *intérêt*)

SCANNER (**n_ex**, *manipulateur*, *cap*, *csp*)

IRMN (**n_ex**, *type_collection*, *indic_seq*, *type_ant*, *resolution*, *mode_acq*, *orientation*, *position*)

MEDECIN (*demandeur*, *sexe*, *spécialité*, *adresse_b*, *téléphone*)

4.1.1. Description de chacune des relations.

PATIENT (*pid, nom, date_nai, sexe, adresse_p*)

Cette relation contient les données administratives concernant le patient.

pid.- constituant de type entier, et identificateur du patient,

nom .- chaîne de caractères qui contient le nom du patient.

date_nais .- constituant de type date qui contient la date de naissance du patient.

sexe.- caractère qui peut prendre la valeur *f* (féminin) ou la valeur *m* (masculin).

adresse_p .- chaîne de caractères qui contient l'adresse personnelle du patient.

COMPTE_RENDU (*n_ex, pid, n_doss, date_cr, demandeur, radiologue, date_dem, organe, origine_dem*)

Cette relation contient les données structurées contenues dans un compte rendu.

n_ex.- entier qui contient le numéro d'examen, il identifie d'une manière unique chaque n-uplet de la relation. N_EX est un constituant commun avec la relation IMAGE_INF. Il établit une liaison unique avec l'image qui est associée au compte rendu.

pid.- ce constituant est de même type et de même nature de donnée que celle de la relation PATIENT. Il établit un lien unique entre le patient et le compte rendu.

n_doss.- entier qui sert à identifier le dossier auquel appartient le compte rendu.

date_cr.- donnée de type *date* qui exprime la date à laquelle le compte rendu a été rédigé.

demandeur.- chaîne de caractères qui contient le nom du médecin demandeur de l'examen.

radiologue.- chaîne de caractères qui contient le nom du radiologue qui a rédigé le compte rendu.

date_dem.- type *date* qui contient la date à laquelle l'examen a été demandé.

organe.- chaîne de caractères qui contient le nom de l'organe à explorer.

origine_dem.- chaîne de caractères qui nous indique si la demande vient d'un service de l'hôpital, ou si elle vient de l'extérieur de l'hôpital.

TEXTE_CR (id, n_ex, paragraphe, texte)

Cette relation contient le corps du compte rendu, c'est-à-dire les données textuelles.

id .- est l'identificateur du bloc logique assigné par le système (cf. section 4.2.2.2. Méthode de signature par codage surimposé).

n_ex.- entier qui sert à établir un lien unique avec la relation compte rendu, il contient le numéro de l'examen.

paragraphe.- chaîne de caractères (15) qui contient le nom de la section (introduction, constatations ou conclusion) à laquelle appartient le texte contenu dans le constituant *texte*.

texte.- champ de type *long* qui mémorise un bloc logique du texte de compte rendu préalablement signé (un champ de type *long* peut recevoir une chaîne de caractères d'une longueur maximale de 64 octets).

MEDECIN (demandeur, sexe, spécialité, adresse_b, téléphone)

Cette relation concerne seulement les médecins demandeurs d'examens. Elle sert à rassembler des données administratives qui intéressent le service de radiologie.

demandeur.- chaîne de caractères qui contient le nom du médecin demandeur.

sexe .- sexe du médecin demandeur.

spécialité.- chaîne de caractères qui contient la spécialité du médecin demandeur.

adresse_b et **téléphone** sont deux constituants de type caractère, le premier contient l'adresse et le deuxième contient le numéro de téléphone du médecin demandeur.

IMAGE_INF (n_ex, iid, pid, n_doss, date_ex, source, organe, nbre_coupes, épaisseur, image_mère, taille, unité, nom_fichier)

Cette relation contient des informations concernant une image logique, ainsi que l'adresse de l'image physique. Dans cette relation, sont décrites les images sources ou originales et les images que les médecins ont décidé de rendre indépendantes. Dans ce qui suit nous appelons ces dernières images "sous_image".

n_ex.- entier qui sert à rattacher les images au compte rendu respectif, et à identifier chaque n-uplet de la relation.

iid.- est un identificateur de type entier généré par le système pour identifier une sous_image.

pid.- identificateur du patient auquel appartient l'image. Il est de type entier comme les *pids* déclarés précédemment.

n_doss.- constituant de type entier qui sert à identifier le dossier contenant l'image.

Note : ces trois derniers constituants auront des valeurs NULLS quand le n-uplet concerne une sous_image.

date_ex.- constituant de type *date* qui contient la date de prise des images ou la date de création des sous_images.

source.- chaîne de caractères qui contient le nom de la technique utilisée pour générer l'image (ex. scanner, IRMN, etc).

organe.- chaîne de caractères qui contient le nom de l'organe examiné

nbre_coupes.- entier qui signale le nombre de coupes (appelées aussi clichés) qui composent l'image.

épaisseur.- chaîne de caractères qui contient l'épaisseur des coupes. Cette valeur est constante pour toutes les coupes qui forment une image.

image_mère.- constituant de type entier qui garde l'identificateur de l'image qui a servi comme base pour créer la sous_image. Ce constituant aura la valeur 0 pour les images sources et une valeur > 0 pour les sous_images.

Les trois constituants suivants concernent l'information relative au stockage des images physiques.

taille.- chaîne de caractères qui donne la taille de l'image en pixels (ex. 512 x 512, etc).

unité.- chaîne de caractères qui contient l'unité de codage de chaque pixel, c'est-à-dire, l'espace mémoire occupé par un pixel (octet ou mot).

nom_fichier.- chaîne de caractères qui contient le "path-name" du fichier qui stocke l'image physique.

IMAGE_INDEP (*id*, *iid*, *fonction_appliqué*, *intérêt*)

Cette relation est équivalente à la relation TEXTE_CR. Elle contient les commentaires qui expriment l'intérêt de rendre l'image indépendante. Malgré la similitude de cette relation avec la relation TEXTE_CR, ces deux relations ne peuvent pas être fusionnées. Dans la relation TEXTE_CR le numéro d'examen (*n_ex*) sert à établir le lien entre les textes et les images. Quand l'image est

indépendante, elle n'a plus aucune relation avec le numéro d'examen. Pour cette raison nous avons créé la relation IMAGE_INDEP qui compte l'identificateur de la sous_image pour réaliser la liaison avec la relation IMAGE_INF.

id.- identificateur généré par le système pour identifier le bloc textuel.

iid.- identificateur de sous_image et lien avec la relation IMAGE_INF.

fonction_appliquée.- chaîne de caractères qui contient le nom ou la fonction qui éventuellement a été utilisée pour générer l'image en question.

intérêt.- champ de type *long* qui contient les commentaires des médecins, où ils expriment l'intérêt spécial de cette image.

SCANNER (*n_ex, manipulateur, cap, csp*)

Cette relation est équivalente à une des spécialisations définies avec le modèle TIGRE. Elle a pour fonction le stockage de l'information des images réalisées avec le scanner.

n_ex.- entier qui sert à établir la liaison avec les autres relations (IMAGE_INF, COMPTE_RENDU, EXAMEN) qui contiennent ce même constituant. En même temps, il sert à identifier le n-uplet concernant l'image, dans la relation IMAGE_INF.

manipulateur.- chaîne de caractères qui contient le nom du manipulateur du scanner.

cap.- champ de type entier qui contient le nombre de coupes qui ont été réalisées avec l'injection de produit de contraste.

csp.- champ de type entier qui contient le nombre de coupes qui ont été réalisées sans l'injection de produit de contraste.

IRMN (*n_ex, type_collection, indic_seq, type_ant, resolution, mode_acq, orientation, position*)

Celle-ci serait la deuxième spécialisation définie dans le modèle TIGRE. Comme dans le cas précédent, elle concerne les informations qui sont spécifiques à la technique IRMN.

n_ex.- sert à établir une liaison avec les relations IMAGE_INF et COMPTE_RENDU.

Tous les autres constituants sont définis comme des chaînes de caractères. Ils contiennent des informations techniques propres à l'imagerie par résonance magnétique (IRMN).

Si nous faisons une comparaison du premier schéma défini en utilisant le modèle TIGRE avec le schéma relationnel précédent, nous pouvons remarquer que le modèle relationnel est pauvre en description par rapport à un modèle sémantique comme TIGRE. Par exemple, les relations SCANNER et IRMN n'expriment pas par elles mêmes la dépendance sémantique qu'elles ont avec l'entité IMAGE_INF. De même, si l'on veut obtenir, par exemple, une information concernant une image du scanner, il faut faire mention des relations IMAGE_INF et SCANNER dans le schéma d'accès, pour pouvoir récupérer l'information complète. Dans le SGBD TIGRE il aurait suffi de mentionner l'entité SCANNER, car cette entité est une spécialisation de l'entité IMAGES, et hérite par conséquent de tous ses attributs.

Par ailleurs, les notions d'examen et de dossier ne sont plus explicitement représentées. Par exemple, le seul moyen que nous avons pour déterminer quels sont les objets d'un dossier, c'est le constituant *n_doss*. Comme celui-ci n'existe pas dans la relation PATIENT, il faut passer également à travers le constituant *pid*, pour arriver finalement à récupérer les éléments du dossier. Ainsi la récupération d'un dossier doit être explicite, c'est-à-dire, qu'il faut demander directement les constituants qui composent les objets du dossier. Il en est de même pour retrouver les objets participant à un examen; il faut déclarer explicitement les objets à récupérer.

Comme dans le modèle TIGRE, les images sont accédées à travers les données alphanumériques et les données textuelles qui lui sont associées. Les SGBD classiques n'offrent pas les outils nécessaires pour réaliser les recherches par contenu sur les données textuelles. Or, ceci est nécessaire dans notre application, pour accéder aussi bien aux comptes rendus qu'aux images. Il est donc impératif d'ajouter cette possibilité au modèle relationnel, pour pouvoir modéliser et implémenter notre application. Nous avons donc couplé une technique de *signature* avec le SGBD relationnel pour offrir l'accès par contenu aux données textuelles. Dans la section suivante nous faisons une description de cette technique et de la manière dont nous l'avons couplée avec le SGBD relationnel.

4.2. Technique de signature et couplage avec un SGBD relationnel.

4.2.1. Diverses méthodes d'accès aux données textuelles.

Les bases de données classiques gèrent d'une manière efficace des données bien structurées et de format bien défini, ce qui permet d'assigner facilement des valeurs aux attributs des enregistrements. Cependant, les données textuelles peuvent être difficilement structurées dans des valeurs avec un format précis pour les assigner aux attributs. L'approche généralement suivie par les SGBD relationnels concernant ces données, est de stocker le texte dans un attribut pouvant recevoir un grand nombre de données (par exemple certains SGBD relationnels offrent des attributs pouvant contenir jusqu'à 64koctets de mémoire), mais le problème que ceci entraîne est qu'il n'est pas possible d'accéder à ces attributs par leur contenu et, de plus, ils ne peuvent participer dans des opérations autres que la sélection.

La nécessité de pouvoir réaliser des recherches dans le contenu d'un texte est évident et dans ce but, diverses méthodes d'accès aux données textuelles ont été proposées dans la littérature. En général, ces méthodes peuvent être groupées en quatre classes: *balayage de tout le texte*, *inversion*, *accès multi-attribut*, et *regroupement*. Plus récemment, les filtres matériels ont fait leur apparition [HOLL. 79] et ils deviennent de plus en plus attractifs. Parmi ces techniques nous pouvons mentionner : *le filtrage par automates*, *les comparateurs parallèles* et *les comparteurs par cellules*. Nous faisons une brève description de chacune de ces classes, pour en mentionner les avantages et inconvénients. Tout au long de cette explication, nous appellerons "modèle de texte" une chaîne de caractères (pouvant contenir des jokers (*)) recherchée dans un texte.

* Balayage de tout le texte.

Les méthodes de ce groupe réalisent un balayage séquentiel des documents pour obtenir ceux qui contiennent le modèle de texte cherché. Si la requête donnée contient des opérateurs booléens concernants plusieurs modèles de texte (*modèle de texte AND modèle de texte*, etc.), alors la recherche des documents nécessite deux étapes. La première étape trouve les documents qui satisfont chacun des modèles, et la deuxième sélectionne, parmi ceux-là, ceux qui satisfont l'expression booléenne. Le principal inconvénient de ces types de méthodes est que la recherche peut durer longtemps (sauf dans le cas où un filtre matériel est utilisé). Malgré les efforts qui ont été faits pour trouver des algorithmes qui accélèrent la recherche, cet aspect

continue à être pénalisant surtout pour les grosses bases de données. Parmi les avantages on trouve surtout la facilité de la mise à jour des données. Comme il n'y a ni index ni pointeurs à changer, les insertions, les suppressions et les mises à jour sont des opérations triviales. De plus, ces méthodes n'ont pas besoin d'espace additionnel de stockage.

*** Inversion de termes.**

L'idée de base de ces méthodes est l'utilisation des index. Chaque document est représenté par une liste de mots clés qui décrivent le contenu du document. Ces mots sont associés aux documents au travers de pointeurs et forment avec eux le fichier d'index. Plusieurs systèmes commerciaux suivent cette approche [FALU 85] (STAIRS, DIALOG, MEDLARS, ORBIT). Les principaux avantages de ces méthodes sont le temps de réponse, la facilité d'implémentation et la facilité pour travailler avec des synonymes. Néanmoins, ces méthodes ont besoin d'un très grand espace mémoire pour stocker les index. Cet espace peut représenter de 50 à 300% de la place occupée par les documents. Ceci est un inconvénient auquel s'ajoute le coût des opérations de mise à jour des index.

*** Accès multi-attribut.**

Dans cette classe, sont regroupées toutes les méthodes utilisant le codage surimposé. Pour illustrer le principe de ce codage, considérons un enregistrement (par exemple le titre d'un livre) composé de trois valeurs d'attributs (trois mots clés) :

	attribut-1	attribut-2	attribut-3
enregistrement	<i>technique</i>	<i>codage</i>	<i>surimposé</i>

Pour chaque attribut, il existe un vecteur de bits de taille fixe F , lequel est initialisé à 0. Une fonction de hachage est appliquée à chaque valeur d'attribut dont le résultat indique la position du vecteur qui sera mise à 1. A la fin de cette opération, on aura un nombre m de bits qui ont été mis à 1. On fait ensuite le ou logique de ces vecteurs et on obtient la signature de l'enregistrement. Exemple :

technique 0 0 1 0 0 0 1
codage 0 0 1 0 0 1 0
surimposé 0 1 0 0 1 1 0

----- OU
signature 0 1 1 0 1 1 1

Diverses méthodes basées sur cette technique ont été développées. Parmi elles, nous pouvons mentionner la méthode de hachage multi-attribut et le fichier de signatures [GUST 71].

Dans la méthode de hachage multi-attribut la signature de chaque enregistrement est créée en utilisant le codage surimposé. Ensuite, une fonction *un à un* transforme chaque signature en une adresse dans une table de hachage. Chaque vecteur de taille F doit avoir au moins n bits mis à 1, où $n < F$. La combinaison possible de signatures nous indique le nombre de cases que doit contenir la table, et par conséquent le nombre d'adresses de signatures. Pour réaliser les recherches, les signatures de la requête sont calculées, puis leur adresse est recherchée dans la table de hachage. Une particularité de la méthode est que dans les requêtes conjonctives, le nombre de cases à parcourir dans la recherche décroît avec le nombre de termes participant dans la requête. Par exemple, étant donné une taille de signature de $F = 16$ devant avoir au moins 6 bits mis à 1 ($n = 6$) alors, il aura $com(16, 6) = 8008$ signatures différentes possibles et donc 8008 cases dans la table d'hachage. Ainsi, une requête avec un seul mot touche $comb(15, 5) = 3003$ cases de la table, une requête conjonctive avec deux mots touche $comb(14, 4) = 1001$ cases de la table, etc. La méthode présente de mauvaises performances au fur et à mesure que le nombre de signatures d'enregistrement croît, car la table de hachage croît de la même façon. Il est également assez coûteux de faire des mises à jour.

Dans la méthode du fichier de signatures, les signatures des enregistrements sont stockées séquentiellement dans un fichier indépendant. Dans les procédures de recherche, le fichier est parcouru jusqu'à trouver la signature cherchée. Les avantages de cette méthode sont la facilité d'implémentation, la facilité pour répondre à des requêtes contenant des mots avec jokers (ex. *radio*). De plus, les insertions sont des opérations triviales. Parmi les inconvénients, nous pouvons mentionner la diminution de la performance de la méthode avec de gros fichiers de signatures.

*** Regroupement ou "clustering".**

Le principe de ces méthodes est de regrouper logiquement et parfois physiquement les documents similaires, et former ainsi des groupes ou "clusters". Ceci est basé sur l'idée que les documents contenus dans un "cluster" ont une tendance à être pertinents pour la même requête et, de cette manière, on peut accélérer les recherches. L'importance de la méthode s'appuie plutôt sur des recherches par sujets généraux que par contenu. Un des inconvénients présenté par ces méthodes est que les insertions ne sont pas facilement gérées.

En résumé, parmi les méthodes mentionnées ici, c'est la méthode d'inversion qui semble être la plus rapide pour trouver un modèle de texte. Cependant, c'est la plus coûteuse en espace mémoire et en opérations de mise à jour. A l'opposé, parmi les méthodes les plus lentes en opération de recherche, on trouve la méthode de balayage de tout le texte. Les méthodes basées sur la technique de signature sont un compromis entre ces deux méthodes, car elles sont moins rapides dans la recherche que les méthodes d'inversion, mais moins chères en espace mémoire et en opérations de mise à jour. Elles sont les seules méthodes de recherche par contenu parmi celles mentionnées, qui permettent de répondre aux questions spécifiant des suffixes et des préfixes. De plus, ces méthodes peuvent être choisies lorsque les disques optiques sont utilisés comme moyen de stockage. En effet, étant donné qu'actuellement les disques optiques ne permettent pas la réécriture, les opérations de mise à jour dans une méthode comme celle d'inversion (où il faut faire une réorganisation des pointeurs et des index), deviendraient encore plus chères en termes d'organisation de stockage et d'espace mémoire utilisé.

C'est essentiellement pour ces raisons (et d'autres qui seront mentionnées plus loin), qu'une méthode de signature a été choisie pour réaliser les recherches par contenu dans notre application, tout comme dans le SGBD TIGRE.

*** Filtrage par automate.**

Cette approche transforme le modèle de recherche dans un automate d'états finis. Ce filtrage travaille comme un processeur spécialisé qui traite des données stockées sur un disque à la vitesse normale de transfert. Ainsi, les données lues, parcourent l'automate d'états finis jusqu'à ce qu'elles joignent un de ses états ou jusqu'à ce qu'elles arrivent à une sortie (en cas d'échec). Une table de transitions d'états enregistre des signaux pour les termes qui ont été reconnus. Cette approche peut gérer facilement des recherches de modèles de texte avec un nombre arbitraire de termes (limités à la taille de la table de transitions d'états).

*** Les comparateurs parallèles.**

Dans cette approche, les N termes à chercher sont chargés dans N comparateurs. Ensuite, on réalise parallèlement la lecture du texte et la comparaison avec les N comparateurs chargés. Quand il y a un comparateur qui joint le texte, un signal est produit comme réponse. A la fin, l'ensemble de signaux produits est testé par une table de vérité et les documents concernés sont sélectionnés. Un inconvénient de cette approche est qu'elle ne peut pas réaliser des recherches de termes contenant des jokers (*).

*** Les comparateurs par cellules.**

Cette approche utilise une cellule simple pour reconnaître un caractère à la fois. Les cellules sont interconnectées entre elles pour former le modèle de recherche. Ainsi, si le caractère qui arrive est égal à celui de la cellule, alors un signal est produit. Pour évaluer des expressions logiques, on connecte (soit en série, soit en parallèle) les modèles de recherche formés préalablement avec les cellules.

4.2.2. Une méthode d'accès basée sur la technique de signature

Nous faisons dans ce qui suit une description plus détaillée de deux variantes de la méthode de fichier de signatures dont une d'elles a été implémentée. Dans la méthode de fichier de signatures, on travaille avec deux fichiers, l'un contenant l'information textuelle et l'autre contenant l'ensemble des *signatures* créées (les *signatures* sont des index les textes auxquels elles sont associées). Cette méthode contient des avantages et des inconvénients, dont la plupart ont déjà été mentionnés. Parmi les divers avantages, on trouve [FALU 85], [RAZI 84] :

- * le fichier signature sur lequel on va réaliser les recherches occupe environ 10% de la place occupée par le fichier source.
- * des questions spécifiant des préfixes et des suffixes sont supportées d'une manière performante.
- * la facilité d'implémentation.

Parmi les inconvénients, nous pouvons citer le problème de "false-drop" qui consiste en la récupération d'une *signature* alors que le texte associé ne concerne pas la condition cherchée. La probabilité des "false-drop" dépend de la taille f de la *signature* et de la quantité d'information qu'elle représente. Dans [CHFA 84] on trouve un modèle mathématique pour le calcul de la taille optimale f de la

signature qui permet de réduire la probabilité de "false-drop" au minimum. Un autre problème que nous avons affronté est l'implémentation de requêtes négatives. Ce problème est lié au problème de "false-drop", car si l'on cherche à récupérer les *signatures* qui ne contiennent pas la *signature* cherchée, on risque de ne pas avoir toutes les réponses (c'est-à-dire toutes les *signatures* qui ont une collision avec la *signature* cherchée et, qui ne concernent pas le texte que l'on a nié).

Plusieurs variantes de cette méthode existent, toutefois nous décrivons deux: la méthode de "*signature par mot*" et la méthode de *signature par codage surimposé* [FALU 85].

4.2.2.1. Méthode de signature par mot.

La méthode de *signature par mot* consiste à appliquer à chaque mot significatif du document source une fonction de hachage, afin d'obtenir une chaîne de bits de longueur f . Cette chaîne, initialement mise à 0, contiendra à la fin un certain nombre de bits mis à 1.

Après avoir signé tous les mots d'un document, on applique une opération de concaténation de toutes les *signatures*, afin de créer la *signature du document*. Illustrons cette méthode par un exemple. Etant donné le document

tomodensitométrie du cancer des bronches

et une taille de *signature* de $f = 5$ préalablement calculée. On applique une fonction de hachage sur chacun des mots significatifs et on obtient:

Document	<i>tomodensitométrie</i>	<i>cancer</i>	<i>bronches</i>
<i>signature du mot</i>	10011	01010	01011

<i>signature du document</i>	100110101001011		

Pour réaliser une recherche par mot, on suit le même principe, c'est à dire que l'on applique la même fonction de hachage sur le mot de la requête pour obtenir sa *signature*. On compare ensuite cette *signature* à toutes les *signatures* des documents contenues dans le fichier de *signatures*, et on récupère celles qui contiennent la *signature* cherchée.

4.2.2.2. Méthode de signature par codage surimposé.

Dans la méthode de *signature par codage surimposé*, chaque document est divisé en blocs logiques. Un bloc logique est une partie du texte qui contient un nombre constant de mots significatifs et différents. Chaque mot est représenté par un minimum de m bits qui sont mis à 1 par une fonction de hachage. La *signature* du bloc est formée par le OU logique des *signatures* des mots.

Afin de permettre la recherche par mots contenant des jokers (*), on découpe chaque mot du bloc en triplets surimposés, en ajoutant un blanc au début et à la fin du mot.

Par exemple, le mot poumon génère 6 triplets :

po, pou, oum, umo, mon, on

("_" représente le caractère blanc)

On applique une fonction de hachage sur chacun des triplets et, à la fin de cette opération, on aura un nombre t de bits qui auront été mis à 1. Si $t < m$, on doit mettre $m - t$ bits à 1, en utilisant un générateur de nombres aléatoires. Cette génération doit être initialisée avec la valeur numérique du mot en question. Si, par contre, $t > m$, on garde tous les bits t dans la *signature*.

Pour réaliser la recherche d'un mot dans le document, on suit le même processus, c'est-à-dire que l'on découpe le mot de la requête en triplets surimposés. On applique la même fonction de hachage sur chacun des triplets pour obtenir ainsi la *signature* du mot. On examine ensuite, chaque *signature* du bloc et, une fois de plus, on récupère les *signatures* contenant la *signature* cherchée. Cette méthode suppose que des séquences de moins de trois caractères sont rarement spécifiées; dans le cas contraire, il y aura un grand nombre de blocs qui pourraient être qualifiés.

Reprenons l'exemple donné dans la méthode de *signature par mot*, pour illustrer la méthode de *signature par codage surimposé*. Nous allons supposer que chaque bloc logique est composé de trois mots, et que la taille f de la *signature* du bloc est de 28 bits. Ainsi, en appliquant une fonction de hachage à chaque triplet des mots du document "*tomodensitométrie du cancer des bronches*", on obtient.

signature de mot

tomodensitométrie	0 1 1 1 0 1 0 1 0 1 1 0 0 0 1 1 1 0 1 0 0 1 1 1 0 0 1 1
cancer	0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1
bronches	1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0

signature du bloc 1 1 1 1 0 1 0 1 0 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 0 0 1 1

Parmi ces deux méthodes, c'est cette dernière que nous avons décidé d'implémenter pour les raisons suivantes :

- la méthode annule automatiquement les mots doubles. Ceci, est du au fait que la *signature* du bloc se forme en faisant le *ou* logique des *signatures* de mots. Tandis que dans la méthode de *signature par mot*, la *signature* du document est formée en faisant la concaténation de chaque *signature* de mot;
- elle demande une très simple opération de bits dans la recherche d'information. Quand on cherche une *signature* à l'intérieur des documents, il suffi de faire un *ou* logique entre la *signature* cherchée et les *signatures* des documents. Tandis que dans la méthode de *signature par mot* il faut parcourir séquentiellement chaque bit de chaque *signature* pour repérer celles qui comprennent la *signature* cherchée;
- la recherche par contenu supporte d'une façon très performante, les requêtes spécifiant des préfixes et des suffixes, grâce à l'application d'une fonction de hachage sur chacun des triplets.

Une modalité que nous avons adoptée, est de prendre en compte le blanc (tabulateur ou interligne) qui sépare un mot d'un autre, c'est-à-dire que nous formons un triplet que nous appelons *triplet d'union*, avec le dernier caractère du mot précédant, le blanc, et le premier caractère du mot suivant. Ceci, dans l'idée de préserver au maximum l'ordre dans lesquels les mots se trouvent. Par exemple, pour le texte "veine cave", nous aurons les triplets suivants:

ve, vei, ein, ine, ne, e_c, _ca, cav, ave, ve_

Où "e_c" est le triplet d'union entre le mot *veine* et le mot *cave*. Pour la recherche des mots adjacents, il faudra que la *signature* soit créée en prenant en compte ce triplet d'union. Toutefois, il faut remarquer que le triplet d'union n'est pas unique et pourtant ce n'est pas un qualificatif déterminant pour retrouver les mots adjacents. C'est-à-dire, que différents mots peuvent générer le même triplet d'union.

4.2.3. Couplage de la méthode de fichier de signatures avec le SGBD relationnel.

Pour réaliser cette implémentation, nous n'avons pas modifié le SGBD relationnel, mais nous avons plutôt réalisé un couplage entre les deux méthodologies. Ce couplage a été effectué grâce à la possibilité de travailler en langage de haut niveau à partir du SGBD. De cette façon, les données textuelles avant d'être stockées dans la base de données, passent par la procédure de signature. Dans cette procédure, les données textuelles sont découpées en blocs logiques et signées. Un identificateur unique généré par le système est assigné à chaque signature créée et au bloc logique correspondant. Une fois ces opérations effectuées, les données textuelles seront insérées dans la base de données, où le n-uplet de la relation qui les reçoit contiendra un bloc logique, son identificateur, plus d'autres données qui éventuellement ont été associées.

De son côté, les signatures sont insérées avec leur identificateur respectif, dans le fichier de signatures. Ainsi, la communication entre le SGBD et le fichier de signatures est établie à travers l'identificateur. Etant donné que le fichier de signatures est externe à la base de données, nous devons passer par un langage de haut niveau pour avoir un environnement commun (cf. figure 4.2).

Les signatures ne sont pas stockées dans la base de données, simplement parce que ce sont des données binaires surimposées, et que le SGBD n'est pas capable de travailler à ce niveau de représentation, c'est-à-dire que le SGBD pourrait comparer deux signatures et déterminer si elles sont égales ou non, mais il ne serait pas capable de dire si une signature est contenue dans une autre. C'est pour cette raison que l'interrogation de la base de données se fait également en passant par le langage de haut niveau.

La technique de signature est, comme nous l'avons mentionné, une méthode d'accès aux données textuelles, mais il est clair qu'aucune analyse sémantique ni syntaxique ne participe à l'indexation. Ces aspects sont plutôt traités par les études sur le traitement du langage naturel. Cependant, les intégrer à une méthode d'indexation augmenteraient très sensiblement la puissance de l'outil. Nous avons implémenté la méthode de signature (couplée avec un SGBD relationnel c.f. 4.3. Fonctions et architecture du prototype réalisé), pour offrir un premier accès aux données textuelles, mais des travaux sont en cours de réalisation pour traduire les comptes rendus médicaux écrits en langage naturel, dans un langage cible [YAN 86], [BECI 87], [BCNM 87], [CINQ 87].

Le langage cible exprimerait d'une façon non ambiguë et uniforme les concepts médicaux et traiterait les problèmes de paraphrasage souvent trouvés dans les comptes rendus. Les termes médicaux qui apparaissent dans les comptes rendus font référence à des connaissances médicales qui appartiennent à plusieurs domaines (anatomie, physiologie, pathologie, etc). Il est souhaitable de pouvoir interroger la base de données à travers ces connaissances, car elles sont souvent d'un niveau conceptuel plus général que les informations analytiques ou synthétiques du compte rendu.

4.2.4. Prédicats de recherche par contenu.

Nous décrivons ensuite les prédicats des conditions de recherche par contenu que l'on pourra exprimer dans la clause *where* d'une requête. Ces conditions nous permettront d'identifier les données multimédia que l'on veut récupérer à partir des données textuelles. Pour ce faire nous allons utiliser un vocabulaire de symboles dont nous donnons l'interprétation. Tout d'abord, un *mot* est l'unité de recherche, il est toujours délimité par des parenthèses. Le joker * représente un suffixe ou un préfixe, c'est-à-dire qu'on cherche la chaîne de caractères donnée, ayant un suffixe ou un préfixe quelconque (tout dépend de la position de l'astérisque). Tout ce qui est contenu entre crochets [] est optionnel. La chaîne ADJ remplace le mot adjacent, et elle sera utilisée pour chercher deux *mots* adjacents.

Les prédicats doivent être délimités par des guillemets (" "). Dans les exemples donnés pour chaque prédicat, nous utilisons une relation fictive appelée DOCUMENT qui a un constituant de type *long* appelé *texte* qui contient des données textuelles. Le but de ces exemples n'est pas de montrer le chemins d'accès aux données textuelles, mais d'illustrer le format du prédicat de recherche.

4.2.4.1. Prédicats disjonctifs (OR).

" ([*]mot[*]) OR ([*]mot[*]) [OR ([*]mot[*]) OR ([*]mot[*])] "

Sa fonction est de chercher les documents contenant un ou plusieurs mots donnés. Les mots peuvent être tapés de différentes façons :

- masqués à l'une des deux extrémités du mot (recherche des mots avec un préfixe ([*]mot) ou un suffixe (mot[*])) quelconque,
- masqués aux deux extrémités (recherche des mots avec un préfixe et un suffixe quelconques [*]mot[*]);
- sans masque (mot).

Exemple

```
select texte
from DOCUMENT
where " (foie) OR (hepatomega* ) " ;
```

Cette requête cherche les documents contenant les mots *foie* ou les mots *hepatomega*, où le mot *hepatomega* peut avoir un suffixe quelconque.

Pour répondre à ce type de requêtes, le processus crée une *signature* pour chaque mot donné dans le prédicat. Chaque *signature* est comparée aux *signatures* existantes dans le fichier. Ce type de prédicat implique un balayage séquentiel du fichiers de *signatures* pour chaque mot contenu dans le prédicat. Dans chaque passe on récupère l'identificateur de la *signature* qui contient la *signature* cherchée. Ensuite, on utilise ces identificateurs pour récupérer les blocs logiques correspondants lesquels contiennent le texte cherché.

4.2.4.2. Prédicats conjonctifs (AND).

```
"([*]mot1[*]) AND ([*]mot2[*]) [AND ([*]mot3[*])...AND ([*]motn[*]) |"
```

Sa fonction est de chercher les textes contenant tous les mots donnés dans le prédicat. Le texte récupéré peut contenir n'importe quelle chaîne de caractères entre les mots spécifiés.

Exemple:

```
select texte
from DOCUMENT
where " (cancer) AND (poumon) " ;
```

Cette requête récupère les documents contenant les mots *cancer* et *poumon*.

Tous les mots de cette requête forment une seule *signature* de recherche, à la différence de la requête précédente où une *signature* est créée pour chaque mot. De cette manière, le fichier de *signature* n'est balayé qu'une seule fois pour récupérer les identificateurs des *signatures* contenant la *signature* cherchée, et obtenir (dans le cas où ils sont demandés) les blocs contenant le texte

4.2.4.3. Prédicats adjacents (ADJ).

" ([*]mot1) ADJ (mot2[*]) "

ou

" ([*]mot1) ADJ (mot2) [ADJ (mot3) ADJ (motn-1) ADJ (motn[*])] "

Cette requête cherche les mots adjacents (ADJ) dans un bloc logique. Le principe de recherche des *signatures* est le même que celui des prédicats conjonctifs (AND), la seule différence est que les prédicats adjacents (ADJ) prennent en compte le triplet d'union pour la formation de la *signature* à chercher. On pourra remarquer que l'utilisation du joker "*" est limitée à l'extrémité gauche du premier mot et à l'extrémité droite du dernier mot du prédicat.

Ce type de prédicats est possible grâce à l'enregistrement du triplet d'union < dernier caractère du mot1 + blanc + premier caractère du mot2 >. Etant donné que ce triplet d'union prend en compte le blanc (de même que le tabulateur ou le caractère d'interligne), l'utilisation du joker "*" doit être limitée de façon à ne pas interférer la construction de triplets d'union, c'est-à-dire qu'un joker "*" ne peut pas être utilisé à l'endroit où un triplet d'union peut exister.

Exemple

```
select texte
from document
where " (veine) ADJ (cave) ADJ (supérieure) " ;
```

La requête récupère les documents contenant la chaîne *"veine cave supérieure"*. Néanmoins, il faut remarquer que, dans ce type de prédicats, la probabilité de "false-drop" n'est pas forcément diminuée, parce que le triplet d'union et les mots

cherchés peuvent apparaître dans la phrase, sans pour cela répondre aux besoins de recherche. Prenons un exemple pour montrer ceci :

Supposons qu'on cherche la chaîne "veine cave supérieure". Dans ce texte, les triplets d'union sont:

"e_c" et "e_s"

Supposons en plus qu'il existe un document qui contient le texte "tumeur opprimant la veine cave derrière le lobe supérieur". Bien que dans ce texte la chaîne "veine cave supérieure" n'existe pas, il sera quand même récupéré. En effet, tous les mots cherchés, ainsi que les triplets d'union existent dans le texte; toutefois, le triplet d'union "e_s" joint les mots *cave supérieure* dans le premier cas et *lobe supérieur* dans le second.

4.2.4.4. Prédicats composés.

Les trois prédicats que nous venons de décrire peuvent apparaître ensemble dans une même clause *where*. Ils doivent être liés par un opérateur booléen AND ou OR. Pour distinguer la fin d'un prédicat et le commencement du suivant, nous utilisons des parenthèses. La recherche des textes qui satisfont la condition donnée se fait de la manière suivante :

il faut récupérer d'abord les identificateurs des *signatures* qui satisfont la condition de chaque prédicat participant dans le prédicat composé (ceci se fait de la même manière que pour les prédicats). Ensuite, nous travaillons au niveau des identificateurs des *signatures* pour appliquer les opérateurs booléens qui lient les prédicats. Cette dernière opération nous permet de déterminer les identificateurs des *signatures* qui répondent à la condition donnée et de les utiliser par la suite pour récupérer les blocs de texte correspondants (dans le cas où ils ont été demandés).

Ainsi, un prédicat composé peut avoir la forme suivante :

"(prédicat) op-bool (prédicat) [op-bool (prédicat) op-bool (prédicat)] "

où

prédicat peut être l'un des trois prédicats décrits préalablement.

op-bool est l'un de deux opérateurs booléens AND ou OR.

Exemple

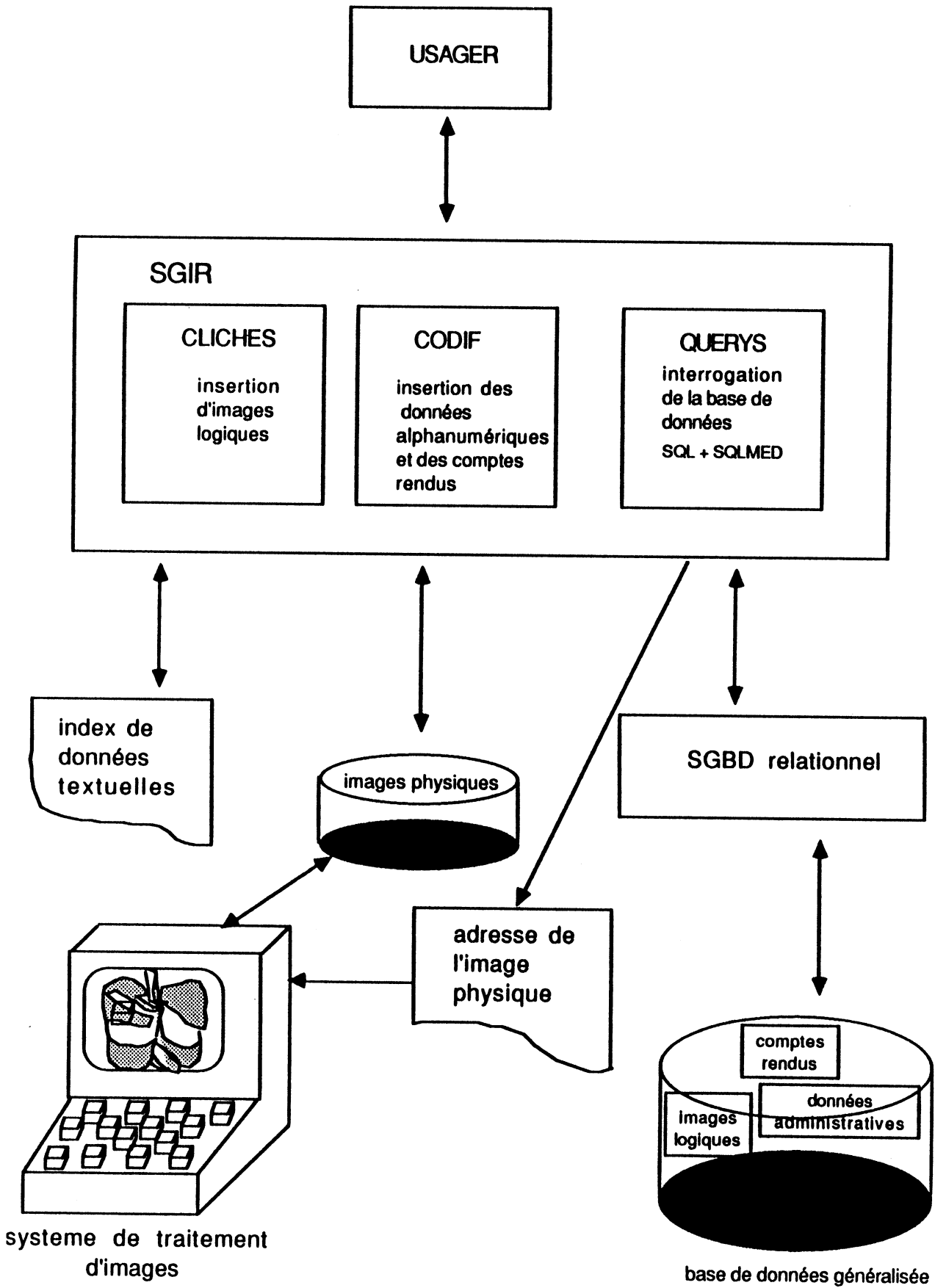
```
select texte
from DOCUMENT
where " ((tumeur) ADJ (cancéreuses)) AND ((poumon) OR (bronches)) " ;
```

Cette requête cherche les documents qui parlent de *tumeurs cancéreuses* mais qui contiennent également le mot *poumon* ou le mot *bronches*. Ce type de prédicats permet donc d'exprimer des requêtes assez complexes afin de réaliser une recherche par contenu sur les données textuelles.

4.3. Fonctions et architecture du prototype réalisé.

Le prototype que nous avons réalisé comprend le schéma relationnel décrit dans la section précédente. Deux types d'objets complexes sont manipulés: les images et les comptes rendus. Comme nous l'avons mentionné précédemment, les images sont divisées en images logiques et images physiques. Elles sont récupérées à travers les données textuelles et les données alphanumériques qui leur sont associées. Pour réaliser l'accès aux données textuelles, nous avons couplé le SGBD relationnel utilisé, avec la méthode de signature par codage surimposé.

Nous avons donc implémenté une couche multimédia au-dessus du SGBD qui permet à l'utilisateur de travailler à un même niveau avec les données alphanumériques (gérées par le SGBD) et les données textuelles (indexées par la méthode de signature). Cette couche, s'appelle le SGIR pour "Système de Gestion d'Information Radiologique". Le SGIR est composé d'une série de modules écrits dans le langage de programmation C, contenant des appels au SGBD. C'est à travers le SGIR que l'utilisateur va interroger et mettre à jour la base de données (cf. figure 4.1).



Architecture générale du système SGIR

figure 4.1

Les modules CLICHES et CODIF permettent la mise à jour de la base de données. Ils réalisent les insertions des données concernant les images, les patients, les comptes rendus, les demandes d'examens, ainsi que les données concernant les médecins demandeurs d'examens. Pour sa part, le module QUERYS permet l'interrogation de la base de données.

Un langage d'interrogation propre à l'application a été développé; c'est un langage basé sur le langage relationnel SQL. Toutefois, il diffère de celui-ci sur les points suivants:

- * des mots clés ont été introduits dans la clause *select* pour manipuler les objets de la base. Un objet est formé d'un ensemble de constituants (voir figure 4.2). Chaque mot clé correspond donc à un objet de la base, et son utilisation implique la récupération de l'objet auquel il fait référence.
- * la clause *from* du langage SQL disparaît. Ainsi, l'utilisateur n'est plus concerné par les trajets d'accès de la requête. Il doit seulement spécifier les différents noms des constituants qu'il veut sélectionner (clause *select*), et les conditions de restriction à utiliser (clause *where*).
- * la clause *where* peut comprendre les prédicats SQL classiques et les prédicats de restriction donnés dans la section précédente. En outre, l'utilisateur ne doit plus donner les prédicats de jointure utilisés dans une requête SQL.

Nous avons appelé ce langage SQLMED, parce qu'il dérive du langage SQL, et qu'il est destiné à notre application médicale.

SQLMED permet de manipuler les données alphanumériques et les données multimédia d'une manière intégrée. Ainsi, le module QUERYS peut recevoir une requête SQLMED (qui est transformée en requête SQL) ou une requête SQL pure. QUERYS dispose donc, de toute la puissance du langage relationnel SQL, plus les avantages que SQLMED apporte pour manipuler les données multimédia. De plus, l'absence dans SQLMED de trajets d'accès aux données et de prédicats de jointure, fait de ce langage un outil plus convivial pour les utilisateurs non informaticiens, comme c'est souvent le cas des médecins, qui sont les utilisateurs potentiels de ce système. Les trajets d'accès et les jointures sont générés par le module QUERYS à partir des objets et/ou des constituants donnés par l'utilisateur dans les clauses *select* et *where* (c.f. 4.5.3. Le module QUERYS).

Ainsi, la vue que l'utilisateur a de la base de données, ne comprend pas tout le schéma défini préalablement, mais un sous-ensemble que nous représentons dans la figure 4.2.

OBJETS DE LA BASE DE DONNEES

VUE DE L'USAGER

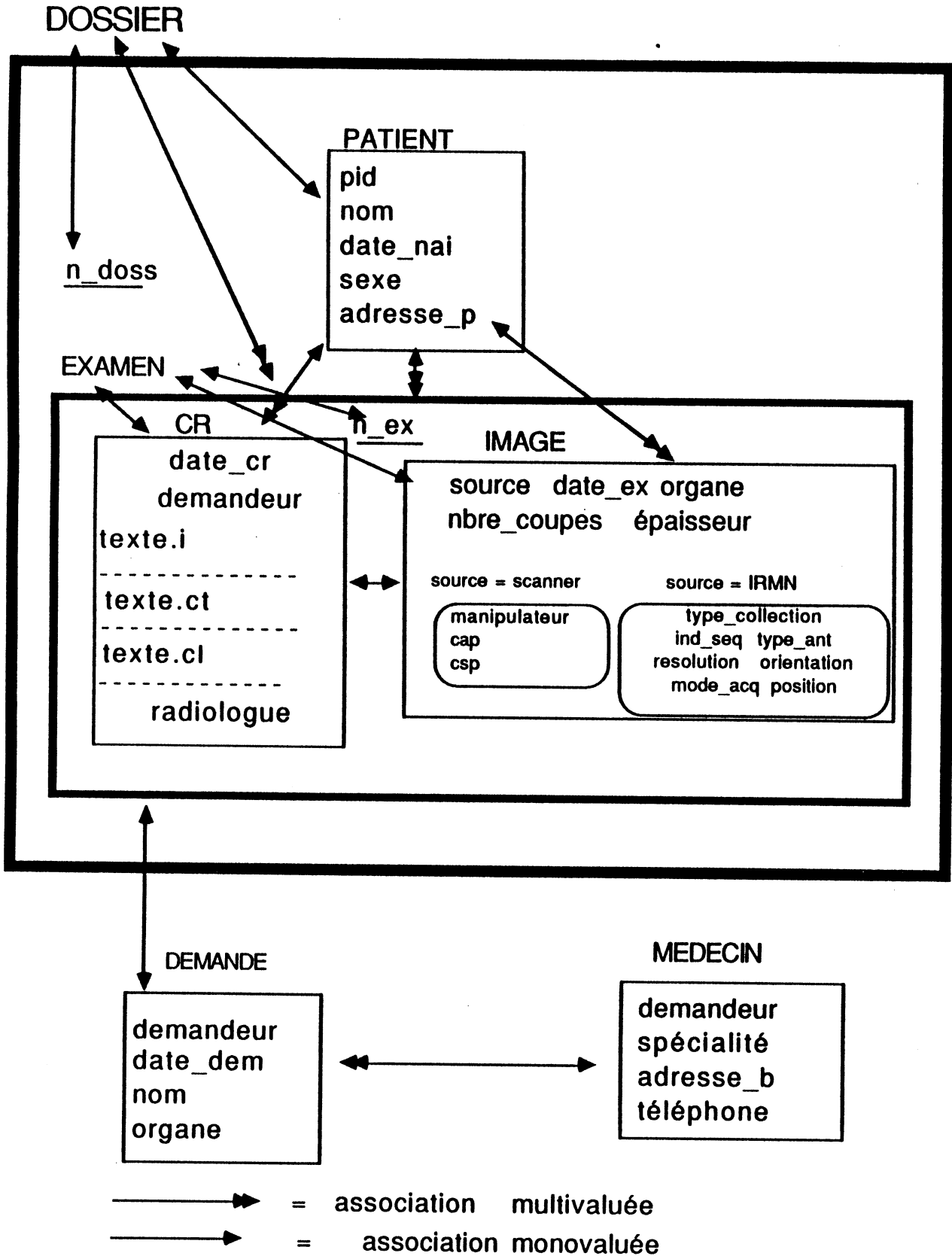


figure 4.2

La vue de l'utilisateur est donc formée de 7 objets qui sont :

* Le DOSSIER : un DOSSIER est composé d'un objet PATIENT, d'un ou plusieurs objets EXAMEN, et d'un constituant de type entier qui contient le numéro de dossier (*n_doss*). L'utilisation du mot clé *@dossier* dans la clause *select* de la requête, implique la récupération de tous les éléments qui le composent.

* L'EXAMEN : un EXAMEN est composé d'un objet COMPTE_RENDU, d'un objet PATIENT, et d'un constituant de type entier qui contient le numéro d'examen (*n_ex*). L'utilisation du mot clé *@examen* dans la clause *select*, implique la récupération des éléments qui le composent. Un examen ne peut appartenir qu'à un PATIENT et à un DOSSIER.

* Le PATIENT : un patient est un objet composé de 5 attributs dont :

pid = identificateur du patient,
nom = nom du patient,
date_nai = date de naissance du patient,
sexe = sexe du patient,
adresse_p = adresse personnelle du patient.

Un patient ne peut être contenu que dans un seul DOSSIER. L'utilisation du mot clé *@patient* dans la clause *select* implique la récupération des 4 premiers constituants. L'adresse du patient n'est récupérée qu'explicitement en donnant le nom du constituant dans la clause *select*. En effet, lorsque l'utilisateur récupère une image ou un compte rendu, il s'intéresse souvent au nom du malade, à son sexe, ou à sa date de naissance. Son adresse personnelle n'a pas le même niveau d'intérêt.

* Le COMPTE_RENDU : un COMPTE_RENDU est l'objet qui représente la structure physique d'un compte rendu. Il est composé de trois constituants de type alphanumérique :

date_cr = date du compte rendu,
demandeur = nom de médecin qui a demandé l'examen,
radiologue = médecin qui rédige et signe le compte rendu,

plus une partie textuelle composée de :

une introduction (récupérée avec le mot clé *@texte.i*),
des constatations (récupérées avec le mot clé *@texte.ct*),
une conclusion (récupérée avec le mot clé *@texte.cl*).

Le mot clé *@text* doit être utilisé pour récupérer la partie textuelle complète d'un compte rendu, tandis que le mot clé *@texte.** est utilisé pour récupérer une phrase du compte rendu, sans prendre en compte la section auquel elle appartient. Le mot clé *@cr* permet de récupérer le compte rendu complet.

* **L'IMAGE** : l'objet IMAGE représente toutes les informations concernant les images logiques. L'utilisation du mot clé *@image* implique la récupération des images en général, sans considérer les données liées aux techniques d'obtention. Ces dernières sont récupérées explicitement en donnant les noms des constituants concernés. Un objet CR et un objet IMAGE ont une relation unique entre eux, et n'appartiennent qu'à un seul PATIENT, à un seul EXAMEN, et à un seul DOSSIER.

* **Le MEDECIN** : l'objet MEDECIN contient toutes les informations administratives concernant les médecins demandeurs d'examens. Il est composé de 4 attributs :

demandeur = nom du médecin qui demande l'examen,
spécialité = spécialité du médecin demandeur,
adresse_b et téléphone = adresse et téléphone professionnel du demandeur.

Le mot clé *@médecin* implique la récupération de ces constituants. Un médecin peut demander un ou plusieurs examens.

* **La DEMANDE** : l'objet DEMANDE représente l'information concernant la demande d'examen. Il est composé de 4 constituants :

demandeur = nom du médecin qui demande l'examen,
date_dem = date à laquelle l'examen a été demandé,
nom = nom du patient qui doit suivre l'examen,
organe = organe à explorer. Le mot clé *@demande* implique la récupération de cette information. Une demande ne concerne qu'un examen, et elle ne provient que d'un médecin.

Ainsi, quand l'utilisateur veut récupérer un objet complet, il fait appel au mot clé qui lui est associé. Si par contre il ne s'intéresse qu'à une partie de l'objet, alors, il utilise les noms des constituants concernant l'information qu'il veut récupérer, sans faire mention de l'objet auquel ils appartiennent.

Dans l'objet COMPTE_RENDU, le mot clé *@texte* est le mot utilisé pour manipuler les données textuelles, soit pour les récupérer, c'est-à-dire soit pour les utiliser dans un prédicat de restriction. Ainsi, l'utilisateur sélectionne les objets ou les parties des objets donnant les noms correspondants, sans s'occuper des constituants qui font les liens entre eux (prédicats de jointure), ni de mentionner les trajets pour les accéder.

Une requête SQLMED a la syntaxe suivante :

```
select [@dossier][, @examen][, @patient][, @cr]
      [, @texte.[*][i][ct][cl]][, @image][, @scanner]
      [, @irmn][, @médecin][, @demande]
      [, constituant...,constituant]
[ where [ @texte = prédicat ] [op-bool prédicat classique de SQL] ] ;
```

Les crochets [] représentent des termes optionnels.

Clause *select*:

Au moins l'une des options données dans la clause *select* est obligatoire. Les options ont la signification suivante:

@dossier : permet la récupération des éléments composant un dossier radiologique constitué d'un *patient*, des *images* et des *comptes rendus*.

@examen : cette option est un sous-ensemble de l'option précédente, elle permet la récupération des éléments composant un examen constitué des *images* et des *comptes rendus*.

@patient : permet de récupérer les données qui décrivent un patient.

@cr : permet de récupérer un compte rendu complet.

@texte : permet de récupérer un objet de nature textuelle.

Quatre options peuvent accompagner le mot clé *@texte* :

[.*] : indique que l'on veut récupérer seulement un bloc logique de texte.

[.i], [.ct], [.cl] : indiquent le titre de la section du compte rendu que l'on veut récupérer. Ainsi, ".i" sélectionne l'introduction, ".ct" les constatations, et ".cl" les conclusions.

@image : permet de récupérer une image complète.

@scanner : permet de récupérer toute l'information concernant une image, plus les données spécifiques à la technique du scanner.

@irmn : permet de récupérer toute l'information concernant une image, plus les données spécifiques à la technique d'imagerie par résonance magnétique (irmn).

@médecin : permet de récupérer les données concernant le médecin qui fait la demande d'examen.

@demande : permet de récupérer les informations concernant la demande d'un examen, constitué par une date, un demandeur, un patient à examiner, et un organe à explorer.

constituant : c'est le nom d'un attribut alphanumérique qui fait partie d'un des objets donnés dans la figure 4.2. Il est utilisé pour récupérer une partie d'un objet.

Clause *where*:

La clause *where* est optionnelle. Elle est utilisée pour restreindre les faits qui ont été demandés dans la clause *select*. Elle a comme paramètre optionnel deux types de prédicats de restriction : un prédicat de restriction sur les données textuelles, et un prédicat de restriction classique du langage SQL. Les termes contenus dans la clause *where* ont la signification suivante :

@texte = prédicat : @texte est le mot clé qui indique la présence d'une recherche par contenu. La restriction de la recherche est donnée dans "prédicat" où "prédicat" peut être l'un des quatre prédicats décrits dans la section précédente.

op-bool : est l'un des opérateurs booléens manipulés par le SGBD.

Un prédicat classique de SQL est une condition de restriction dans laquelle interviennent:

- un constituant alphanumérique de la vue que l'utilisateur a du schéma de la base de données,
- un opérateur arithmétique (=, >, <, !=), ou un opérateur de chaînes de caractères (like),
- une condition de restriction, cohérente avec l'attribut en question.

La requête doit être finie par un point virgule (;).

Nous donnons ici un exemple de requête SQLMED :

```
select @patient, @cr, organe
where @texte = " (cancer) AND (bronches) ";
```

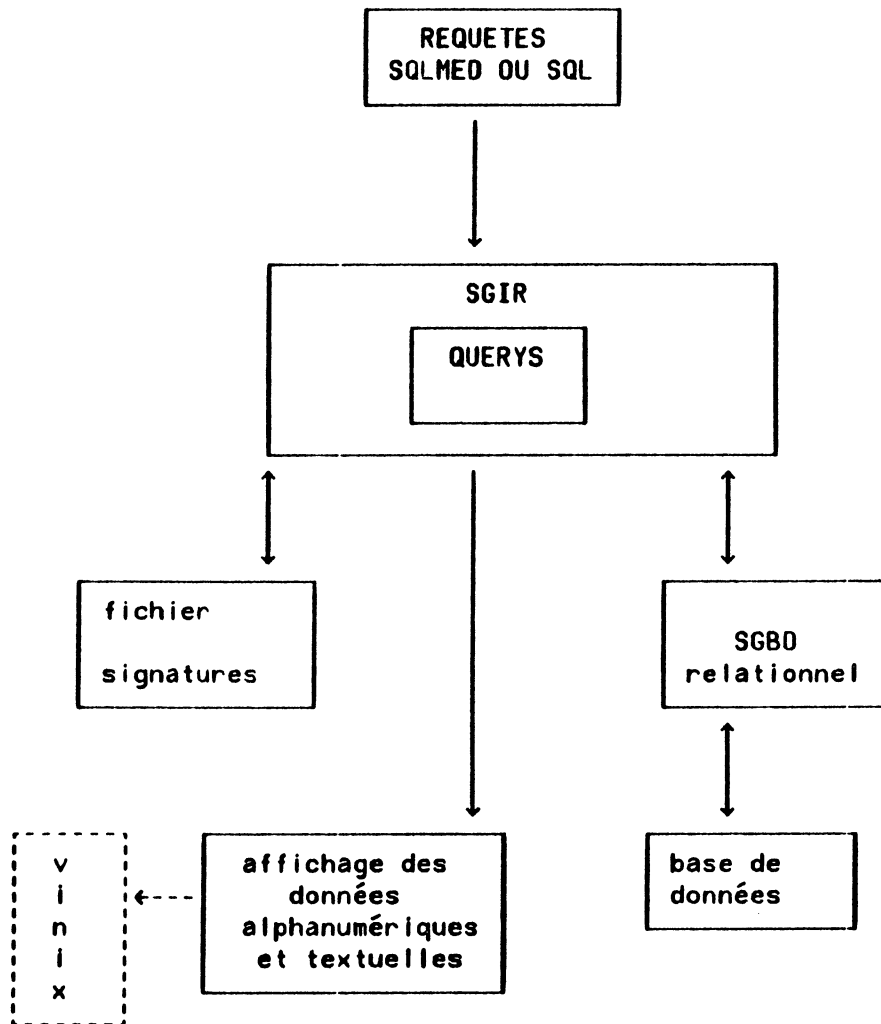
Cette requête a pour but de sélectionner tous les patients et leurs comptes rendus qui contiennent les mots *cancer* et *bronches*, ainsi que l'organe examiné.

4.3.1. Interrogation de la base de données.

Comme nous l'avons déjà mentionné, l'interrogation de la base de données se fait à travers le module QUERYS. Ce module peut recevoir des requêtes SQL et de requêtes SQLMED. Lorsque QUERYS reçoit une requête SQLMED, il effectue une analyse pour détecter les demandes de l'utilisateur. S'il y a des prédicats de restriction sur les données textuelles, alors les données concernées sont signées dans la procédure de signature. Les signatures produites sont comparées avec les signatures existantes. Ensuite, les identificateurs de signatures qualifiant le prédicat de restriction de la requête sont récupérés.

Après cette étape, QUERYS transforme la requête SQLMED en une requête SQL pur. Autrement dit, les mots clés qui opèrent la sélection des objets, sont transformés en énoncés de sélection du SQL pur. Ensuite, les prédicats de restriction sur les données textuelles sont remplacés par des prédicats de restriction classiques contenant le numéro d'identification de la signature associée. Puis, la clause *from* est constituée à partir du trajet d'accès correspondant. Enfin, les prédicats de jointure sont ajoutés à la clause *where* de la requête.

La requête ainsi construite est envoyée au SGBD relationnel qui l'exécute et retourne sa réponse au module QUERYS. Ce dernier assure l'affichage des données en accord avec les demandes de l'utilisateur. Lorsque QUERYS reçoit une requête SQL pur, elle est envoyée directement au SGBD relationnel, sans faire aucune transformation. La figure 4.3 montre l'aspect général de la procédure d'interrogation.



vinix = système de traitement d'images

figure 4.3

Pour afficher et réaliser les opérations sur les images, un système de traitement d'images est intégré à l'architecture présentée dans la figure 4.3. Actuellement, cette intégration est manuelle. Lorsque QUERYS détecte le mot clé *@image*, il affiche comme résultat de la requête une série de données alphanumériques concernant l'image et, il crée en parallèle un fichier contenant les "path-names" des fichiers des images physiques sélectionnées. Ce fichier est donné comme paramètre d'entrée au système de traitement d'images, pour permettre l'affichage ou la modification de l'image.

C'est en fait à ce moment là que l'utilisateur peut décider de rendre l'image indépendante. Il fait alors connaître au système son intention, et rentre les informations (les commentaires, la fonction appliquée, etc) qui serviront à identi-

fier l'image. Cette étape est également manuelle pour l'instant, car l'utilisateur ne peut pas appeler le système de traitement d'images à partir du QUERYYS. Ainsi, pour instancier l'image dans la base de données, il doit explicitement passer du système de traitement d'images au SGIR, et entrer parmi les données de l'image indépendante, le nom du fichier qui la contient.

4.3.2. Mise à jour de la base de données.

La mise à jour de la base de données se fait aussi par l'intermédiaire du SGIR. Les opérations de mise à jour comprennent l'insertion, la modification et la suppression de données.

4.3.2.1. Insertion de données.

L'insertion de données dans la base se fait à travers le système SGIR. En effet, les données textuelles doivent passer préalablement par la procédure de signature. Une fois les données signées, les insertions sont réalisées dans la relation correspondante. Les contraintes d'intégrité vérifiées sont celles qu'offrent normalement les SGBD classiques, telles que le type de donnée, l'unicité de la clé, la taille des champs, et les valeurs non nulles. Comme nous l'avons mentionné précédemment, les insertions sont réalisées par le module CLICHE (pour les insertions des images logiques) et le module CODIF (pour les insertions de toutes les autres données). Une description détaillée de ces deux modules est donnée au paragraphe 4.5.1 et 4.5.2.

4.3.2.2. Modification de données.

Les modifications concernent les changements de valeurs dans les constituants. Toutes les modifications peuvent se réaliser avec le langage de manipulation (SQL) du SGBD. Cependant, il est impossible de modifier les champs de type *long* et les identificateurs correspondants. Ces champs contiennent les données textuelles, et leur mise à jour implique une modification du fichier de signatures. Ainsi, pour modifier un compte rendu ou un commentaire, il faut d'abord repérer le bloc logique à changer, l'effacer de la base de données, puis, effacer sa signature du fichier en utilisant l'identificateur du bloc. On rentre ensuite le nouveau bloc qui est immédiatement signé et inséré dans la base. En résumé, une modification consiste en une suppression et une insertion.

4.3.2.3. Suppression de données.

Comme dans la modification, la suppression d'un champ de type *long* implique aussi la suppression de la signature correspondante dans le fichier de signatures.

4.4. Exemple de requêtes.

Dans les lignes suivantes nous reprenons quelques questions parmi celles développées dans la section 3.5. Ici, les requêtes correspondantes sont écrites dans le langage SQLMED. Pour les deux premières, nous donnons la requête SQL qui nous permet de récupérer l'information dans la base de données, afin de mettre en évidence les avantages de SQLMED apportés par rapport à SQL. Dans les deux requêtes SQL, nous supposons que l'identificateur de la signature a été préalablement récupéré, et nous utilisons un numéro quelconque pour le représenter.

Exemple 1.

Donner les comptes rendus et les images concernant les processus expansifs découverts dans la loge de Baréty.

SQLMED :

```
select @cr, @image  
where @texte = " (processus) AND (expansifs) AND (loge) AND (Baréty) " ;
```

Les mots clés *@cr* et *@image* demandent au système la récupération des comptes rendus complets et des images associées qui satisfont la condition donnée.

SQL

```
select c.n_ex, c.pid, c.n_doss, date_cr, demandeur, radiologue,  
       paragraphe, texte, date_ex, source, organe, nbre_coupes,  
       épaisseur, taille, unité, nom_fichier  
from COMPTE_RENDU c, IMAGE_INF i, TEXTE_CR t  
where c.n_ex = i.n_ex and  
       c.n_ex = ( select n_ex from TEXTE_CR where id = 5)  
order by id ;
```

Une telle requête SQL effectue un affichage par n-uplet des résultats obtenus. Chaque n-uplet contient une grande quantité de constituants alphanumériques, et un constituant (qui contient le texte) de type *long*. Ceci ne permet pas de visualiser un n-uplet complet sur une même ligne. De plus, le texte du compte rendu sera coupé et mélangé aux données alphanumériques. Il est donc évident, que ce type d'affichage ne convient pas à l'utilisateur. C'est ici que QUERYS intervient en faisant d'abord un affichage cohérent des données alphanumériques, puis des données textuelles.

Exemple 2.

Donner les patients, la conclusion des comptes rendus et les images où l'on peut observer des processus expansifs de densité tissulaire dans la loge de Baréty.

SQLMED :

```
select @patient, @texte.cl, @image
where @texte = " ((processus) ADJ (expansifs)) AND
                ((densite) ADJ (tissu!aire)) AND (loge) AND (Baréty) " ;
```

SQL :

```
select p.pid, nom, date_nai, sexe, texte, i.n_ex, n_doss,
       source, organe, nbre_coupes, épaisseur, date_ex,
       taille, unité, nom_fichier
from PATIENT p, IMAGE_INF i, TEXTE_CR t
where p.pid = i.pid and
       i.n_ex = (select n_ex from TEXTE_CR where id = 15) and
       paragraphe like 'conclusion'
order by id;
```

Les trois derniers constituants de la clause *select* (*taille*, *unité* et *nom_fichier*) donnent l'information nécessaire à l'affichage de l'image physique.

Exemple 3.

Combien de thymomes a-t-on observé parmi les processus expansifs de densité tissulaire dans la loge de Baréty?

Cette requête compte les occurrences des comptes rendus vérifiant la condition donnée :

```
select count (n_ex)
where @texte = "(tymomes) AND ((processus) ADJ (expansifs)) AND
((densité) ADJ (tissulaire)) AND (loge) AND (Baréty) " ;
```

Exemple 4.

Récupérer les comptes rendus radiologiques où l'on a observé des anévrismes de la veine azygos de monsieur LEPETIT, ainsi que la spécialité de son médecin traitant.

```
select spécialité, @cr
where @texte = "(anévrisme*) AND ((veine) ADJ (azygos)) "
AND nom like 'LEPETIT' ;
```

Cette requête est un exemple de la cohabitation d'un prédicat classique SQL avec un prédicat textuel.

Exemple 5.

Donner les nom des patients, leur date de naissance, leurs comptes rendus et leur images où l'on a observé des anévrismes dans le médiastin antérieur.

```
select nom, date_nai, @cr, @image
where @texte = "(anévrisme*) AND (médiastin) AND (antérieur) " ;
```

Exemple 6.

Combien d'examens du scanner a fait le docteur LEGRAND entre le 1er. Janvier 1986 et le 15 Janvier 1986.

```
select count (n_ex)
where source = 'scanner' and radiologue like 'LEGRAND' and
date_ex >= '01.01.86' and date_ex <= '15.01.86' ;
```

Exemple 7.

Montrer les images du scanner qui ont une densité hydrique après injection intra-veineuse de produit de contraste.

Cette requête peut être donnée de deux façons différentes. La première

consiste à chercher, dans le compte rendu, le fait "injection intra-veineuse de produit de contraste". La deuxième consiste à chercher cet aspect dans les constituants de la relation SCANNER. Ainsi :

```
1.- select @image, manipulateur, cap, csp
      where @texte = " (densité) AND (hydrique) AND (après)
                    (injection) AND (*veine*) AND (produit) AND (contraste) "
                    AND source = 'scanner' ;
```

```
2.- select @image, manipulateur, cap, csp
      where @texte = " (densité) AND (hydrique) " AND
                    source = 'scanner' AND cap > 0 ;
```

cap est le nom de constituant qui contient le nombre de coupes qui ont été réalisées avec l'injection de produit de contraste.

Exemple 8.

Donner les numéros d'examen, les numéros de dossier et les conclusions des comptes rendus où sont observés des kystes broncho-géniques du médiastin postérieur qui sont calcifiés.

```
select n_ex, n_doss, @texte[.cl]
      where @textete = " (kyste*) AND (broncho-génique*) AND
                    (médiastin) AND (postérieur) AND (calcifié*) ;
```

Exemple 9.

Donner les images de scanner du poumon où l'on a utilisé le produit de contraste.

```
select @image, manipulateur, cap, csp
      where source = 'scanner' and organe = 'poumon' and cap > 0 ;
```

Exemple 10.

Donner le dernier examen subi par monsieur Jack DUPONT.

Cette requête fait intervenir la notion de version d'un objet. Elle ne peut être

traitée directement ni par le modèle relationnel, ni par notre système SGIR. Pour répondre à cette requête, il faut récupérer les dates et les numéros d'examens, et faire un tri décroissant par date. Puis, prendre par programme la première occurrence de la liste (celle qui correspondrait donc à la date du dernier examen). Avec le numéro d'examen correspondant, nous pourrions ensuite obtenir le compte rendu et l'image concernés.

```
select date_ex, n_ex  
where nom like "DUPONT"  
order by date_ex DESC ;
```

cette deuxième partie est générée par programme

```
select @examen  
where n_ex = :n_ex ;
```

où ":n_ex" est une variable qui contient la première valeur de la liste de données envoyée par la première requête.

Exemple 11.

Donner le dossier de monsieur LEPETIT.

```
select @dossier  
where nom like 'LEPETIT' ;
```

Exemple 12.

Donner les examens qui ont été demandés par le docteur 'PARAMELLE' depuis le 10 mai 1987.

```
select @examen  
where demandeur like 'PARAMELLE' and date_dem > '10-05-87' ;
```

4.5. Réalisation.

Nous utilisons ORACLE [ORAC 84] comme SGBD relationnel, qui a SQL comme langage de manipulation.

ORACLE offre diverses possibilités pour travailler avec le langage SQL et par conséquent, pour manipuler la base de données :

* SQL interactif : l'utilisateur entre en contact avec la base de données en utilisant directement et uniquement le langage SQL.

* Pro*SQL : l'utilisateur utilise une série de constructeurs insérés dans un langage hôte (COBOL, FORTRAN, C, PL/I). Les constructeurs sont des appels aux sous-programmes qui permettent l'écriture d'applications en combinant les avantages du langage hôte et ceux du langage SQL. Ainsi, ORACLE peut être interfacé avec les langages procéduraux via Pro*SQL.

* Pro*ORACLE : l'utilisateur peut introduire directement les énoncés de SQL dans un langage de programmation procédural de haut niveau (COBOL, FORTRAN, C, PL/I). Un précompilateur transforme en instructions du langage hôte, tous les énoncés SQL retrouvés dans le programme, et génère un fichier qui peut être ensuite compilé et exécuté normalement.

Quatre précompilateurs sont offerts par Pro*ORACLE. Ce sont : Pro*FORTRAN, Pro*COBOL, Pro*C, et Pro*PL/I.

L'exécution du programme implique l'exécution des énoncés SQL qui sont contenus dans celui-ci, et c'est à l'utilisateur de traiter les données retournées.

Dans l'écriture d'une application, il existe des énoncés SQL connus par avance et d'autres complètement inconnus (ex. les énoncés d'interrogation). Pro*ORACLE offre, dans chaque précompilateur, les outils nécessaires pour manipuler ces deux possibilités. Dans notre application, nous nous sommes servis du Pro*C, qui dispose de quatre méthodes permettant de travailler avec des énoncés dynamiques.

Note : dans un programme, une section spéciale contient les déclarations de variables propres à ORACLE. Celles-ci peuvent être des variables d'entrée (pour paramétrer les énoncés), ou des variables de sortie (pour stocker les données envoyées par la base de données au programme d'application).

- **méthode 1** : utilisation d' EXECUTE IMMEDIATE

cette méthode précompile un énoncé SQL quelconque, à l'exception d'un énoncé d'interrogation. L'énoncé ne peut pas avoir de variables d'entrées, ni de variables de sortie, et il ne peut être exécuté qu'une seule fois. Exemple :

```
strcpy (enonce, "DELETE FROM PATIENT WHERE PID = 12345");
```

l'énoncé SQL est contenu dans la variable ORACLE *enonce*, son exécution est donnée par :

```
EXEC SQL EXECUTE IMMEDIATE : enonce ;
```

- **méthode 2** : utilisation de PREPARE et EXECUTE

cette méthode exécute un énoncé SQL quelconque, à l'exception d'un énoncé d'interrogation. L'énoncé peut contenir des variables d'entrées peut être réexécuté plusieurs fois.

Exemple :

```
strcpy(enonce, "DELETE FROM PATIENT WHERE PID = : pidpat");  
pidpat est une variable ORACLE qui va recevoir des valeurs dans le programme.
```

L'énoncé est d'abord préparé :

```
EXEC SQL PREPARE S1 FROM : enonce ;
```

La donnée est demandée :

```
scanf("%d", &pidpat) ;
```

une donnée est lue au clavier et stockée dans *pidpat*.

L'énoncé est exécuté tant qu'il reçoit des valeurs différentes de 0 :

```
while (pidpat != 0)  
{ EXEC SQL EXECUTE S1 USING : pidpat ;  
  scanf("%d", &pidpat);  
}
```

l'énoncé est exécuté avec chaque valeur de *pidpat* donnée.

- **méthode 3** : utilisation de **PREPARE**, **DECLARE**, **OPEN**, **FETCH** et **CLOSE**.

cette méthode exécute un énoncé SQL quelconque. L'énoncé peut contenir des variables d'entrée et de sortie. Cette méthode est très similaire à la méthode précédente, à l'exception de la commande **SELECT** qui, cette fois, peut être exécutée. Dans les énoncés d'interrogation, la clause **SELECT** doit être connue à l'avance, alors que la clause **where** peut être variable.

Exemple :

Une donnée est associée à la variable *date* :

```
strcpy (date, "01-01-50");
```

L'énoncé est copié dans la variable *énoncé* :

```
strcpy (enonce, "SELECT NOM, DATE_NAI FROM PATIENT WHERE  
DATE_NAI < : date");
```

```
EXEC SQL PREPARE S1 FROM : enonce ;
```

prépare l'énoncé SQL donné dans la variable *enonce* et lui assigne le nom de *S1*.

```
EXEC SQL DECLARE C1 CURSOR FOR S1 ;
```

associe un curseur (*C1*) à l'énoncé *S1*.

```
EXEC SQL OPEN C1 ;
```

ouvre le curseur *C1*.

```
EXEC SQL WHENEVER NOT FOUND GOTO END ;
```

```
for ( x = 0; ; ++x)  
{ EXEC SQL FETCH C1 INTO : pnam, pdate;  
  printf("%s\t, %s\n", pname, pdate);  
}
```

exécute l'énoncé tant qu'il y a des données à récupérer et passe à l'instruction *END* quand il n'y en a plus. **FETCH** permet de déplacer le curseur et de

récupérer le n-uplet sélectionné. Les données sont stockées dans les variables de sortie *pname* et *pdate*.

END :

EXEC SQL CLOSE C1 ;

ferme le curseur C1.

- **méthode 4** : utilisation de DESCRIBE, PREPARE, OPEN, FETCH et CLOSE.

Cette méthode est utilisée pour l'exécution d'énoncés dynamiques, c'est-à-dire d'énoncés qui ne sont pas connus de l'application au moment de son écriture. Les énoncés de la méthode précédente sont utilisés avec l'énoncé DESCRIBE, qui, dans le cas d'une requête, analyse la clause SELECT et détermine le nombre et le type de constituants participants. De cette manière, le programme peut réserver un espace mémoire pour recevoir les données récupérées par la requête.

Comme dans la méthode précédente, un curseur est associé à l'énoncé pour mémoriser les n-uplets de la zone mémoire réservée.

Dans cette section, nous décrivons les trois modules (CODIF, CLICHES et QUERYS) qui forment actuellement le système SGIR. Ils sont écrits en utilisant le langage C et ils s'exécutent sur VAX 750 sous VMS.

4.5.1. Le module CODIF.

Le module CODIF réalise les insertions de données alphanumériques et de données textuelles concernant les comptes rendus, les patients, les médecins demandeurs, ainsi que les demandes d'examens. Nous supposons que les données concernant chacun de ces objets sont concentrées dans des fichiers séparés. A l'intérieur des fichiers, les enregistrements sont séparés les uns des autres par le symbole " # ". Ces fichiers forment l'entrée du module CODIF. A partir de ces informations, il insère des données dans le fichier de signatures et dans la base. La figure 4.4 illustre ce processus.

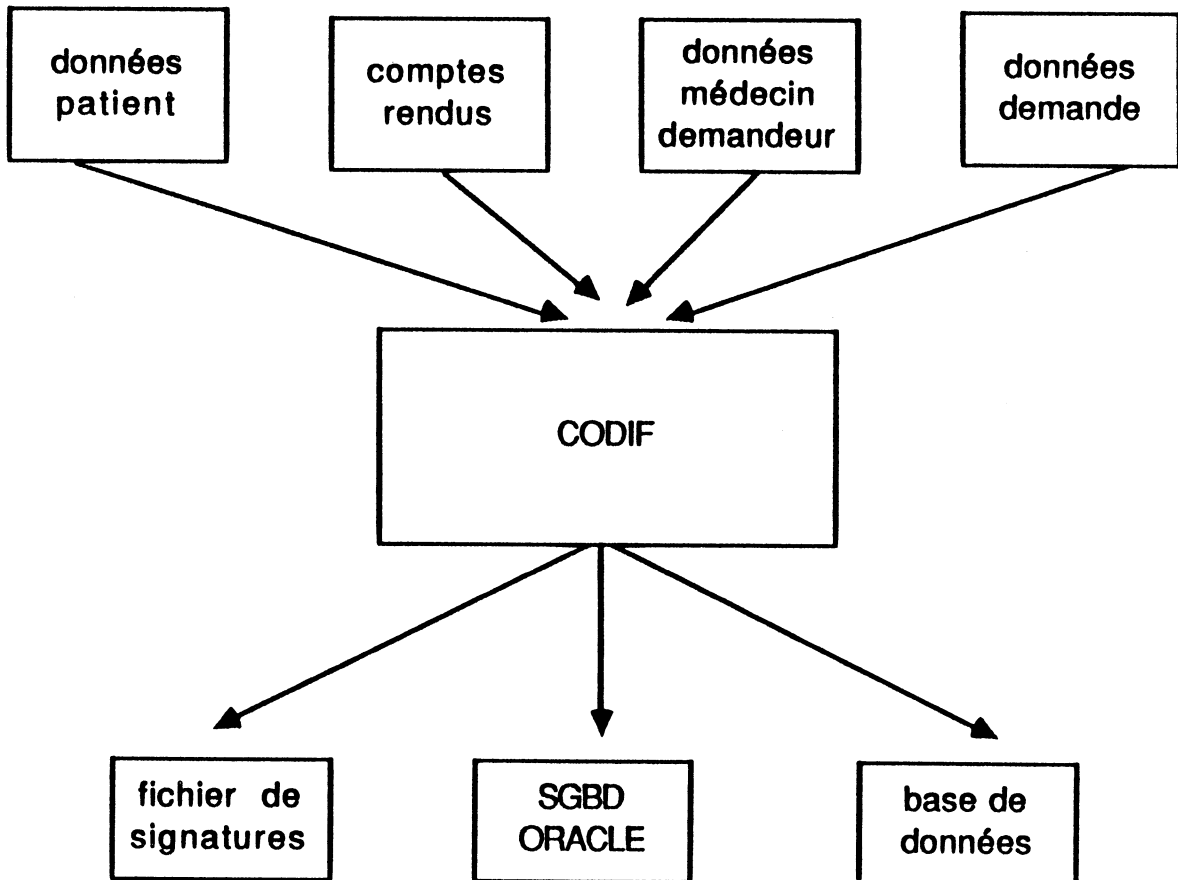


figure 4.4

Nous décrivons dans ce qui suit la structure des différents fichiers de données.

a) Le fichier de comptes rendus.

Les comptes rendus sont saisis séquentiellement dans un fichier standard. Les données doivent être ordonnées, afin de permettre son identification. Ainsi, les deux premières lignes du fichier des comptes rendus doivent respectivement correspondre au numéro d'examen et au numéro de dossier. La troisième ligne de données correspond à la date de rédaction du compte rendu et, dans la ligne suivante, on peut trouver soit l'identification du demandeur (médecin externe ou service de l'hôpital), soit le nom du patient. Par conséquent le demandeur n'est pas une donnée obligatoire. Afin d'identifier le type de la données, il faut que le nom du médecin soit précédé de son titre professionnel (ex. Prof, Dr, etc) et que le nom du patient soit précédé du terme générique qui fait appel à son sexe (ex. Madame, Monsieur). Si ni l'un, ni l'autre de ces deux discriminants est repéré, alors on présume que la donnée concerne le demandeur, et qu'il s'agit d'un service interne à l'hôpital.

On peut trouver ensuite la date de naissance du patient (elle n'est pas une donnée obligatoire) et dans la ligne suivante son identification. Avec cette donnée, finissent les données administratives contenues dans le compte rendu. On passe ensuite au corps du compte rendu. Comme nous l'avons mentionné, il peut être formé d'une introduction, des constatations et d'une conclusion. Afin d'identifier le début et la fin de chaque section, il faudra soit introduire (au commencement de chaque section) son nom, soit mettre un identificateur qui signale la fin de la section. Après le corps, on trouve le nom du radiologue. Cette donnée marque la fin du compte rendu.

CODIF reçoit donc ce fichier comme une de ses entrées et commence sa lecture séquentiellement. Il identifie la nature de chaque donnée et les concentre dans des endroits précis de la mémoire. Au fur et à mesure que les données sont lues, CODIF vérifie qu'elles correspondent aux données attendues. S'il détecte des anomalies graves (ex. le compte rendu n'as pas de numéro d'examen, etc), le compte rendu est refusé, il fait connaître à l'utilisateur les causes, et il continue avec le compte rendu suivant. Quand CODIF arrive aux données textuelles, il fait appel à la procédure de signature et commence à former les blocs logiques. Chaque bloc formé est ensuite signé et concentré dans une zone de la mémoire. La lecture du nom du radiologue indique la fin des données textuelles, et aussi la fin du compte rendu. L'exemple suivant illustre la structure d'un enregistrement de ce fichier.

N_EX : 12000
N_DOSS : 13000

Le 25.04.87

Dr. PARAMELLE

Monsieur DUPONT Pierre
Né le 15/05/43
No. Sec. Soc. 123456

Tomodensitométrie thorax :
(processus expansif médiastino-pulmonaire droit - étiologie non connue au moment où on pratique l'examen).

Les constatations sont les suivantes :

- processus expansif et invasif de grande dimension siégeant au niveau du médiastin,
 - . au niveau de la loge de Barety jusqu'au niveau de la loge pré-carénales. Les structures anatomiques apparaissent refoulées et comprimées, notamment la veine cave supérieure dont la paroi postérieure est envahie.
 - . loge de la bifurcation
 - . médiastin antérieur au devant du tronc veineux brachio-céphalique.
- opacités pulmonaires en continuité.

En conclusion :

Processus expansif et invasif médiastino-pulmonaire avec atteinte pleurale, dans lequel il est difficile de dissocier une tumeur primitive et des adénopathies régionales de grande dimension. L'hypothèse d'un APC à point de départ droit peut être évoquée.

Dr. J. LEGRAND

#

b) Le fichier patient.

Ce fichier contient les données concernant les patients qui vont être insérés dans la base de données. L'enregistrement de comptes rendus ou des images nécessite impérativement l'existence du patient dans la base de données. Un enregistrement de ce fichier doit suivre le format suivant :

nom du patient
identificateur
sexe
date de naissance
adresse personnelle

Seules les deux premières lignes de données sont obligatoires dans chaque enregistrement. L'exemple suivant met en évidence la structure de ce fichier.

DUPONT Pierre
123456
M
12 Decembre 1945 2, Av. Gabriel Perri 38406 St. Martin d'Herès

PIERROT Anne
678910
F
10 août 1960
#

c) Le fichier médecin.

La structure de ce fichier est très similaire au fichier précédent. Il contient les informations concernant les médecins qui font des demandes d'examens au service de radiologie. Un enregistrement doit avoir le format suivant:

nom du médecin
sexe
spécialité
adresse professionnelle
téléphone professionnel

Exemple :

DELACROIX Jean Paul
M
Urologue
11, Place de la Concorde, Paris
1485562122
#

d) Le fichier demande.

Le fichier *demande* contient les informations concernant les demandes d'examen. Les données sont extraites des lettres de demande que reçoit le service de radiologie.

nom du demandeur
date de la demande
nom du patient à examiner
organe à examiner

Ces données sont insérées dans la relation EXAMEN, néanmoins la relation ne contient pas le nom du malade, mais son identificateur. CODIF fait la recherche de l'identificateur du patient dans la base de données. S'il trouve des ambiguïtés pour obtenir l'identificateur (c'est-à-dire, deux patients de même nom), il avertit l'utilisateur et lui demande des informations complémentaires pour lever l'ambiguïté. Nous donnons ici un exemple d'enregistrement :

DELACROIX Jean Paul
15 septembre 1986
DUPONT Pierre
rein gauche

Insertion de données dans la base.

L'insertion des données se fait en utilisant des énoncés d'insertion précompilés (cf. méthode 2 et 3 du Pro*^C). Pour cela, il existe un énoncé d'insertion pour chaque relation mise à jour. L'énoncé est compilé une seule fois, mais exécuté autant de fois que cela est nécessaire.

Les premières données insérées dans la base sont les données administratives concernant les patients. La contrainte d'intégrité sur "l'unicité de la clé" nous permet de contrôler qu'un patient ne sera pas inséré deux fois.

Suivent ensuite, les insertions concernant les médecins demandeurs et les lettres de demande associées, et enfin les insertions de comptes rendus. Avant d'insérer un compte rendu, on commence par vérifier que le patient existe déjà dans la base de données. Si la réponse est positive, alors on insère, dans la relation COMPTE_RENDU, les données administratives du compte rendu en question. Par la suite, les blocs de texte et les identificateurs sont insérés dans la relation TEXTE.

4.5.2. Le module CLICHES.

Ce module est utilisé pour insérer, dans la base de données, les images logiques générées par l'IRMN. Ce module ne peut pas être utilisé pour toutes les techniques d'imagerie, étant donné que chacune possède un format spécifique pour l'en-tête du fichier contenant les images. Nous avons préparé l'insertion pour les images provenant de l'IRMN, mais le processus est similaire pour insérer des images provenant d'autres sources. Dans ce module, nous utilisons les requêtes précompilées pour réaliser les insertions (cf. méthode 2 et méthode 3 du Pro*c).

Les fichiers du IRMN contiennent un bloc de 512 octets qui constitue l'en-tête du fichier. Ici, sont contenues les informations concernant les coupes stockées dans les autres blocs du fichier. Le module CLICHES a pour fonction d'extraire de l'en-tête les informations qui avaient été sélectionnées préalablement pour caractériser l'image dans la base de données. Parmi les informations contenues dans le bloc, on trouve le numéro d'examen et l'identificateur du patient. Comme dans le cas des comptes rendus, avant de réaliser les insertions dans la relation IMAGE_INF et la relation IRMN, le module CLICHES vérifie d'abord l'existence du patient dans la base. L'existence du compte rendu n'est pas obligatoire pour pouvoir faire l'insertion d'une image. Si celui-ci est absent, la base sera quand même dans un état cohérent. Si le compte rendu associé n'existe pas dans la base de données, l'accès à l'image étant limité aux données qui composent l'image logique.

4.5.3. Le module QUERYYS.

Ce module concerne l'interrogation de la base de données. Le module commence par ouvrir la base de données et par réserver un espace dans la mémoire pour recevoir les requêtes de l'utilisateur. La méthode d'énoncés dynamiques est utilisée pour interroger la base (cf. méthode 4 du Pro*C). Le module reçoit comme entrée une requête SQL ou une requête SQLMED. Dans le cas d'une requête

SQLMED, il réalise une lecture séquentielle en faisant au fur et à mesure l'analyse syntaxique. En même temps, il détecte les demandes de l'utilisateur, ses conditions de sélection ainsi que les objets et les constituants participants. Si la requête n'est pas syntaxiquement correcte, un message d'erreur est envoyé à l'écran et le système se place en attente d'une nouvelle requête.

Si la clause WHERE contient un prédicat de recherche par contenu sur les données textuelles, QUERYS envoie ce prédicat à la procédure de signature pour le signer et obtenir l'identificateur de la signature susceptible de contenir l'information cherchée. Ces identificateurs serviront plus tard pour récupérer les données textuelles ou pour réaliser des opérations booléennes avec d'autres éléments de la base de données. L'étape suivante consiste à former la clause FROM de la requête. Dans sa lecture, QUERYS a repéré les constituants participant dans la requête. Ces constituants vont permettre de déterminer quelles sont les relations qui sont concernées ainsi former les trajets d'accès.

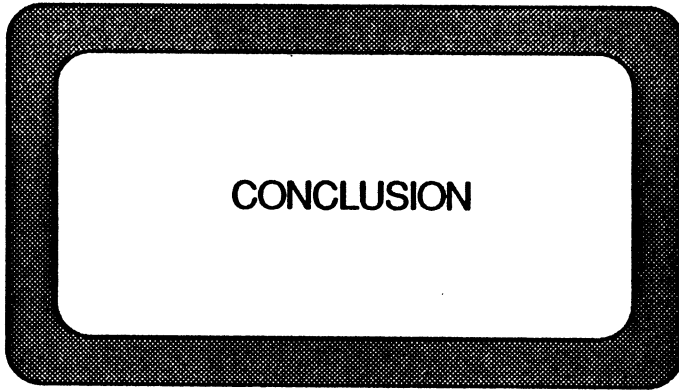
Dans la base de données, il existe une relation qui nous permet de former les chemins d'accès valides reliant une relation à une autre. Chaque n-uplet de cette relation contient les noms des relations qui peuvent être joints sur un constituant donné. Ainsi, une fois que les relations participant à la requête sont mises en évidence, cette relation est consultée pour en extraire les éléments qui vont nous servir à former les jointures correspondantes.

Après cette étape, QUERYS transforme la requête reçue en une requête purement relationnelle. Cette requête est envoyée au SGBD. Si des données alphanumériques sont récupérées, elles sont affichées par QUERYS sous forme de tableau. Si des comptes rendus (complets ou partiels) sont sélectionnés ils sont affichés à l'écran selon les conditions imposées par l'utilisateur. L'utilisateur a la possibilité de garder le compte rendu affiché dans un dossier des comptes rendus sélectionnés. De cette façon, à la fin de la session de travail, il pourra réafficher les comptes rendus qu'il a sélectionnés et éventuellement afficher aussi les éléments associés tels que les patients et/ou les images.

Si une image est demandée QUERYS affiche les données alphanumériques concernant l'image logique. En même temps, il construit un fichier contenant le "path-name" de l'image physique. Si l'utilisateur désire afficher l'image, alors il le fera manuellement en utilisant le contenu de ce fichier.

L'utilisateur sort de QUERYYS avec une commande exit, qui implique la fermeture de la base de données et la libération de la place mémoire retenue.







5. CONCLUSION

Au long de ce travail, nous avons présenté les besoins réels de ce que nous appelons une "application nouvelle". Ainsi, nous avons cherché, d'une part, à décrire les exigences de l'application médicale en question et, d'autre part, à expérimenter un nouveau système de base de données dit généralisé. Même si nous ne sommes pas allés jusqu'à l'implémentation avec le SGBD TIGRE, nous avons pu mettre en évidence les avantages que ce modèle généralisé offre par rapport à un modèle relationnel.

En modélisant l'application, nous avons pu constater que les structures de données offertes par le modèle TIGRE sont mieux adaptées aux besoins réels que celles du modèle relationnel. Les notions de généralisation et de spécialisation nous ont apporté un aspect très important d'abstraction du monde réel. D'un autre côté, la notion de document généralisé, nous a permis de bien représenter la structure naturelle d'un compte rendu. Pour sa part, les associations et les rôles des entités participantes nous ont permis de représenter le comportement des objets du dossier radiologique et les liens existants. En général, nous pouvons conclure que TIGRE offre un grand confort pour intégrer les concepts et pour les attacher des comportements dynamiques.

Néanmoins, en ce qui concerne les opérations sur les images, il est évident que ni TIGRE ni notre système n'offrent directement des fonctions de traitement d'images. TIGRE étant un modèle orienté plutôt bureautique n'a pas considéré ces aspects. Or, comme nous l'avons montré dans l'état de l'art présenté au chapitre 2, plusieurs systèmes se servent d'un système de traitement d'images (avec des résultats très performants) pour effectuer ces opérations. Ainsi, une extension intéressante à ce travail, serait de marier le langage de manipulation de la base de données au langage de manipulation d'un système de traitement d'images.

En se basant sur la philosophie du modèle TIGRE, nous avons réalisé une modélisation de l'application sur un modèle relationnel. Pour offrir un environnement généralisé, nous avons construit un logiciel (qui est opérationnel) nous permettant de manipuler les données multimédia de manière homogène et intégrée. Ce logiciel représente donc une couche multimédia entre l'application médicale et le SGBD relationnel (ORACLE), qui permet l'insertion et l'interrogation des données multimédia.

Nous avons créé un langage d'interrogation (SQLMED) qui exploite tous les avantages du langage SQL et qui offre en plus des facilités pour manipuler les

données multimédia, en acceptant en outre des recherches par contenu sur les données textuelles. SQLMED a pour avantage d'offrir aux usagers une vue des objets très similaire à celle qu'ils ont dans la vie réelle. Cependant, son inconvénient principal est sa spécificité pour l'application médicale en question; spécificité qui explique sa rigidité. Divers travaux qui généralisent cette idée sont menées actuellement et visent à libérer l'utilisateur de la vision multirelationnelle d'une base de données [BOSC 86], [LITW 86]. Ainsi, l'utilisateur va voir une base de données comme un ensemble d'objets composés d'un ensemble d'attributs. De ce fait, la formulation de questions ne relève plus des navigations logiques entre les relations.

En ce qui concerne les recherches par contenu sur les données textuelles, notre approche considère un texte comme un chaîne de caractères ou comme un ensemble de chaînes de caractères. Il est important de noter que cette approche s'avère insuffisante si l'on s'intéresse au contenu du texte plutôt qu'aux textes en tant que tels. Ainsi, il est clair que les recherches par contenu pourraient largement s'enrichir avec l'introduction des aspects sémantiques et syntaxiques, qui ont un rapport plus direct avec le traitement de la langue naturelle. Ceci est valable pour les deux modélisations de l'application que nous avons faites (modèle TIGRE, modèle relationnel).

Ce cette manière, l'intégration des résultats de recherches menées actuellement par un autre groupe du laboratoire, pour traduire les comptes rendus médicaux écrits en un langage quasi naturel dans des concepts médicaux standardisés [CBCD 87], viendrait enrichir l'application.

L'application concernée ne représente qu'un petit module de ce que pourrait être une application médicale ou un système d'information hospitalier. Dans notre travail, nous avons traité principalement deux objets complexes: les images radiologiques et les comptes rendus. Mais, dans une application médicale plus générale, le nombre d'objets complexes à manipuler est plus important. A l'issue de cette expérience, nous avons montré la faisabilité de concevoir un système qui puisse intégrer et manipuler d'une manière homogène les données généralisées. En même temps, nous avons ouvert le chemin pour continuer l'informatisation du service de radiologie en particulier, mais aussi les autres services de l'hôpital.

Nous proposons les extensions suivantes au travail réalisé :

La première porte sur la transformation d'une phrase écrite en langage quasi naturel, en une représentation interne décrivant la réalité médicale de la façon la moins ambiguë et la plus exacte possible. Ceci nous permettrait de "*comprendre*" les

textes médicaux aussi bien pour le stockage, que pour la récupération. Autrement dit, l'information qu'une phrase apporte n'est pas strictement liée à la suite des symboles (caractères) qui la représentent. Elle dépend également du contexte qui constitue un ensemble de connaissances implicites. Ainsi, on dit qu'un système informatique a "*compris*" un texte lorsqu'il est capable de répondre à toutes sortes de questions concernant ce texte. Il est connu, que le traitement automatique de la langue naturelle, pose de grandes difficultés liées à l'extrême variété des formes syntaxiques et, plus encore aux ambiguïtés très nombreuses qui ne sont souvent levées que par référence au contexte. Néanmoins, des résultats très satisfaisants peuvent être atteints dans des domaines plus restreints, comme ce serait le cas pour notre application sur la Tomodensitométrie du cancer des bronches.

Quand nous parlons de "*compréhension des textes*" par des programmes informatiques, nous entrons dans le domaine de l'intelligence artificielle, laquelle vise à faire traiter par l'ordinateur des problèmes usuellement résolus par l'homme. Un autre aspect de l'intelligence artificielle qu'il serait intéressant d'exploiter est le concept *de base de connaissance ou de base de faits*.

Dans un dossier médical, on peut trouver deux types d'informations : les informations reflétant les connaissances médicales et les informations reflétant les renseignements recueillis sur les malades et qui forment les *faits*. Ainsi, une base de connaissances et des faits pourrait être construite à partir des données du dossier radiologique. En effet, les cas qui sont représentatifs d'une certaine pathologie, pourraient (au lieu de former un objet indépendant du dossier comme c'est le cas dans notre système), alimenter une base de connaissance, base qui serait utilisée dans un système d'aide au diagnostic ou dans l'enseignement assisté par ordinateur.

Finalement et, comme nous l'avons mentionné, l'application médicale, ne représente qu'un élément de toute une structure hospitalière importante. Il serait intéressant d'avancer dans cette structure et d'élargir le champ d'application à d'autres services, pour former ainsi un système d'information hospitalière. Le milieu hospitalier est une source très riche et complexe d'informations, ainsi, plusieurs recherches concernant l'organisation et la manipulation de l'information trouveront des applications pour mettre en œuvre leurs théories. Par exemple, les modèles de données généralisés, les recherches sur la représentation du facteur temps, les contraintes d'intégrité, la cohérence de l'information, la manipulation de valeurs floues ou incertaines, trouveront dans l'information médicale, un milieu idéal pour l'évaluation de ces résultats.





BIBLIOGRAPHIE



6. BIBLIOGRAPHIE

- [ABSG 86] ABITEBOUL. S, SCHOLL. M, GARDARIN. G, SIMON.
Towards DBMSs for supporting new applications.
proc. VLDB 1986. Kyoto Japon. Août 1986.
- [ADIB 85] ADIBA. M.
Les bases de données généralisées.
RIO 1985.
- [ADIB 86] ADIBA. M.
Modeling complex objects for multimedia databases.
Congrès Entity-Relationship. Dijon, France. Novembre 1986.
- [ADBU 86] ADIBA. M, BUI QUANG N.
Historical multi-media databases.
Proc. VLDB 1986. Kyoto, Japon. Août 1986.
- [AREN 79] ARENSON. R.L., LONDON. J.W.
Comprehensive analysis of a radiology operations management
computer system.
Radiology, Vol. 133. November 1979
- [ASSM 83] ASSMAN. K, HOHNE. K.H.
Investigation of structures and operations for medical image
data bases.
Picture Archiving & Communication Systems for Medical
Applications. Vol 418. 1983
- [ASSM 86] ASSMAN . K, VENEMA. R, HOHNE. K.H.
Software tools for the development of pictorial information
systems in medicine. The ISQL experience.
Pictorial Information Systems in medicine.
Springer-Verlag NATO ASI series. Heidelberg, 1986.
- [ASSM 84] ASSMAN. K, VENEMA, R, RIEMER. M, HOHNE. K.H.
The ISQL-Langage. A uniform tool for managing images and
non-image data in an image data base management system.
Proc. ISMIII 1984
- [BAKK 86] BAKKER. A.R.
Integrated hospital information system.
Pictorial Information Systems in medicine. Springer-Verlag
NATO ASI Series. Heidelberg, 1986.
- [BCNM 87] BERRUT. C, CINQUIN, P, NIE. Y, MUNOZ. G,

- CHIARAMELLA. Y, DEMONGEOT. J, COULOMB. M.
Modélisation sémantique de comptes rendus radiologiques.
Soumis au Congrès sur l'intelligence artificielle et santé.
Toulouse, Octobre 1987.
- [BECI 87] BERRUT. C, CINQUIN, P.
An information retrieval system for medical reports: outline
of the indexing process.
Congrès AIME 87 (Artificial Intelligence in MEdecine).
Marseille, Septembre 1987.
- [BENC 84] BENCHIMOL. C.D.
Introduction à l'imagerie médicale.
Premier colloque image (traitement, synthèse, technologie et
applications). Biarritz. Mai 1984.
- [BERT 86] BERTRAND. J.C, HUSSON. P, PERARNAUD. J,
GAYET. J.
La filière SYMPHONIE: C.H.R. Tours et Metz.
EUROMED 86. Montpellier, Novembre 1986.
- [BIZA 86] BIZAIS, Y.
Système DIMI: système de gestion et de traitement d'images
dans un département d'imagerie médicale.
Informatique médicale, Imagerie, Systèmes d'information
Hospitaliers
Sauramps Médical. Montpellier, 1986
- [BOHM 79] BOHM. M, REMIER. M, HOHNE. K.H.
Implementation of relational data base system on top of a
commercial DBMS.
Lecture Notes in Medical Informatics, 5.
Springer-Verlag. pp.611-618. 1979.
- [BOSC 86] BOSC. P, CAUVET.C, FLORY. A, GRANGER. V,
LYNGAT. J.Y, ROLLAND. C.
La relation universelle : diverses approches étudiées
à travers un exemple.
Nouvelles perspectives des bases de données.
Editions EYROLLES, 1986.
- [BRAM 85] BRAMI. P.
Un analyse morphosémantique des mots composés du langage
médical.

U.S.M.G. Thèse de Docteur en Médecine, Diplôme d'Etat.
Grenoble, Décembre 1985.

[BROL 72] BROLIN. I.

MEDELA: An electronic data-processing system for radiological reporting.

Radiology, Vol. 103. Mai 1972.

[BROD 81] BRODIE. M.

On modeling behavioural semantics of data.

Proc. Conf. VLDB, Cannes, France, 1981.

[BUI 86] BUI QUANG N.

Aspects dynamiques et gestion du temps dans les systèmes de bases de données généralisées.

Thèse de Docteur. INPG, Grenoble 1986.

[CBCD 87] CINQUIN. P, BERRUT. C, CHIARAMELLA. Y,
DEMONGEOT. J, COULOMB. M.

Interest of quasi natural reporting for medical images retrieving.

A paraître. Proceedings 7th. International Congress on Medical Informatics in Europe. Rome, 21-25 Septembre 1987.

[CDCG 86] CINQUIN. P, DEMONGEOT. J, CHABRE-PECCOUD. M,

Première étape de la conception d'un système d'information hospitalier intégré: Application d'une méthode de conception de système d'information à la spécification d'un système d'information en hématologie

EUROMED 86. Montpellier, Novembre 1986.

[CHAN 77] CHANG. S.K, DONATO. N, McCORMICK. B.H,
REUSS. J, ROCCHETTI. R.

A relational database system for pictures.

Workshop on picture data description and management.

Proc. IEEE 1977.

[CHAN 79] CHANG. S.K, LIN B.S.

Picture algebra for interface with pictorial database systems

Workshop on picture data description and management.

Proc. IEEE 1979.

[CHAN 80] CHANG. S.K, LIN B.S, WALSER. R.

A generalized zooming technique for pictorial database system.

Pictorial Information Systems. Springer-Verlag.

Lecture in Computer Science. 1980.

- [CHAP 80] CHAPERON. J, TREHONY.A, CHAUVET. G, SIX. P
Problèmes posés par la conception d'un système d'information
médicale et épidémiologique dans un Centre Hospitalier
Régional et Universitaire Français.
Pergamon Press. Montréal , 1980
- [CHFA 84] CHRISTODOULAKIS. S, FALOUTSOS. CH.
Design consideration for a message file server.
IEEE Transactions on Software Engineering. Vol. SE-10, No. 2
Mars 1984.
- [CHNG 81] CHANG. N.S, FU. K.S.
Picture query language for pictorial data-base systems.
IEEE, Vol. 0018-9162. Novembre 1981
- [CHNG 80] CHANG. N.S, FU. K.S.
Query-by-Pictorial-Exemple.
IEEE Transactions on Software Engineering. Vol SE-6, No. 6.
Novembre 1980
- [CHRI 84] CHRISTODOULAKIS. S.
Framework for the development of an experimental
mixed-mode message system.
ACM-BCS Symposium on Research and Development in
Information Retrieval. Cambrige, Angleterre, 1984.
- [CHRI 86a] CHRISTODOULAKIS. S, THEODORIDOU. M,
HO. F, PAPA. M, PATHRIA. A.
Multimedia document presentation, information extraction, and
document formation in MINOS : a model and a system.
ACM Transactions on Office Information Systems. Vol. 4,
No. 4. Octobre 1986.
- [CHRI 86b] CHRISTODOULAKIS. S, HO. F, THEODORIDOU. M.
The multimedia object presentation manager of MINOS : a
symmetric approach.
Proc. ACM SIGMOD. Mai 1986.
- [CHRI 86c] CHRISTODOULAKIS. S, FALOUTSOS. C.
Design and performance considerations for an optical disk based
multimedia object server.
A paraître.
- [CHU 87] CHU Grenoble
Rapport d'observation du service de radiologie du C.H.R.U. de

Grenoble.

Rapport interne Centre Régional d'Informatique Hospitalière de
Grenoble. Grenoble. 22 Janvier 1987

[CIMC 87] CINQUIN. P, MUNOZ. G, CHIARAMELLA, Y.

Accès à une base de données radiologique par des comptes
rendus écrits en langue casi naturelle.

Congrès sur la recherche en imagerie médicale.

Paris 18-20 Mars 1987.

[CINQ 84] CINQUIN. P.

Réflexions préliminaires sur le développement d'un système
d'information hospitalier intégré.

Thèse de Docteur en Médecine. Grenoble, Octobre 1984.

[CINQ 87] CINQUIN. P, YAN. J, MUNOZ. G, BERRUT. C,
CHIARAMELLA. Y, DEMONGEOT. J, COULOMB. M.

Traitement automatique de comptes rendus radiologiques
écrits en langage quasi naturel.

Congrès COGNITIVA 87. Paris, France. Mai 1987.

[CINQ 88] CINQUIN. P, MUNOZ. G, DEMONGEOT. J.

Bases de données d'images en radio. Entretiens de Lyon sur
l'imagerie médicale et l'intelligence artificielle.

10-12 Mars 1988.

[CIRD 84] CINQUIN. P, RIALLE. V, DEMONGEOT. J.

Propositions pour une structure d'imagerie médicale intégrée.

Document interne. Département de Mathématiques, Statistiques
et Informatique Médicale.

Faculté de Médecine de Grenoble. Grenoble, Novembre 1984

[CODD 70] CODD. E.F.

A relational model of data for large shared data banks.

Communications of the ACM, Vol. 13, No. 6 Juin 1970.

[COPPERN] COPERNIQUE

Serveur multimédia.

Notice COPERNIQUE. SERVEUR 32, DORSAL 32.

[DACH 82] DACHELET. R, NORMIER. B.

Etude de la langue des comptes rendus d'hospitalisation en vue
d'une analyse automatique.

Etude de faisabilité de la société ERLI. Octobre 1982.

[DATE 83] DATE. C.J.

- An introduction to database systems. Volume II.
Addison-Wesley systems programming series. 1983.
- [DEAD 82] DELOBEL. C, ADIBA. M.
Bases de données et systèmes relationnels.
DUNOD informatique. Paris , 1982.
- [DEKE 86] DEKEYSER. L.J, MARTENS. B, WILLEMS. J.L.
Write-Once-Read-Multiple times storage and medical record
applications.
MEDINFO 86. North-Holland, 1986.
- [DEMO 86] DEMONGEOT. J, CINQUIN. P.
Prespectives de l'informatique médicale.
8èmes Journées francophones sur l'informatique.
Grenoble, 14-15 Janvier 1986.
- [DIAS 86] DIAS. S.C.
Data base design and retrieval methodology in clinical data
search.
MEDINFO 86. North-Holland, 1986.
- [DUBI 77] DUBIEN, R.J. et al.
A database system implementation providing data independence
for medical applications
MEDINFO 77. Amsterdam, 1977
- [DUER 82] DUERINCKX. A.J.
Digital picture archiving and communication systems (PACS) in
diagnostic radiology: a review.
IEEE Vol. CH1808-4. 1982.
- [DUJO 86] DUJOLS. P, AUBAS. P, GREMY. F.
Confidentialité et accessibilité des données médicales dans un
système d'information hospitalier. L'exemple du système
Pasteur au C.H.R. de Montpellier.
Actes du Congrès Informatique Médicale, Imagerie et Systèmes
d'Informatique Hospitaliers.
EUROMED 86. Montpellier, Novembre 1986.
- [DUQU 69] DUQUESNEL. P.
Langage normalisé pour la description des informations
médicales en radiologie.
Thèse de Docteur de Troisième Cycle. Université de Grenoble.
Grenoble, Juin 1969.

- [ECONO 83] ECONOMOPOULOS. P, LOCHOVSKY. F.H.
A system for managing image data.
R.E.A. Mason. Information Processing. 1983
- [FALU 84] FALOUTSOS, C, CHRISTODOULAKIS. S.
Signature files: an acces method for documents and its
analytical performance evaluation.
ACM Transactions on Office Information Systems.
Vol. 2, No 4, Octobre 1984.
- [FALU 85] FALOUTSOS, C.
Design of a signature file method that accounts for
non-uniform occurrence and query frequencies.
Proc. VLDB 85. Stockolme Suede. 1985.
- [FIES] FIESCHI.M, GOUVERNET. J, GIUSIANO. B.
Outlook in computers in health services.
Rapport interne. Lab. d'informatique médicale, Faculté de
Médecine. Marseille.
- [FOX 78] FOX. M.
ACIS - A clinical Information system.
Lectures Notes in Medical Informatics 1.
Springer-Verlag. Berlin, 1978.
- [GARD 85] GARDARIN. G, VALDURIEZ. P.
Bases de données relationnelles. Analyse et comparaison
des systèmes.
EYROLLES. 1985.
- [GELL 76] GELL. G, OSER. W, SCHWARZ. G.
Experience with the AURA free-text documentation system.
Radiology, Vol. 119. Avril 1976.
- [GRIE 78] GRIESSER. G, JAINZ. M.
Computerized support of activities and identification of
related data.
Lectures Notes in Medical Informatics, 1.
Springer-Verlag. Berlin, 1978.
- [GUST 71] GUSTAFSON. R.A.
Elements of the randomized combinatorial file structure.
Proc. ACM SIGIR Symposium on Information Storage and
Retrieval. ACM, New York, 1971.
- [GWIL 86] GWILYM. S, LODWICK. M.D.

Pictorial information systems and radiology improving the quality of communications.

Pictorial Information Systems in medicine. Springer-Verlag
NATO ASI Series. Heidelberg, 1986.

[HACH 86] HACHIMURA. K.

Image query system in clinical environment.

MEDINFO 86. North-Holland, 1986.

[HASK 82] HASKIN. L, LORIE. R.

On extending the fonctions of a relational database system.

ACM SIGMOD Conf. Int. Orlando, Florida. Juin 1982.

[HEAU 81] HEAULME de. M

Conditions que doit remplir un système d'information
hospitalier pour répondre aux projets d'analyse d'activité.

Lectures notes in medical informatics. MIE 81. Toulouse, 1981.

[HEAU 80] HEAULME de. M.

REMEDE: objectifs - conception - fonctions.

CHU Pitié-Salpêtrière. Septembre 1980.

[HEAU 79] HEAULME de. M, TAINURIER. C, THOMAS. D.

Traitement par ordinateur des comptes rendus médicaux:
l'exemple du système REMEDE.

La nouvelle Presse Médicale, Vol. 8, No. 40.

Paris, 22 Octobre 1979.

[HOLL 79] HOLLAAR. L.A.

Specialized merge processor networks for combining sorted lists.

ACM Trans. Database System,

Vol. 3, No. 3. Septembre 1979.

[HUET 86] HUET. B

Revue en méthodologie de conception des systèmes
d'information hospitaliers.

Actes du Congrès Informatique Médicale, Imagerie et Systèmes
d'Information Hospitaliers.

EUROMED 86. Novembre, 1986.

[IMAM 86] IMAMURA. K, IMANISHI. Y, ISHIKAWA. T, et al.

Survey of the access to stored X-ray films in an university
hospital.

MEDINFO 86. North-Holland, 1986.

[JAME 86] JAMES. A. E, CARROLL. F, PICKENS. D. R, CHAPMAN. J.

- ROBINSON. R.R, PENDERGRASS. H.P, ZANER. R.
Medical image management.
Radiology, Vol. 160. 1986.
- [JERU 86] JERUSSALMY. M, GOMES. J.E.A.
A general use real time information system using the
relational model, and its hospital applications.
MEDINFO 86. North-holland, 1986.
- [JOST 82] JOST. R.G, RODEWALD. S.S, HILL. R.L, EVENS. R.G.
A computer system to monitor radiology activity: a
management tool to improve patient care.
Radiology, Vol 145. Novembre 1982.
- [JOST 80] JOST. R.G, TRACHTMAN. J, HILL. R.L, SMITH. B.A,
EVENS. R.G.
A computer for transcribing radiology reports.
Radiology, Vol 136. Juillet 1980.
- [JUNE 86] JUNET. M, FALQUET. G, LEONARD. M.
ECRINS/86 : An extended Entity-Relationship data base
management system and its semantic query language.
Proceedings VLDB 86. Kyoto, Japon. Août 1986.
- [KANA 80] KANAMORI. Y, MASUNAGA. Y, KIDO. K,
NOGUCHI. S.
Design of a database system for skull line drawings
processing orthodontics based on the relational model.
MEDINFO 86. North-Holland, 1980.
- [KLON 80] KLONK. J, SAUTER. K.
Steps towards a methodology for data base design.
MEDINFO 80. North-Holland 1980.
- [LACR 71] LACRONIQUE. F.
Automatization of radiological records.
Methods of Information in Medicine. Vol 10, 1971.
- [LAMA 86] LAMARQUE. J.L, PUJOL. J, ROUANET. J.P, BRUEL.
La nouvelle imagerie médicale: mythe promotionnel ou réalité
médico-sociale?
Actes du Congrès Informatique Médicale, Imagerie et Systèmes
d'Information Hospitaliers.
EUROMED 86. Montpellier, Novembre 1986.
- [LARS 83] LARSON, P-A.

- A method for speeding up text retrieval.
Proc. ACM SIGMOD. Mai 1983.
- [LAVA 72] LAVAL-JEANTET. M, LACRONIQUE. F.
Une approche sémantique du compte rendu radiologique.
Société Française de Radiologie Médicale. Journées Nationales.
Octobre 1972.
- [LEEM 79] LEEMING. B.W, SIMON. M, JACKSON. J.D,
HOROWITZ. G.L, BLEICH. H.L.
Advances in radiologic reporting with computerized language
information processing (CLIP)
Radiology, Vol. 133. Novembre 1979.
- [LEVE 81] LEVEN. F.J.
A methodology for data base software selection in health care
systems.
MIE 81. Lectures Notes in Medical Informatics 11.
Springer-Verlag. Toulouse, 1981.
- [LITW 86] LITWIN. W, ABDELLATIF. A.
Jointures implicites dans le système MRDSM.
Nouvelles perspectives des bases de données.
Editions EYROLLES, 1986.
- [LOPE 86] LOPEZ. M.
Modèles de données, taxonomie et concepts de haut niveau.
Convention Informatique, 1986.
- [LOPV 83] LOPEZ. M, PALAZZO. J, VELEZ. F.
The TIGRE data model.
IMAG et Centre de Recherche BULL.
Rapport de recherche TIGRE. Grenoble, Novembre 1983.
- [MATR 86] MATRA
Base de données multimedia.
Etude bibliographique préliminaire à l'étude theorique.
Val-de-Reuil, 6 Juin 1986.
- [MUNO 86] MUNOZ. G
Etude préliminaire pour le stockage et l'exploitation de
comptes rendus médicaux.
Laboratoire de Génie Informatique.
IMAG Rapport de recherche. Grenoble, Avril 1986.
- [MUNO 86] MUNOZ. G

Stockage et exploitation de dossiers médicaux de radiologie
au moyen d'une base de données relationnelle.

Informatique Médicale, Imagerie, Systèmes d'information
Hospitaliers. Sauramps Médical. Montpellier, 1986.

[MUNZ 86] MUNZ.J, DVORAK.M, GOTFRYD.O, VOLEJNICEK,O.
Patient oriented data base structure and data processing for
departmental IS.

MEDINFO 86. North-Holland, 1986.

[OHSWA 78] OHSAWA, I et al.

Shared hospital information system (SHIS).

Proc. of MEDIS 78. Osaka, Japon, 1978.

[ORAC 84] ORACLE CORPORATION.

ORACLE manuel de référence, guide d'utilisation.

Mars 1984.

[PIET 86] PIETRI. P, BERTRAND. Ph, PERARNAUD. P,
HUSSON. J.C, GAYET. J.

Le système d'accès transactionnel de la base de de données
répartie de la filière hospitalière SYMPHONIE.

EUROMED 86. Montpellier, Novembre 1986.

[POOL 77] POOLE. F, LONGSTAFF. J, ANDERSON. J.

The value of the medical data from a relational data-base.

Proc. Int. Symposium on medical computing.

Toulouse, Mars 1977.

[RABI 84] RABITTI. F, ZIZAKA. J.

Evaluation of access method to text documents in office systems

ACM-BCS Symposium on Research and Development in

Information Retrieval. Cambridge Anglaterrre, 1984.

[ROGA 85] ROGALA. J.P, ASFAR. L.

Définition et réalisation d'une base de données d'images de
télédétection.

Centre Scientifique IBM France. Rapport. Paris, Mai 1985.

[SATO 86] SATOMURA. Y, MASAYUKI. H.

Generalized information system for medical data analysis

CUPIDS.

MEDINFO 86. North-Holland, 1986.

[SAUT 80] SAUTER. K

Databases in medical information systems.

- MEDINFO 80. North-Holland, 1980.
- [SCAR 83] SCARABIN. J.M, GIBAUD. B, CERTAINES. J,
COATRIEUX. J.L.
Serveur d'imagerie médicale accessible via un réseau numérique
expérimental. Projet SIRENE.
Groupe de recherche "signaux et images en médecine".
Université de Rennes 1. Juin 1983.
- [SELT 77] SELTZER. R.A, REIMER. G.W, COOPERMAN. L.R,
ROSSITER. S.B.
Computerized radiographic reporting in a community hospital:
a consumer's report
Am. J. Roentgenol, Vol. 128. Mai 1977.
- [SMIT 84] SMITH. D.F, ROWBERG. A.A.
Data-base management system on a CT scanner computer:
application to teaching file and procedure records.
Am. J. Roentgenol, Vol. 142. Juin 1984.
- [STON 86] STONEBRAKER. M, ROWE. L.A.
The design of POSTGRES.
ACM Vol 191, No 1. 1986.
- [TAKA 80] TAKAO. Y, ITOH. S, HISAKA. J.
An image-oriented database system.
Springer-Verlag. 1980.
- [TAMU 80] TAMURA. H.
Image database management for pattern information processing
studies.
Pictorial Information Systems. Springer-Verlag.
Lecture Notes in Computer Science. 1980.
- [TANG 81] TANG. G.Y.
A management system for an integrated database of pictures and
alphanumerical data.
Computer Graphics and Image Processing, Vol. 16. 1981.
- [TANT 86] TAINURIER. C, HEAULME. M, MERY. C.
Aide à l'analyse d'activité d'un service de cardiologie à
partir du traitement par ordinateur des résumés
d'hospitalisation.
MIE 81. Lecture Notes in Medical Informatics 11,
Springer-Verlag. Toulouse, 1981.

- [TEMP 84] TEMPLETON. A.W, DWYER. S.J, JHONSON. J.A, et al.
An on-line digital image management system.
Radiology, Vol 152, 1984.
- [TEMP 82] TEMPLETON.A.W, DWYER.S.J, ROSENTHAL.S.J, et al.
A peripheralized digital image management system: prospectus.
Am. J. Rœntgenol, Vol. 139, Novembre 1982.
- [TSCH 83] TSICHRITZIS. D, CHRISTODOULAKIS. S,
ECONOMOPOULOS. C, FALOUTSOS. CH.
A multimedia office filing system.
Proc. VLDB 83. Florence Italy, 1983.
- [TSIC 83] TSICHRITZIS. D, CHRISTODOULAKIS. S.
Message files.
ACM Transactions on Office Information Systems.
Vol. 1, No 1, Janvier 1983.
- [TORI 80] TORIWAKE. J, HASEGAWA. J, FUKUMURA. T,
TAKAGI. Y.
Pictorial information retrieval of chest X-ray image data base
using pattern recognition techniques.
MEDINFO 80. North-Holland, 1980.
- [VELE 84] VELEZ. F.
Un modèle et un langage pour les bases de données généralisées
INPG Grenoble. Thèse de Docteur Ingénieur.
Grenoble, Septembre 1984.
- [VRIE 86] VRIES. J.K
An expert system for indexing and retrieving medical
information.
Université de Pittsburgh, Ecole de Médecine, et univeristé
Carnegie-Mellon. Rapport interne. Aout 1986.
- [VUON 84] VUONG. T.D.H.
Contribution à l'étude de l'informatisation du dossier
d'imagerie médicale.
Thèse de Docteur en Médecine, Diplôme d'Etat.
Université de Rennes, Octobre 1984.
- [WHEE 76] WHEELER. P.S, SIMBORG. D.W, GITLIN. J.N.
The Johns Hopkins radiology reporting system.
Radiology, Vol. 119. Mai 1976.
- [WIED 81] WIEDERHOLD. G.

Database for health care. Lectures Notes in Medical Informatics
Springer-Verlag. Berlin, 1981.

[YAMA 80] YAMAGUCHI. K, OHBO. N, KUNII. T.L,
KITAGAWA. H.

ELF: Extended relational model for large, flexible picture
databases.

IEEE, Vol. CH1530-5. 1980.

[YAN 86] YAN. J. .

Interprétation sémantique des comptes rendus médicaux.

Rapport de DEA, Université de Grenoble. Septembre 1986.

[YOKO 82] YOKOYA. N, TAMURA. H.

A database system of microscopic cell images.

IEEE, Vol. CH1804-4. 1982.

[ZOB 80] ZOBRIST. A.L, BRAYANT. N.A.

Designing an image based information system.

Pictorial Information Systems. Springer-Verlag.

Lecture in Computer Science. 1980.

ANNEXES



7. ANNEXE 1.

Définition du schéma conceptuel.

Définition des types construits.

Type Sexe : (f, m) ;

```
Type adresse : record
    numéro : integer ;
    rue : string (20) ;
    ville : string (20) ;
end;
```

```
Type CLICHES : document
defpres TSPI ;
    structure
        CLICHES : begin
            suite_coupes = list (1..*) of cliches_struct = begin
                no_cliché : text ;
                heure_commencement : date ;
                heure_achèvement : date ;
                pixels : image ;
            end ;
        end;
    constantes
        fonction temps_acquisition (heure_commencement, heure_achève-
ment) ;
        end;
end.
```

```
Type SECTION : document
defpres TSIP ;
  structure
    SECTION : begin
      signature : image ;
      corps = list (1..3) of
        paragraphe = begin
          titre : text ;
          contenu : list (1, *) of text ;
          end ;
        end ;
    end ;
end.
```

Définition de types de classe.

```
Type COMPTE_RENDU : entity
  nom_demandeur : string(35) ;
  nom_radiologue : string(35) ;
  signature : image ;
  date_cr : date ;
  texte : SECTION ;
end ;
```

```
Type PATIENT : entity
  key pid : integer end_key ;
  nom : string (35) ;
  sexe : Sexe ;
  date_nai : date ;
  adresse : Adresse ;
End ;
```

Type MEDECIN : entity

key nom : string (35) end_key ;
spécialité : string (20) ;
adresse : Adresse ;
téléphone : string (12) ;

End ;

Type IMAGES : entity

Key n_ex : integer end_key ;
source : (scanner, IRMN) ;
organe : string (20) ;
date_ex : date ;
nombre_coupes : integer ;
epaisseur_coupes : real ;
coupes : CLICHES ;

End ;

Type IMAGE_INEDP : entity

key iid : integer end_key ;
fonction_applique : string (15) ;
source : (scanner, IRMN) ;
organe : string (15) ;
intérêt : SECTION ;
coupes : CLICHES ;

End ;

Type IMAGE_REF : union of SCANNER and IRMN

End;

Type **SCANNER** : Specialisation of **IMAGES**

Where source : 'scanner' ;

manipulateur : string (20) ;

nombre_coupes_ss_inj : integer ;

nombre_coupes_ac_inj : integer ;

End ;

Type **IRMN** : Specialisation of **IMAGES**

Where source : 'IRMN' ;

type_collection : string (15) ;

indicateur_sequence : string (15) ;

type_antenne : string (8) ;

resolution : string (5) ;

mode_acquisition : string (13) ;

orientation_plans_acquisition : string (13) ;

mode_positionnement_coupes : string (13) ;

End ;

Type **DEMANDEUR** : Specialisation of **MEDECIN**

Where spécialité <> 'radiologue' ;

End ;

Type **RADIOLOGUE** : Specialisation of **MEDECIN**

Where spécialité : 'radiologue' ;

End ;

Type **EXAMEN** : Relationship between **PATIENT** : examiné (1,*) and **IMAGES** : produites (1,1) and **COMPTE_RENDU** : élaboré (1,1) ;

date_dem : date ;

nom_demandeur : string (25) ;

organe_ex : string (20) ;

End ;

Type OBSERVATION : Relationship between RADIOLOGUE : observe (1,*)
and IMAGES : est_observée (1,1) ;

End ;

Type DERIVATION : Relationship between IMAGES : source (0,*) and
IMAGE_DERIVEES : dérivées (0,*) ;

End ;

Type EN_TRAITEMENT : Relationship between PATIENT : est_traité_par
(1,1) and DEMANDEUR : traité (1,*) ;

End ;

Type DOSSIER : Relationship_aggregation of EXAMEN
no_dossier : integer ;

End ;

LISTE DES RAPPORTS TIGRE

- TIGRE 1 Présentation générale du projet TIGRE
Janvier 1983
- TIGRE 2 The Tigre data model
M. Lopez, J. Palazzo-Oliveira, F. Velez - Novembre 1983
- TIGRE 3 Proposition de modèle pour la normalisation des documents
G. Bogo, H. Richy, I. Vatton - Mars 1983
- TIGRE 4 Recommandations pour l'ergonomie d'un poste de travail attaché à un système bureautique
I. Forestier - Mars 1983
- TIGRE 5 La correspondance de schémas entre les modèles TIGRE et relationnel
J. Palazzo-Oliveira, F. Velez - Novembre 1983
- TIGRE 6 Description des éléments du modèle de document
G. Bogo, H. Richy, I. Vatton - Septembre 1983
- TIGRE 7 Programmation en logique pour une base de données généralisées
Nguyen G.T., J. Olivarès, P. Winninger - Novembre 1983
- TIGRE 8 Machine base de données - Etat de l'art
M. Burnier - Novembre 1983
- TIGRE 9 Caractérisation des formulaires pour une base de données généralisées
C. Collet - Novembre 1983
- TIGRE 10 Accès concurrent aux documents
S. Baltas - Janvier 1984
- TIGRE 11 Définition formelle du langage LAMBDA
F. Velez - Mars 1984
- TIGRE 12 Logic programming for a generalized data management system
M. Adiba, Nguyen G.T. - Janvier 1984
- TIGRE 13 Modèle et fonctionnalités pour un système intégrant outils BD et outils de conception
D. Rialhe - Janvier 1984
- TIGRE 14 SAGE : Un Système d'Autorisation GÉnéralisé
M. Adiba, F. Azrou - Janvier 1984
- TIGRE 15 Coopération de Prolog et d'un SGBD généralisé : Principes et applications
Nguyen G.T., J. Olivarès, P. Winninger - Avril 1984

- TIGRE 16 Two-dimensional editing
V. Joloboff, V. Quint - Juin 1984
- TIGRE 17 Aspects logiciels de la communication homme-machine sur les postes de travail individuels
V. Joloboff - Juin 1984
- TIGRE 18 Information processing for CAD/VLSI on a generalized data management system
M. Adiba, Nguyen G.T. - Juin 1984
- TIGRE 19 Représentation de la sémantique et des méta-connaissances dans une Base de Données Généralisées
Nguyen G.T., J. Olivarès - Juillet 1984
- TIGRE 20 Description de l'éditeur Tigre - 1) Fonctionnalités, architecture, interfaces
H. Richy, I. Vatton - Juillet 1984
- TIGRE 21 Gestion des paramètres d'un document Tigre
C. Collet - Novembre 1984
- TIGRE 22 Les formulaires électroniques dans Tigre
C. Collet - Janvier 1985
- TIGRE 23 Notion de temps dans les bases de données généralisées
M. Adiba, Bui Q.N., J. Palazzo-Oliveira - Décembre 1984
- TIGRE 24 L'intégrité sémantique dans une base de données généralisées
Nguyen G.T., J. Olivarès - Novembre 1984
- TIGRE 25 Prolog et bases de données relationnelles
Nguyen G.T., P. Winninger - Décembre 1984
- TIGRE 26 SYCSLOG : Système logique d'intégrité sémantique
Nguyen G.T., J. Olivarès - Janvier 1985
- TIGRE 27 Grif : un éditeur interactif de documents structurés
V. Quint, I. Vatton - Mars 1985
- TIGRE 28 Semantic data organization on a generalized data management system
Nguyen G.T., J. Olivarès - Mai 1985
- TIGRE 29 Gestion des historiques pour les bases de données généralisées TIGRE
Bui Quang Ngoc - Juin 1985
- TIGRE 30 Un mécanisme transactionnel pour le SEBD TIGRE
E. Pedraza - Novembre 1985
- TIGRE 31 La prise en compte de documents structurés dans un SEBG aspects modèle, langage et architecture du système
F. Velez - Janvier 1986
- TIGRE 32 Semantics of CAD objects for generalized databases
D. Rieu - G.T. Nguyen - Mars 1986
- TIGRE 33 CADB : Un logiciel pour la CAO
MC. Fauvet - Mars 1986
- TIGRE 34 Etude préliminaire pour le stockage et l'exploitation de comptes-rendus médicaux
G. Munoz - Avril 1986



AUTORISATION DE SOUTENANCE

DOCTORAT 3ème CYCLE, DOCTORAT-INGENIEUR, DOCTORAT USTMG

Vu les dispositions de l'Arrêté du 16 avril 1974,

Vu les dispositions de l'Arrêté du 5 juillet 1984,

Vu les rapports de M. Jack DEMONGEOT.....

M. Michel ROUX.....

M. Michel SCHOLL

Mlle. MUNOZ-BACA Guadalupe..... est autorisé

à présenter une thèse en vue de l'obtention du Doctorat Nouveau.....

Régime, Spécialité Informatique.....

Grenoble, le..... 17 JUIN 1987

Le Président de l'Université Scientifique
Technologique et Médicale



J. J. PÂYAN



Résumé

Notre travail présente les besoins d'une application médicale utilisant des données multimédia. C'est une application pilote pour le modèle et le SGBD TIGRE. L'application concerne le stockage et l'exploitation de dossiers médicaux. Une modélisation de l'application est réalisé avec le modèle TIGRE; elle montre les objets du dossier radiologique et les liens qui existent entre eux. Une deuxième modélisation et une implémentation du schéma obtenu sont réalisées avec un SGBD relationnel. Une couche multimédia entre l'application et le SGBD relationnel a été construite. Elle permet (grâce au langage SQLMED créé) la manipulation homogène de données classiques et de données multimédia.

Mots-clés : bases de données généralisées, modèle de données, données multimédia, dossier médicaux.