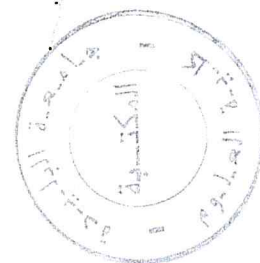


MI 6.000
P2A

République Algérienne Démocratique et Populaire.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : Intelligence Artificielle

Sujet :

Conception et Réalisation d'un Système d'Information pour la Gestion d'un CTS

Présenté par : Benmouffok Said Smain
Moktar Kamel

Promoteur : Mme Benstiti .S

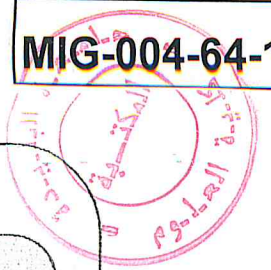
Organisme d'accueil : Centre régional de transfusion sanguine de Blida.

Soutenue le:, devant le jury composé de :

Président.....

Examineur.....

Examineur.....



Remerciement

Je tiens à remercier en premier lieu mon bon dieu tout puissant qui ma dote de toute la santé, la capacité Et la volonté pour réaliser ce mémoire.

Mes remerciements vont aussi à tout ce qui ont contribué, de près ou de loin à mener à terminer ce travail.

Notamment les enseignants qui m'ont instruit Tout au long de mon long parcours.

Pans Oublier notre promotrice Mme. P. Benstili qui nous a fait bénéficier de ses nombreux et riches conseils durant toutes nos années d'études universitaires.

Je remercie chaleureusement Tout le personnel du centre de Transfusion sanguine et Plus particulièrement Mr M. Chaib.

Dédicaces

*J'ai le grand plaisir de dédier le fruit de mes années d'études aux
êtres qui me sont les plus chers, mes parents.*

*A ma mère qui s'est donnée tant de mal pour moi et qui m'a
offert amour et soutien depuis mon premier souffle.*

*A mon cher père pour avoir mis tous les moyens à ma
disposition et pour avoir été à mes côtés tout au long de ces années
d'études.*

*A mon cher frère et mes chères sœurs et toute ma famille et
tous qui m'aime et à tous que j'aime.*

A mon binôme Hamel.

A mes chers amis Youcef, Hakim,, Madjid, Brahim,,.

*A tous mes amis d'étude, Sans oublier tous les étudiants
De l'institut d'informatique. Et à tous les étudiants Qui ont
choisis et devront choisir la spécialité INFORMATIQUE.*

El Hadj A. N.

19/04/2005

Dédicaces

Je dédie ce modeste travail à :

Mes chers parents

Toute ma famille et à ma future partenaire.....

A mes amis d'étude, Sans oublier tous les étudiants

De 5^{ème} année informatique promotion 2004/2005.

Hamel

SOMMAIRE

INTRODUCTION	1
CHAPITRE I : ETUDE DE L'EXISTANT	2
1. Introduction	3
2. Cadre de l'étude	3
3. Analyse de la situation actuelle	5
4. Etude des postes de travail	8
5. Les supports d'informations	10
5.1- Etude des documents	10
5.2. Etude des fichiers	12
6. Etude des procédures de travail	13
7. Critiques et suggestions	27
8. Solutions informatiques	29
9. Conclusion	33
CHAPITRE II : DESCRIPTION DE LA METHODE OMT	34
1. Introduction	35
2. Analyse	35
2.1. Le modèle objet	35
2.2. Le model dynamique	45
2.3. Le modèle fonctionnel	48
2.4. Relation entre les trois modèles	51
3. Conseilles pratiques à utiliser	52
4. Règles de passage de modèle objet vers le modèle relationnel	52
5. Conclusion	53
CHAPITRE III : ANALYSE ET CONCEPTION	54
1. Introduction	55
2. Analyse	55
2.1. Modélisation des objets	55
2.1.1. Identification des classes des objets	55
2.1.2. Identification des associations	56
2.1.3. Dictionnaire de données	58
2.1.4. Utilisation d'héritage	60
2.1.5. Codification proposée	60
2.1.6. Diagramme de classes	65
2.2. Le modèle dynamique	69
2.2.1. Scénario et suivis d'évènements	69
2.2.2. Diagramme d'état	75
2.3. Le modèle fonctionnel	78
3. Conception de base de données	83
3.1. Le passage de modèle objet vers le modèle relationnel	83
3.2. Liste des relations de modèle relationnel	84
3.3. Liste des tables de base de données	85
4. Conclusion	90

CHAPITRE IV : REALISATION ET MISE EN ŒUVRE.	92
1. Introduction	92
2. Environnement de développement.....	92
3. Présentation de système de gestion de base de données utilisé..	92
4. Présentation de langage de programmation utilisé.....	92
5. Sécurité	93
6. Techniques de classement proposée.....	94
7. Présentation du logiciel.....	96
8. Mise en œuvre.....	106
9. Conclusion.....	106
 CONCLUSION GENERALE.....	 107
Bibliographie.....	108
Annexe.....	110
Glossaire.....	116

LISTE DES FIGURES

Figure 1 : Organigramme de CRTS.....	4
Figure 2 : Diagramme des flux d'informations.....	6
Figure 3 : Architecture du réseau client/serveur.....	29
Figure 4 : Architecture multipostes.....	31
Figure 5 : Architecture mono poste	33
Figure 6 : Notation des classes	37
Figure 7 : Association un à un et lien.....	38
Figure 8 : Associations ternaires et liens.....	38
Figure 9 : Exemple d'associations zéro ou un.....	39
Figure 10 : Attributs de lien pour association à plusieurs.....	39
Figure 11 : Modéliser l'association « autorisation » comme une classe.....	40
Figure 12 : Exemple de nom de rôle.	40
Figure 13 : Exemple des noms de rôles dans une association attachée à même classe.	40
Figure 14 : Une association ordonné.....	40
Figure 15 : Une association qualifiée.	41
Figure 16 : Une association qualifiée sans réduire la multiplicité.	41
Figure 17 : L'association qualifiée est préférable de l'autre	41
Figure 18 : Exemple d'agrégation.....	42
Figure 19 : Exemple d'agrégation à plusieurs niveaux.....	42
Figure 20 : Héritage et généralisation en plusieurs niveaux.....	43
Figure 21 : Exemple de contraintes.	44
Figure 22 : Exemple de clefs minimales dans une association ternaire.....	44
Figure 23 : Scénario de téléphone.....	46
Figure 24 : Exemple d'un diagramme de traitement unique.....	47
Figure 25 : résumé de la notation de diagramme d'état.....	47
Figure 26 : Un DFD se présente comme un process.....	49
Figure 27 : Les chemins des données.....	49
Figure 28 : Décomposition d'une donnée en éléments.....	49
Figure 29 : Retrouver et stocker par le même process dans un réservoir.....	50
Figure 30 : la flèche (ok) est un flot de contrôle.....	50
Figure 31 : Feuille 1 (donneur du sang)	65
Figure 32 : Feuille 2 (patient et la demande du sang)	66
Figure 33 : Feuille 3 (analyse du sang)	67
Figure 34 : Feuille 4 (séparation et distribution du sang)	68
Figure 35 : Scénario de consultation.....	69
Figure 36 : Scénario de prélèvement.....	70
Figure 37 : Scénario de sérologie.....	70
Figure 38 : Scénario de groupage.....	71
Figure 39 : Scénario de distribution.....	72
Figure 40 : Scénario de paramètre.....	73
Figure 41 : Scénario des analyses.....	74
Figure 42 : Diagramme d'état de donneur.....	75
Figure 43 : Diagramme d'état de flacon.....	76
Figure 44 : Diagramme d'état de stocke de paramètre.....	77
Figure 45 : DFD de consultation.....	78
Figure 46 : DFD de prélèvement.....	79
Figure 47 : DFD de sérologie.....	80
Figure 48 : DFD de groupage.....	81
Figure 49 : DFD de distribution.....	82

Figure 50 : Classement des fiches des donneurs.....	94
Figure 51 : Classement des demandes de sang et fiches de distribution.....	95
Figure 52 : Schéma global de fonctionnement de logiciel CRTS	96
Figure 53 : Interface de saisir de mot de passe de l'utilisateur.....	105

INTRODUCTION GÉNÉRALE

Dans notre vie quotidienne l'information a pris une grande ampleur dans les différents domaines et secteurs d'activité, des quantités importantes d'informations sont échangées, manipulées, traitées chaque jour, à tout les niveaux des organisations et statuts. Ce qui est nécessaire de la rendre structurée, et de l'organiser au sein d'un « système d'information » et de la stocker dans une structure fiable, « base de données » afin d'améliorer ses performances et ainsi garder sa présence dans un monde compétitif.

Un système d'information est un système organisé de ressources, de personnes et de structures qui évoluent dans une organisation et du comportement coordonné, et qui vise à atteindre un objectif commun.

Un système d'information est caractérisé par :

- ✓ Des frontières
- ✓ Est doté d'une structure.
- ✓ Un ensemble d'activités afin d'atteindre un but.
- ✓ Son évolution dans un environnement.

Néanmoins la conception des systèmes d'informations est en générale une tâche très difficile, ce qui nécessite des simplifications et des abstractions du système étudié en construisant des modèles.

Le but principal de « la modélisation » est de réduire la complexité. C'est dans cette perspective que plusieurs méthodes ont été proposées. Chaque méthode offre une notation pour supporter les différentes vues d'un système ainsi qu'un processus pour orienter le concepteur dans sa tâche de développement d'un système.

Les systèmes d'informations ont connu des générations successives de méthodes d'analyse. Dans notre étude nous avons optés pour la méthode de modélisation OMT (Object Modelling Technique).

Le travail présenté dans ce mémoire porte sur la conception et la réalisation d'un système qui fonctionne sur un réseau client/serveur pour la gestion d'un centre de transfusion sanguine. Ce travail est organisé en cinq chapitres :

Dans le premier nous présentons une analyse globale de système l'existant pour pouvoir détecter les anomalies de mauvais fonctionnement de système existant.

Dans le deuxième chapitre nous exposons la méthode OMT.

Dans la troisième chapitre nous abordons la conception de notre nouveau système suivant la méthode OMT.

Dans la quatrième et le dernier chapitre nous présentons les logiciels qu'on a utilisés pour la création de notre base de données ainsi que la réalisation de notre système, et nous abordons aux différentes étapes qu'on a suivent pour la réalisation de système.

CHAPITRE I

ETUDE DE L'EXISTANT

1. Introduction :

Toute étude conceptuelle devra reposer sur une bonne connaissance du système existant.

Dans ce fait une analyse complète de la situation actuelle est impérative afin de pouvoir cerner les causes de défaillance du système et cette partie fera l'objet de cette analyse qui englobe une description détaillée du domaine dont on souhaite améliorer le fonctionnement.

2. Cadre de l'étude :**2.1. Présentation de l'organisme d'accueil :****➤ Historique de CRTS.**

Le centre régional de transfusion sanguine de BLIDA (CRTS) a été inauguré par le ministre de la santé et de la population monsieur YAHYA GUIDOUM en 1997.

Le centre de transfusion sanguine se situe en approximation de l'hôpital Ferroudja, rue des frères Ramoul, Blida.

Le centre compte 39 employés, parmi lesquels 6 médecins et 19 techniciens.

➤ Emission de CRTS.

Le centre de transfusion sanguine CRTS consiste :

- A satisfaire les malades hospitalisés de la région en terme du sang.
- A assurer les analyses du sang en vue de dépister les maladies contagieuses.
- En la séparation du sang à ses différents constituants au besoin du malade.
- En la sensibilisation des donneurs du sang et ce dans les différents secteurs publics et privés.
- A s'approcher du donneur pour faciliter l'opération du don.

Organigramme de l'organisme d'accueil :

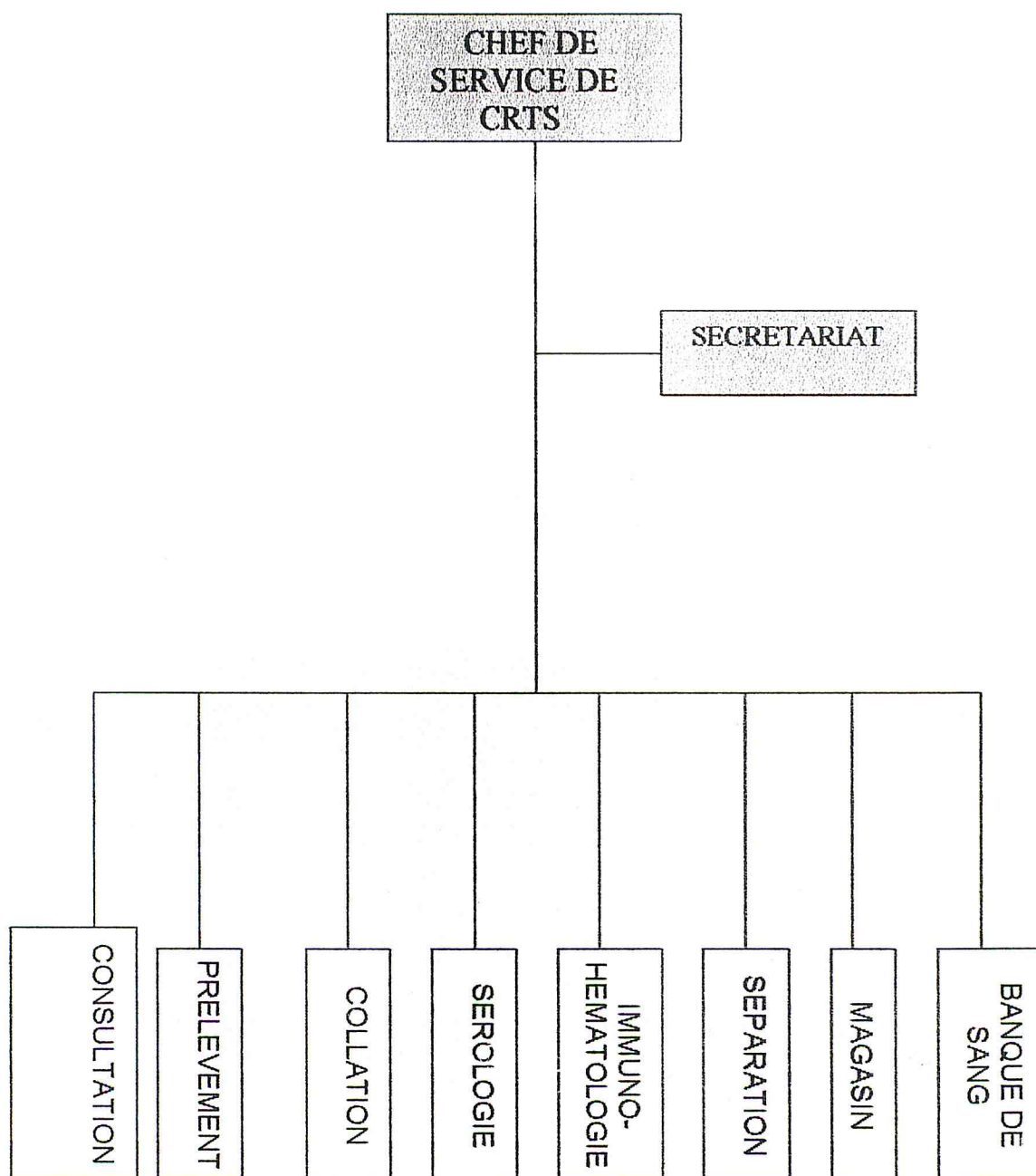


Figure1 : Organigramme de CRTS

2.2. Position de problème :

Les problèmes vécus par les responsables et les utilisateurs sont :

- L'insécurité et la perte de fiches des donneurs.
- Mauvais archivage des documents.
- Accès aux informations de donneur difficile et long.
- Difficulté de contrôler le stocke des flacons du sang.
- Difficulté de contrôler le stocke des paramètres.
- Problème de suivi des fiches de distribution du sang.
- La circulation des informations entre les services se fait manuellement, ce qui signifie la possibilité de perte d'informations.

2.3. Présentation du sujet :

Notre travail a été conçu pour répondre aux objectifs suivants:

- Assurer une meilleure sécurité pour les informations de donneur avec un meilleur stockage de ces informations.
- Pour avoir un bon suivi et contrôle pour les flacons ainsi les fiches de distribution du sang.
- Diminué le temps d'accès aux informations de donneur, ainsi le temps d'attendre des donneurs.
- Contrôler automatiquement la banque du sang.
- Contrôler automatiquement le stocke des paramètres.
- Limité les responsabilités pour chaque service.

3. Analyse de la situation actuelle :

3.1. Fonctionnement de CRTS :

- Recevoir du sang à partir des dons des donneurs du sang.
- Faire des analyses de groupage et de sérologie au sang.
- Faire des séparation au sang selon les besoins et le contenu de stocke.
- Stocker les flacons du sang dans la banque du sang.
- Distribuer le sang sur les hôpitaux pour soigner ses patients.

3.2. Flux d'informations :

Le graphe des flux représente les échanges d'informations entre les différents acteurs de l'organisme, un acteur peut être interne ou externe au champ d'étude.

La construction du graphe des flux est faite à partir d'interviews des responsables de différents services de CRTS, il doit permettre de représenter les flux d'informations et de fixer les limites du champ d'étude.

➤ Diagramme des flux d'informations:

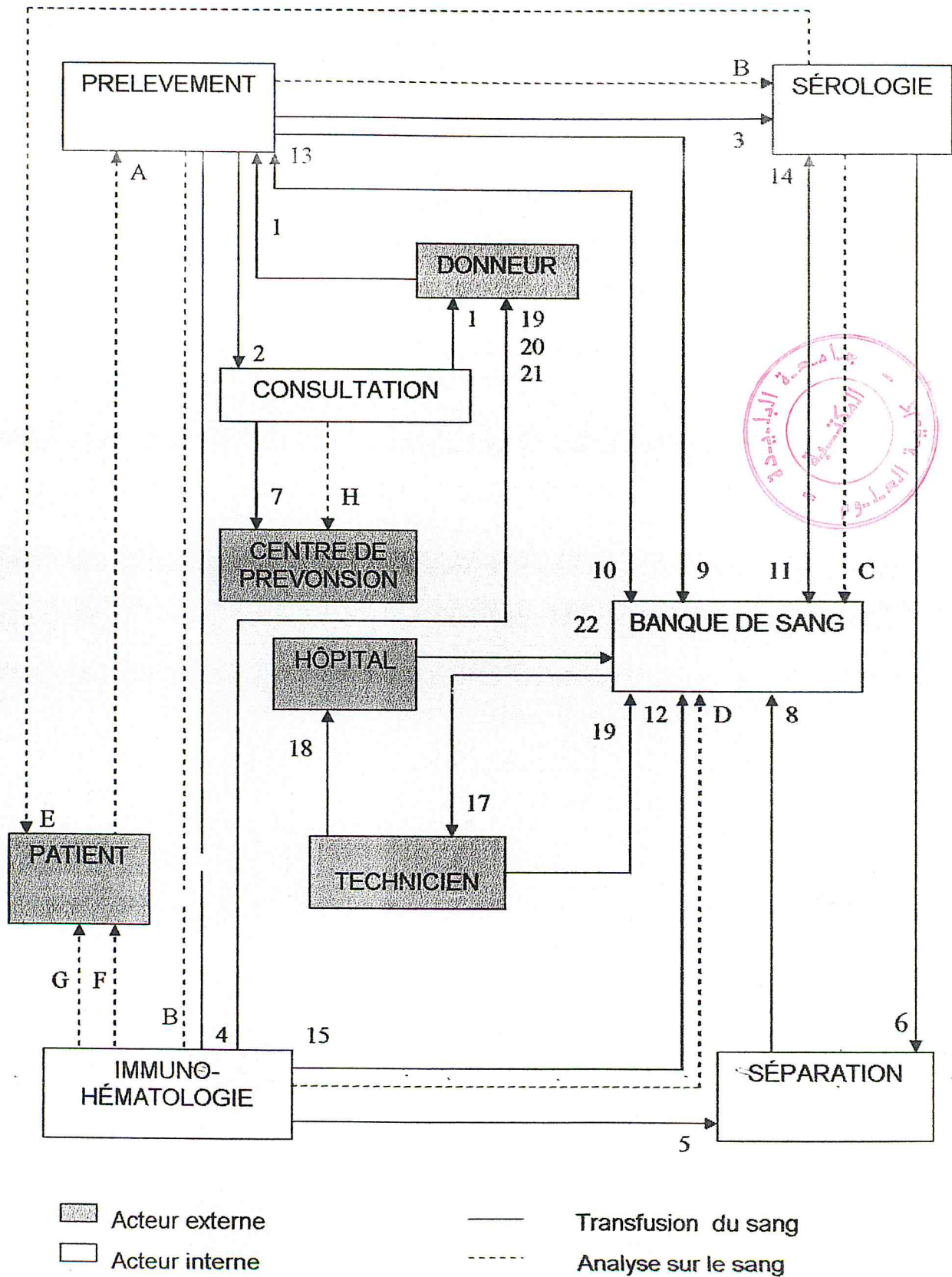


Figure2 : Diagramme des flux d'informations

➤ Description du flux d'informations :

N°	Description
01	Fiche de prélèvement.
02	Fiche de prélèvement de sang mis à jour.
03	Tube du sang
04	Tube du sang + poche du sang.
05	Poche du sang.
06	Résultats de sérologie.
07	Relevé des malades à déclaration obligatoire.
08	Poche du sang.
09	Etat de prélèvement.
10	Registre des entrées du sang.
11	Registre de sérologie des donneurs.
12	Registre de groupage et registre des testes de compatibilité.
13	Registre de prélèvement.
14	Registre de sérologie des donneurs.
15	Registre de groupage et registre des testes de compatibilité.
16	Formulaire de demande de sang + carte professionnel.
17	Poche du sang + fiche de distribution nominative +carte professionnel.
18	Poche de sang + fiche de distribution nominative.
19	Carte de donneur de sang volontaire.
20	Carte de groupage provisoire.
21	Carte de groupage.
22	Fiche de distribution nominative mise à jour.
A	Ordonnance.
B	Tube du sang +ordonnance.
C	Registre de sérologie des externes.
D	Registre des externes.
E	Fiche de sérologie.
F	Carte de groupage provisoire.
G	Carte de groupage.
H	Relevé des malades à déclaration obligatoire.

4. Etude des postes de travail :

Le but de l'étude d'un poste de travail est de recueillir toutes les opérations effectuées et toutes les informations manipulées au niveau de chaque poste.

Dans notre étude nous ne prenons en considération que les postes ayant un rôle actif dans l'application, les postes entrées dans le champ d'étude de l'organisme.

Ces postes de travail sont :

1. service de consultation.
2. service de prélèvement.
3. service de sérologie.
4. service d'immuno-hématologie.
5. service de séparation.
6. banque de sang.

Exemple de description des postes de travail :

Description de poste du travail (Consultation)				
Désignation du poste : consultation Service au quel est attaché le poste : chef du service Responsable de poste : médecin Moyens humains : 02				
1. Responsabilités de ce poste taches à accomplir Responsabilités : - Consulter les donneurs. - Etablissement de fiche de prélèvement de sang. - Mise à jour de fiche de donneur. - Mise à jour de registre de donneur. - Mise à jour de relevé du malade. - Envoyer le relevé des malades au centre de prévention.				
Taches	Fréquence	Délais		Remarques
		Prévus	Réels	
-Etablissement de fiche de prélèvement du sang.	-Un exemplaire pour chaque donneur accepté au don.	1 heure	Très variable	La fiche n'est pas utilisée.
-Mise à jour de fiche de donneur.	- Pour chaque don.	3 mois	Variable	
-Mise à jour de registre de donneur.	- Pour chaque don.	3 mois	Variable	
-Mise à jour de relevé du malade.	-Une fois par jour.	24 heures	Variable	
-Etablissement des relevés des maladies à déclaration obligatoire.	-Une fois pour chaque détection du maladie.	24 heures	Variable	
-Envoyer des relevés des maladies à déclaration obligatoire.		Aucun	Variable	Un relevé peut contenir plusieurs malades.
2. Documents				
Documents provenant à ce poste				
Du poste	Désignation du document	Fréquence	exp	
-Service de prélèvement	- Fiche du prélèvement du sang.	- Pour chaque donneur accepté au don.	01	
Documents remplir par ce poste				
Désignation	Fréquence	exp		
- Fiche de prélèvement du sang.	-Pour chaque don.	01		
-Relevé des maladies à déclaration obligatoire	-Aucun.	01		

Documents diffusés par ce poste						
Désignation	Fréquence		Destinataire		exp	
-Fiche de prélèvement du sang.	-Pour chaque donneur accepté au don.		-Service de prélèvement.		01	
-Relevé des maladies à déclaration obligatoire.	- Aucun.		- Centre de prévention.		01	
3. Fichiers utilisés par ce poste						
Désignation	Support	Localisation du fichier (service)	Désignation des articles	Nombre d'article	Nombre de caractères par article	Opérations réalisées
- Fichier de consultation.	-Registre.	-Consultation	-Donneur	Variable	122	-Mise à jour et consultation de ces fichiers à l'arrivée de donneur.
- Fichier de donneur.	-Fiche cartonnées.	-Consultation	-Donneur	27	331	
- Relevé du malade.	-Registre.	-Consultation	-Malade	Variable	200	
-Relevé des maladies à déclaration obligatoire.	-Feuille	-Consultation	-Malade	Variable	225	-Mise à jour pour chaque détection de malade.
4. Remarques et suggestions						
-Perd beaucoup du temps pour trouver la fiche de donneur à l'arrivée de l'ancien donneur.						
-Utilise des documents non spécialisé pour ce travail (bon pour).						
- Mauvais archivage pour les documents du poste.						
- Manque de sécurité pour les données de donneur.						
- Toutes les taches sont réalisées manuellement.						

5. Les supports d'informations :

5.1- Etude des documents :

L'objet de cette étude est de faire le point sur la qualité des documents et de réaliser un diagnostic qui nous permet d'ajouter un document utile, supprimer un autre inutile ou modifier un document existant.

Il s'agit d'analyser les rubriques constituant les documents et de déterminer les rubriques manquantes (ou celles qui sont en plus ou redondantes).

Liste des documents :

N°	CODE	DÉSIGNATION
01	ORD	Ordonnance.
02	CG	Carte de groupage.
03	CGP	Carte de groupage provisoire.
04	SRL	Fiche de sérologie.
05	FIDS	Fiche d'identification du donneur de sang.
06	FDN	Fiche de distribution nominative.
07	BON	Bon du prélèvement.
08	FDS	Formulaire de demande de sang.
09	RMO	Relevé des maladies à déclaration obligatoire.

Légende de type de données :

Lettre utilisée	Type
A	Alphabétique
N	Numérique
D	Date

Exemple de description des documents :

Fiche d'analyse de document (Carte de groupage)			
1-Characteristiques du document			
Code : CG Désignation : carte de groupage. Nature du document : saisie - résultat Créer par : service immuno –hématologie. Destiné à : malade/donneur. Objectif : utilise comme une carte personnel. Format : recto.			
2-Etude des parties du document			
Emplacement	Rubrique	Signification	Type
Entête	المركز الاستشفائي الجامعي بالبيدة- -Centre de transfusion sanguine de BLIDA. -Unité hospitalo-universitaire de BLIDA. -Service laboratoire de biologie/HCB. -Carte de groupage.	-centre de transfusion sanguine de BLIDA.	
Corps	Nom Prénom Service Adresse Groupé le	Nom de malade ou donneur. Prénom de malade ou donneur. Service qui remplit la carte. Adresse de malade ou donneur. Date de groupage.	A A A A D

Emplacement	Rubrique	Signification	Type
Corps	Sous le N°	Numéro de groupage.	N
	Au service laboratoire de biologie/HCB. Groupe sanguin	Groupe sanguin de malade ou donneur.	A
Base	Date Le chef d'unité	Date Visa de chef de service	D
3. Remarques et suggestions - Entête est très long. - Document n'est pas spécialisé pour l'unité. - Les cartes sont différentes de groupe sanguin à l'autre.			

5.2. Etude des fichiers :

L'étude des fichiers permet de connaître :

- L'objectif et l'importance de chaque fichier.
- Les informations contenues dans les fichiers ainsi leur type et leur taille.
- Les caractéristiques du chaque fichier.

Liste des fichiers :

N°	Code	Désignation
01	FPD	Etat de prélèvement.
02	FDSV	Carte de donneur de sang volontaire.
03	FD	Fiche de donneur.
04	FC	Registre de consultation.
05	RM	Registre des malades.
06	FES	Registre des entrées de sang du malade.
07	FP	Registre du prélèvement.
08	FDS	Registre de distribution du sang.
09	FSS	Registre des sorties du sang.
10	FG	Registre du groupage.
11	FEX	Registre des externes.
12	FS	Registre de séparation.
13	FTC	Registre des testes de compatibilité.
14	FSD	Registre de sérologie.
15	FSEX	Registre de sérologie des externes.
16	FCGN	Registre des consignés.
17	FBQ	Registre de banque.
18	RMDO	Relevé des maladies à déclaration obligatoire.
19	RPT	Registre des patients.

Exemple de description des fichiers :

Fiche de description de fichier (Etat de prélèvement)			
Code : FPD Désignation : Etat de prélèvement. Nature : interne Support : papier blanche.			
Rubriques	Signification	Type	Taille
Date	Date de prélèvement.	D	10
Nom	Nom de patient.	A	20
Prénom	Prénom de patient.	A	20
Groupe sanguin	Groupe sanguin.	A	03
Produit sanguin	Produit sanguin demandé.	A	20
N°de flacon	Numéro de flacon.	N	05
Quantité	Quantité demandée.	A	06

6. Etude des procédures de travail :

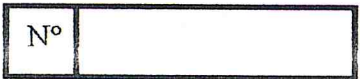
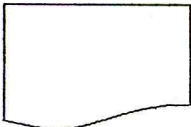
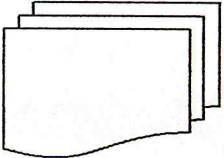
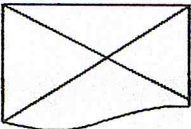
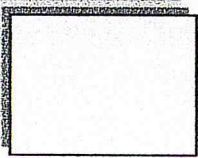
Représente un ensemble des procédures ainsi les informations traitées par ces procédures, les événements qui y ont déclenchée ces procédures et le lieu et le temps de déclenchement.

Une procédure est un ensemble des taches exécutées par un ou plusieurs postes pour objectif commun.

Liste des procédures :

1. Consultation.
2. Prélèvement pour les donneurs.
3. Prélèvement pour les externes.
4. Séparation.
5. Groupage des externes.
6. Groupage des donneurs.
7. Sérologie pour les donneurs.
8. Sérologie pour les externes.
9. Stocke et distribution du sang.

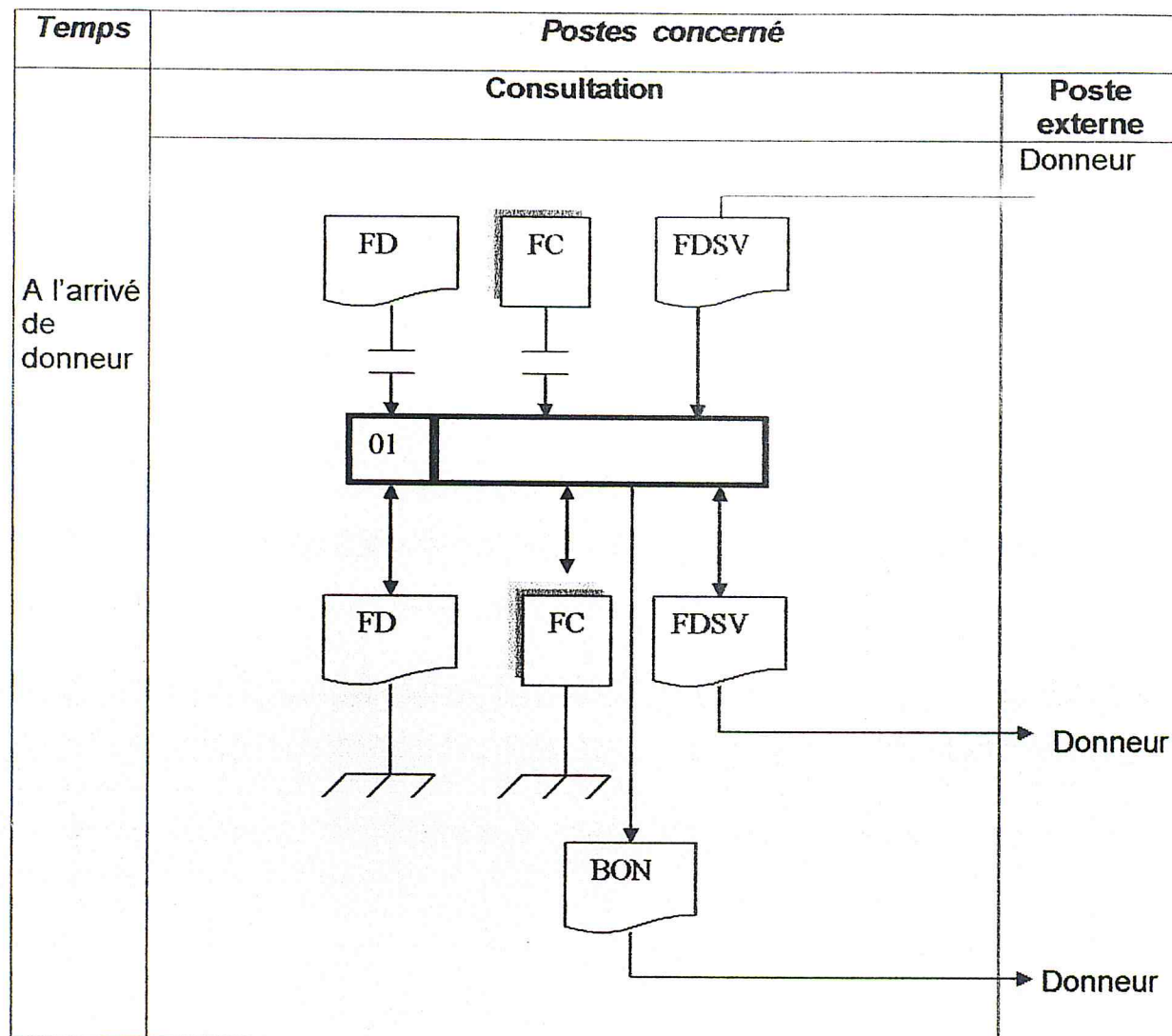
Tableau de symboles utilisés dans les graphes de circulation d'informations :

Symboles	Description
	Opération
	Document
	Document en plusieurs exemplaires.
	Destruction du document
	Registre
	Mise à jour
	Classement
	Sens de circulation de l'information
	Archivage
	Attente

Description des codes :

Code	Désignation
FD	Fiche de donneur.
FC	Registre de consultation.
FDSV	Carte de donneur de sang volontaire.
BON	Bon de prélèvement.
FES	Registre des entrées du sang.
FP	Registre de prélèvement.
FDS	Formulaire de demande de sang.
FPD	Etat de prélèvement demandé.
ORD	Ordonnance.
FS	Registre de séparation.
FEX	Registre des externes.
CG	Carte de groupage.
CGP	Carte de groupage provisoire.
FTC	Registre des testes de compatibilité.
FG	Registre de groupage.
FSD	Registre de sérologie des donneurs.
FSEX	Registre de sérologie des externes.
FSGX	Fiche de sérologie.
FCGN	Registre des consignes.
FBQ	Registre de banque.
FSS	Registre des sorties du sang.
FDBS	Registre de distribution du sang.
FDN	Fiche de distribution nominative.
RDMO	Relevé des malades à déclarations obligatoires.
FPT	Registre des patients.

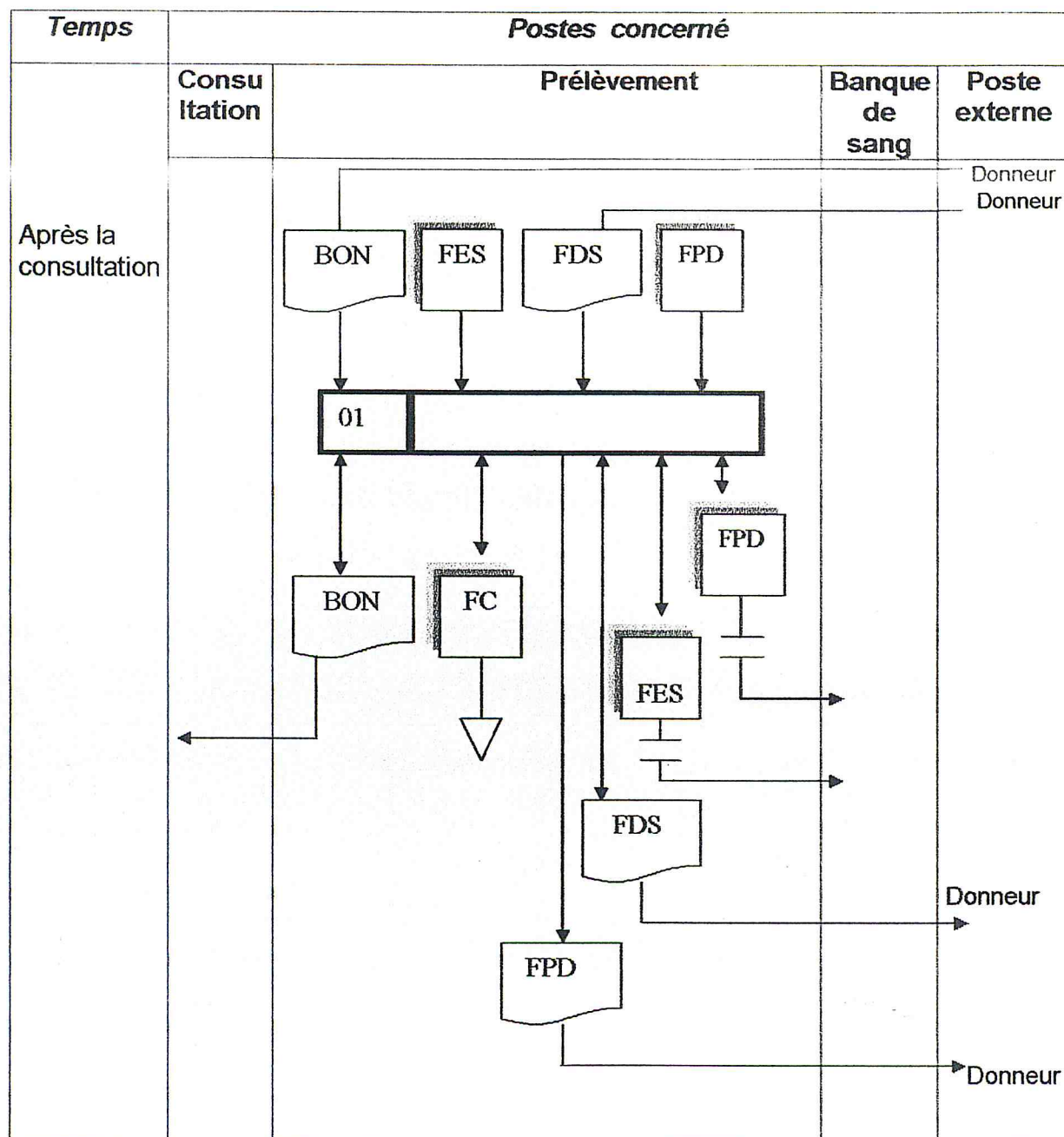
1. Procédure de consultation :



Description des opérations :

N°	Désignation
01	Vérification de carte de donneur volontaire. Consultation de fiche de donneur. Consultation médicale au donneur. Mise à jour de carte de donneur volontaire. Mise à jour de carte de consultation. Mise à jour de fiche de donneur. Etablissement de bon de prélèvement. Classement de fiche de donneur. Classement de registre de consultation.

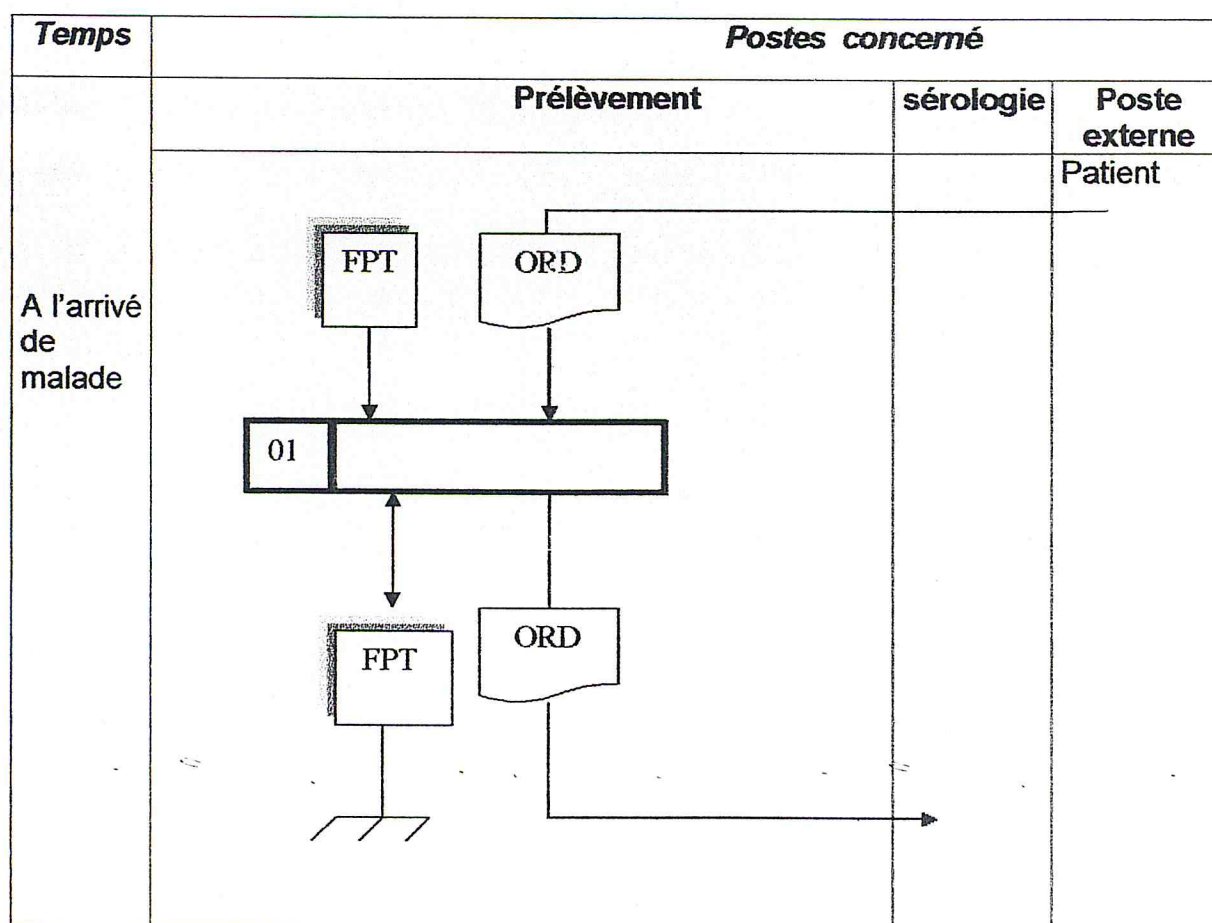
2. Procédure de prélèvement des donneurs :



Description des opérations :

N°	Désignation
01	Consultation de formulaire de demande du sang. Consultation bon de prélèvement. Mise à jour de bon de prélèvement. Mise à jour de registre de prélèvement. Mise à jour de registre des entrées du sang. Mise à jour de formulaire de demande du sang. Etablissement de l'état de prélèvement demandé. Prélèvement d'une poche du sang. Remplir deux tube avec le sang de cette poche. Classement de registre de prélèvement.

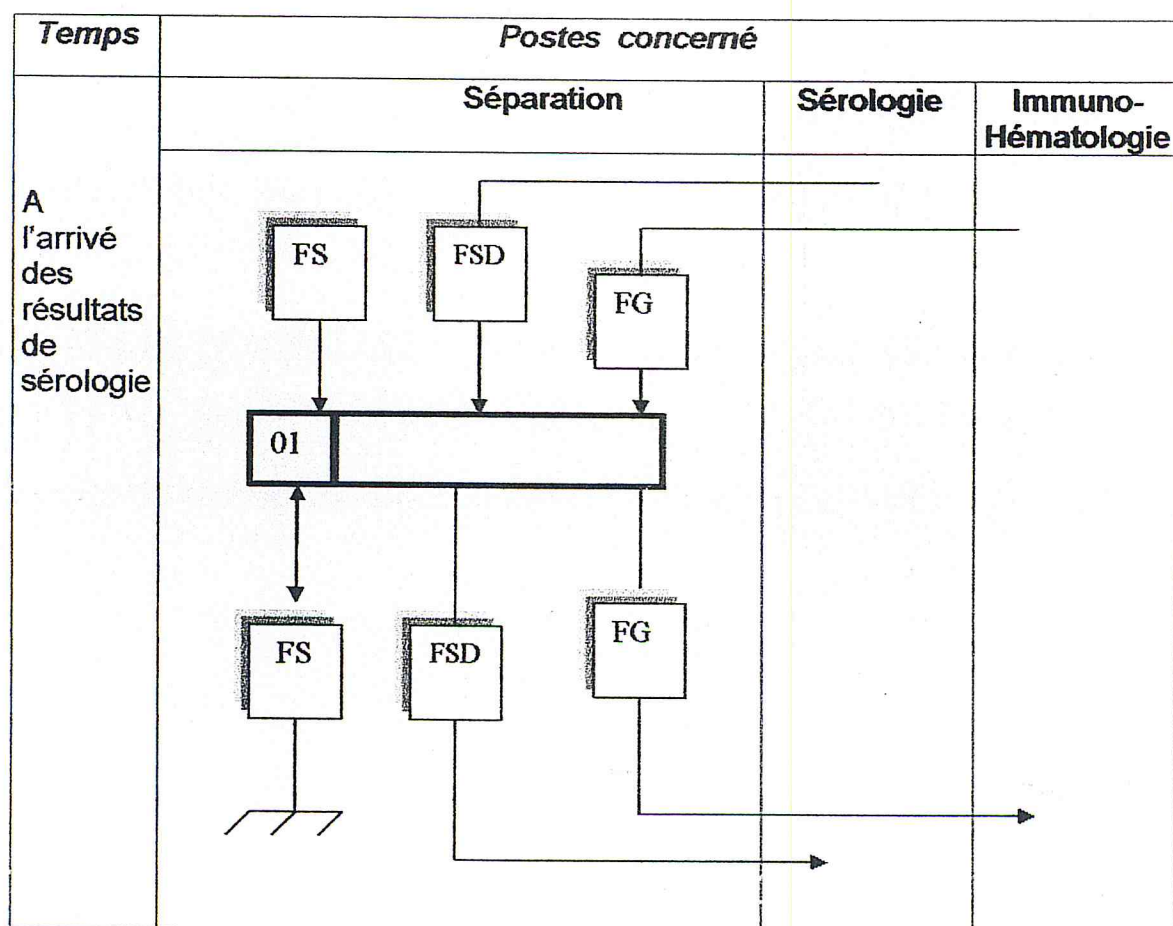
3. Procédure de prélèvement des patients :



Description des opérations :

N°	Désignation
01	Consultation de l'ordonnance. Mise à jour de registre des patients. Prélèvement de tube du sang.

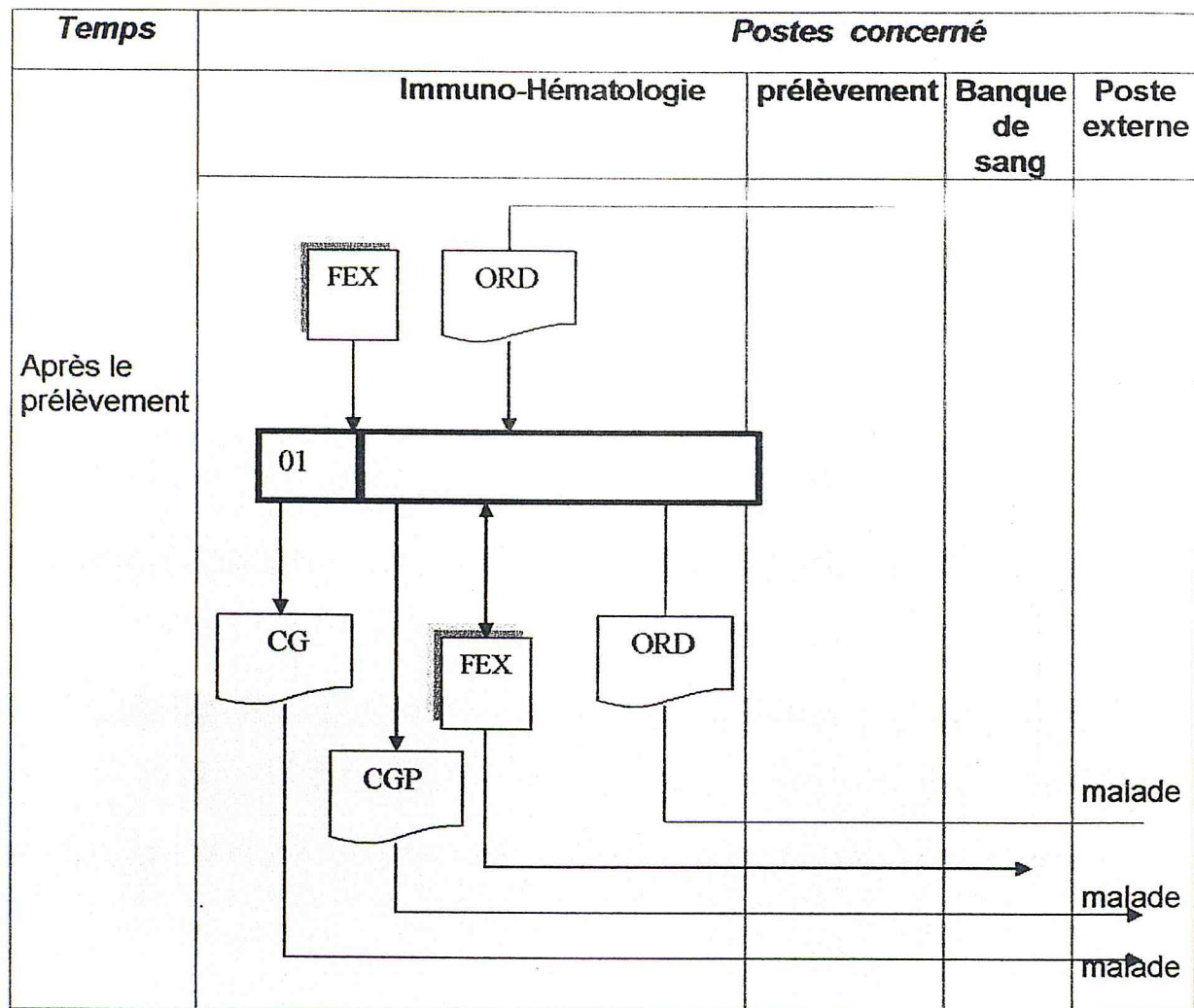
4. Procédure de séparation :



Description des opérations :

N°	Désignation
01	Consultation de registre de sérologie des donneurs. Consultation de registre de groupage. Séparation de sang total en plusieurs produits sanguins. Mise à jour de registre de séparation. Classement de registre de séparation.

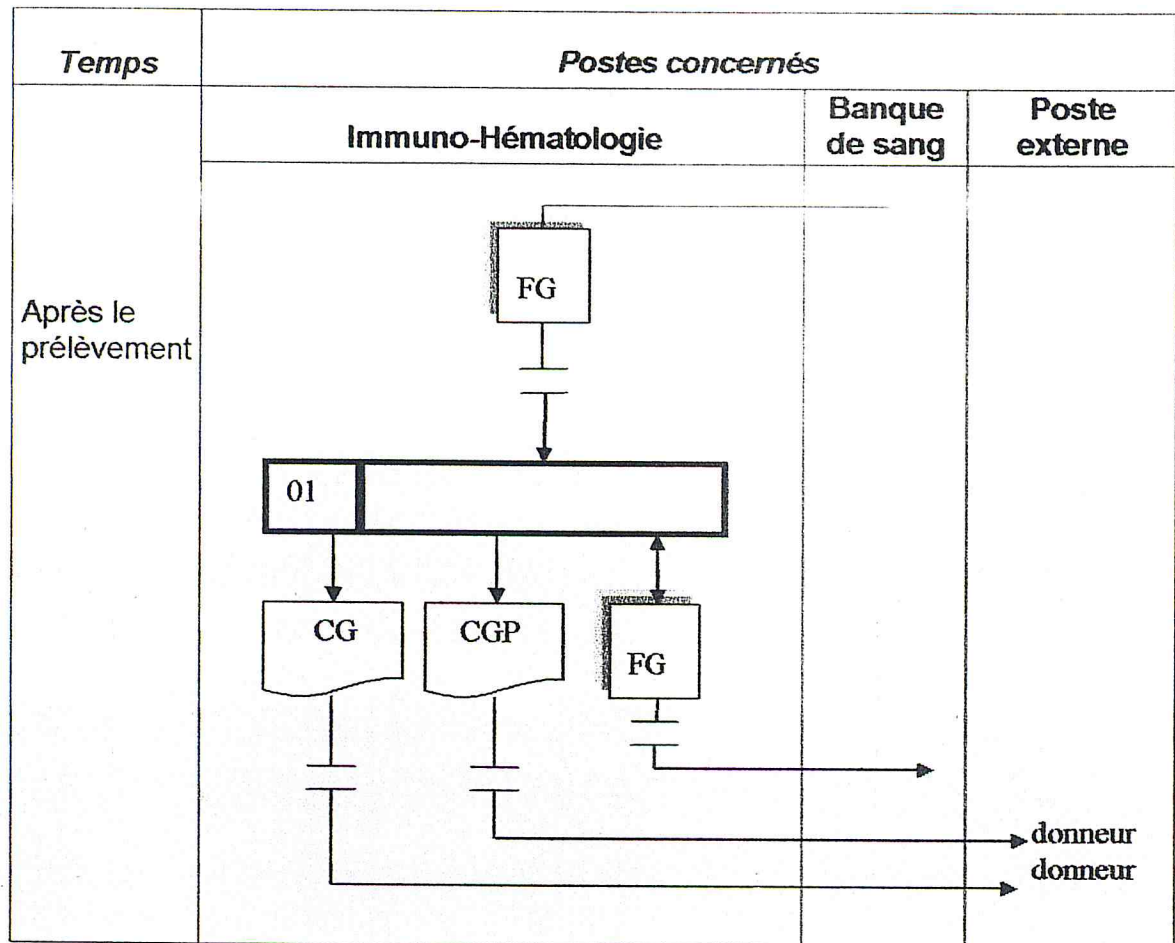
5. Procédure de groupage des externes :



Description des opérations :

N°	Désignation
01	Consultation de l'ordonnance. Consultation de registre de groupage. Groupage de sang de tube. Mise à jour de registre des externes. Etablissement de carte de groupage. Etablissement de carte de groupage provisoire.

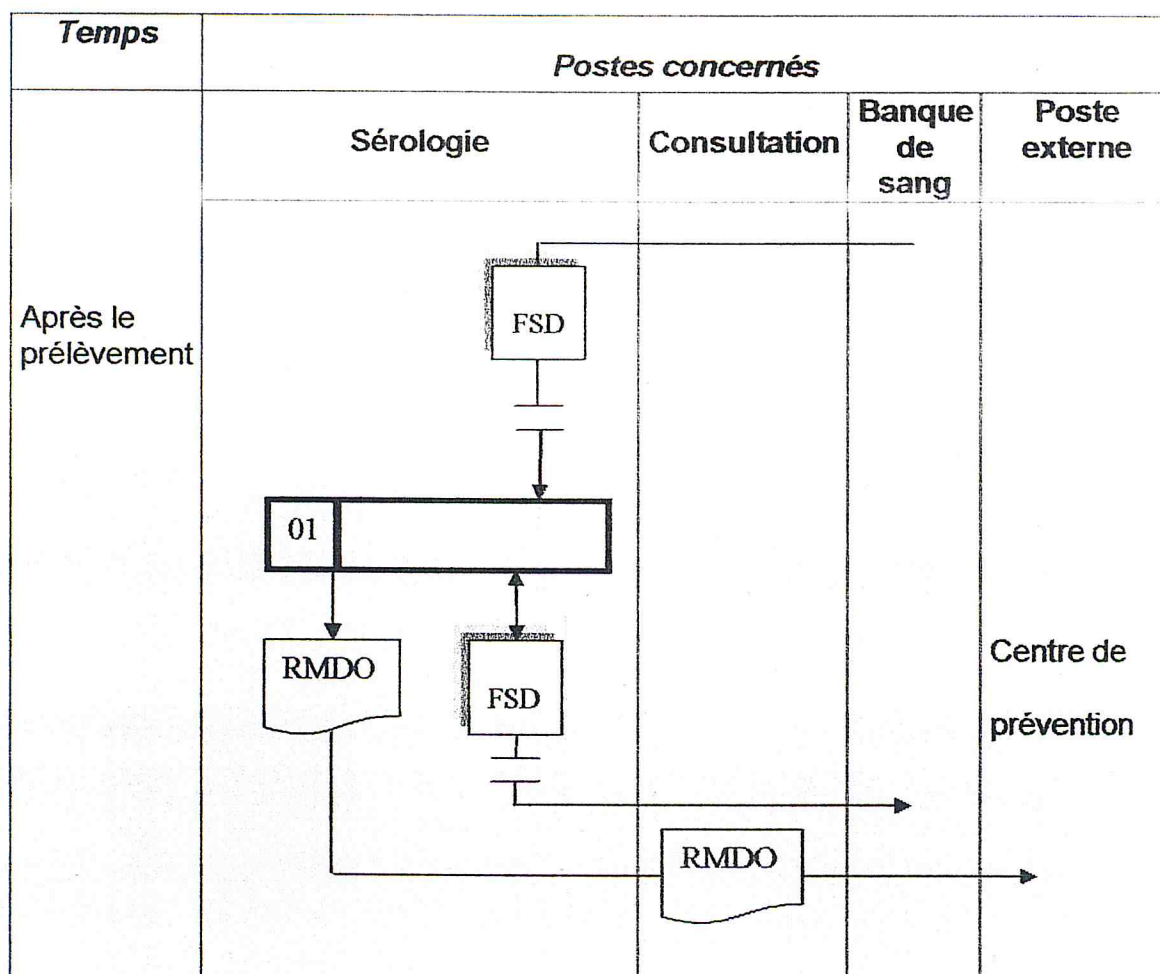
6. Procédure de groupage des donneurs :



Description des opérations :

N°	Désignation
01	Consultation de registre de groupage. Groupage de sang de tube. Groupage de sang de poche et la comparé avec le groupage de tube. Mise à jour de registre de groupage. Etablissement de carte de groupage. Etablissement de carte de groupage provisoire.

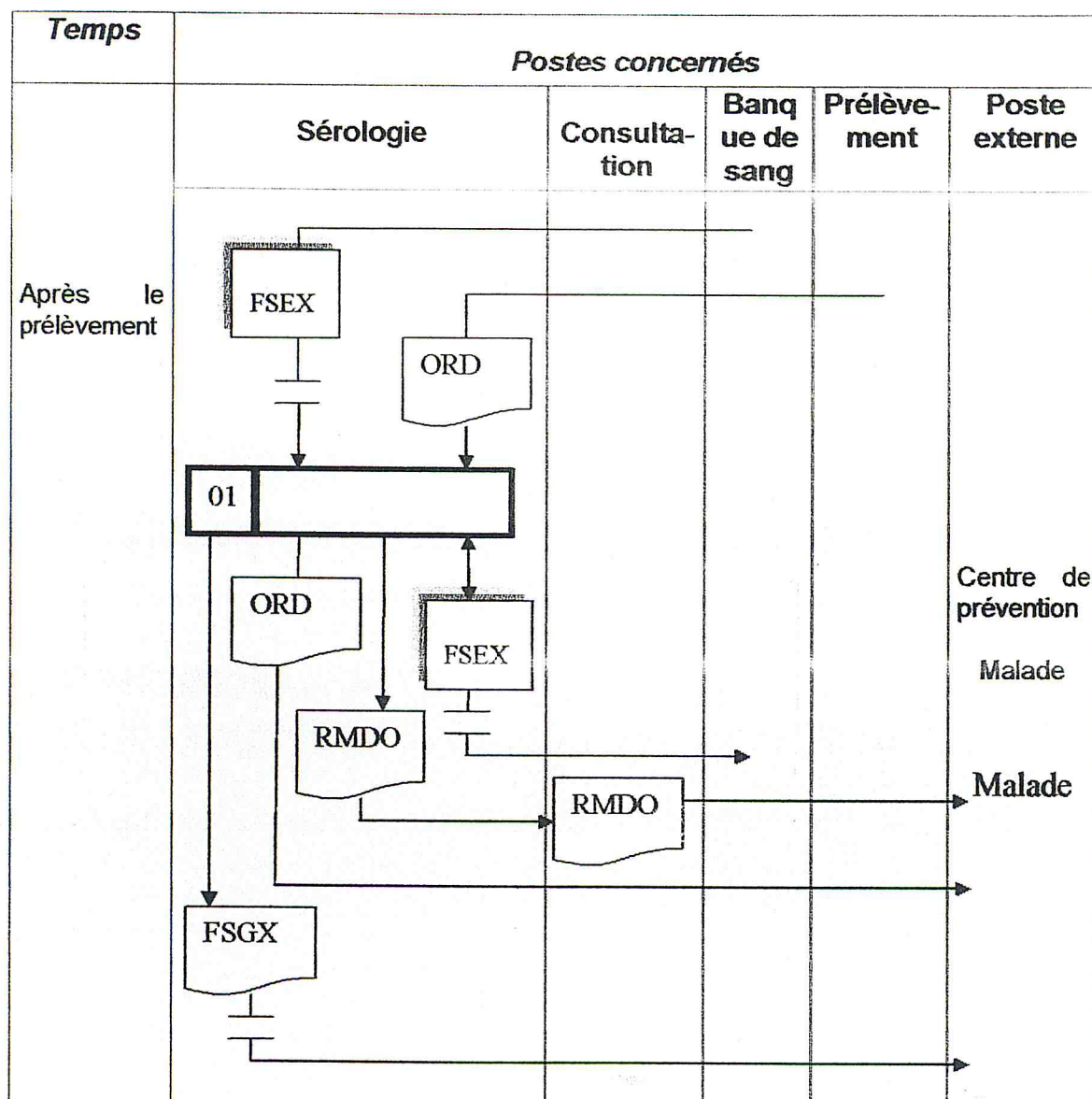
7. Procédure de sérologie des donneurs :



Description des opérations :

N°	Désignation
01	Réalisé des examen de sérologie. Mise à jour de registre de sérologie des donneurs. Établissement de relevé des malades à déclaration obligatoire.

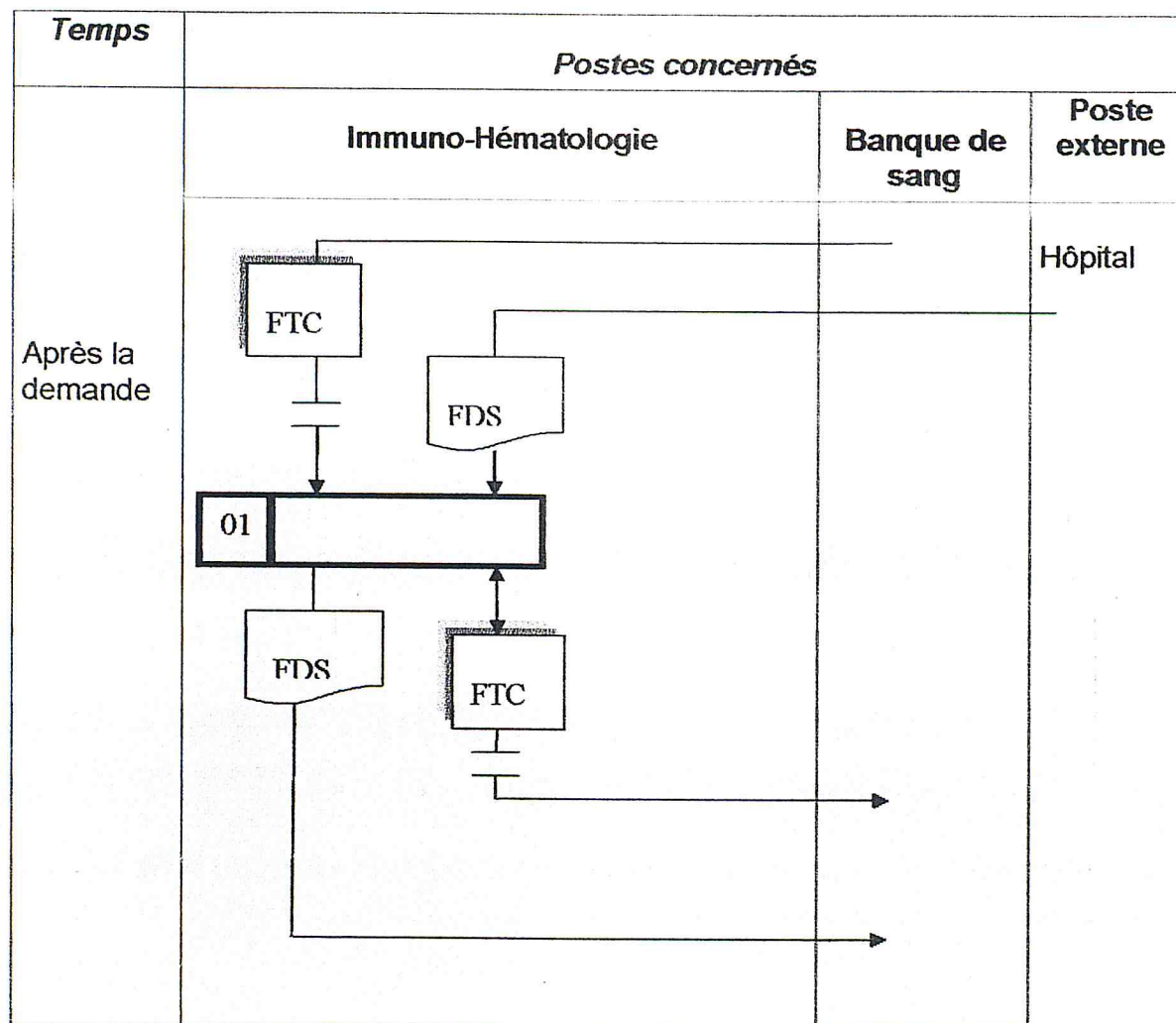
8. Procédure de sérologie des externes :



Description des opérations :

N°	Désignation
01	Consultation de l'ordonnance. Réalisation des examens de sérologie. Mise à jour du registre de sérologie des externes. Établissement de fiche de sérologie. Établissement de relevé des malades à déclaration obligatoire.

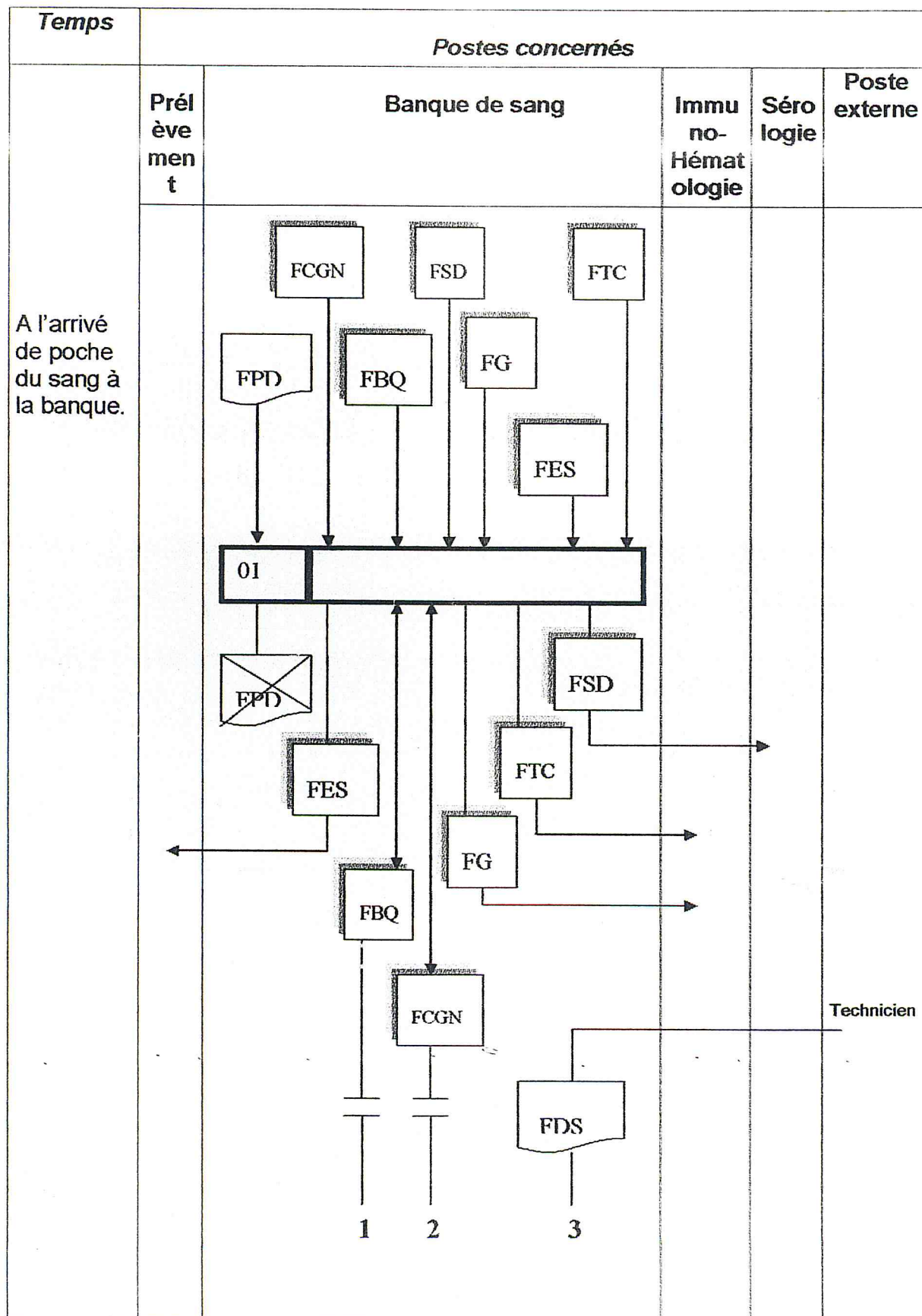
9. Procédure des testes de compatibilité :



Description des opérations :

N°	Désignation
01	Consultation de formulaire de demande de sang. Réalisation des tests de compatibilité. Mise à jour du registre des tests de compatibilité.

10. Procédure de stocke et distribution du sang :



[illegible]

Description des opérations :

N°	Désignation
01	Consultation de registre de groupage. Consultation de registre des testes de compatibilité. Consultation de registre de sérologie des donneurs. Consultation de registre de prélèvement. Consultation de registre des entrées du sang. Mise à jour de registre de banque. Mise à jour de registre des consignes. Classement des poches dans la banque.
02	Consultation de formulaire de demande de sang. Consultation de registre de banque. Consultation de registre des consignes. Mise à jour de registre des sorties du sang. Mise à jour de registre de banque. Classement de registre de banque. Mise à jour de registre des consignes. Classement de registre des consignes. Mise à jour de registre de distribution du sang. Classement de registre de distribution du sang. Etablissement de fiche de distribution nominative.
03	Consultation de fiche de distribution nominative. Consultation de registre des sorties du sang. Mise à jour de registre des sorties du sang. Classement de registre des sorties du sang. Archivage de fiche de distribution nominative.

7. Critiques et suggestions :

Après une étude approfondie et globale sur la situation actuelle, on a recensé quelques anomalies dû au mauvais fonctionnement de système existant qui c'étal à deux niveaux :

- Niveau informationnel.
- Niveau organisationnel.

1- Critiques :**a)- Niveau informationnel :**

- Présence des rubriques inutiles (exp : nombre d'enfants).
- Mauvaise arrangement des données sur les documents (exp : carte de groupage provisoire).
- Les documents sont remplis manuellement.
- Mauvaise circulation de l'information entre les services.

- Les informations de donneur circulent entre les différents services, par contre la loi interne limite leur circulation dans le service de consultation.

b)-Niveau organisationnel :

- Manque de sécurité des informations.
- Les opérations de recherche demande beaucoup de temps car les informations sont gérées manuellement.
- Mauvais archivage des documents.
- Manque des codifications des objets.

2- Suggestions :

a) – Niveau informationnel :

- Eliminer les rubriques inutiles et ajouter les rubriques manquant.
- Restructuré les documents pour qu'elle soit respecté les parties de chaque document (entête fixe, corps, base).
- Automatiser l'établissement des documents.
- Assurer la sécurité des informations.
- Assurer la circulation des informations entre les services.

b)- Niveau organisationnel :

- Recrutement des cadres qualifiés en informatique.
- Prévoir les codifications manquantes.
- Utilisation d'outil informatique pour assurer l'accès rapide aux informations, et assurer la sécurité des informations.
- Prévoir une bonne répartition des taches et fixer les responsabilités pour chaque service.
- Choisir une bonne méthode d'archivage des documents.

8. Solutions informatiques :

Dans l'étude de l'existant, nous avons pu connaître quasi-parfaitement la situation au sein du service, du point de vue de la répartition des postes de travail ainsi que le réseau informationnel qui les relie.

De ce fait, nous avons pu aboutir à quatre solutions envisageables que nous détaillons par la suite :

1^{er} Solution : réseau de terminaux :

La solution est représentée par la configuration suivante :

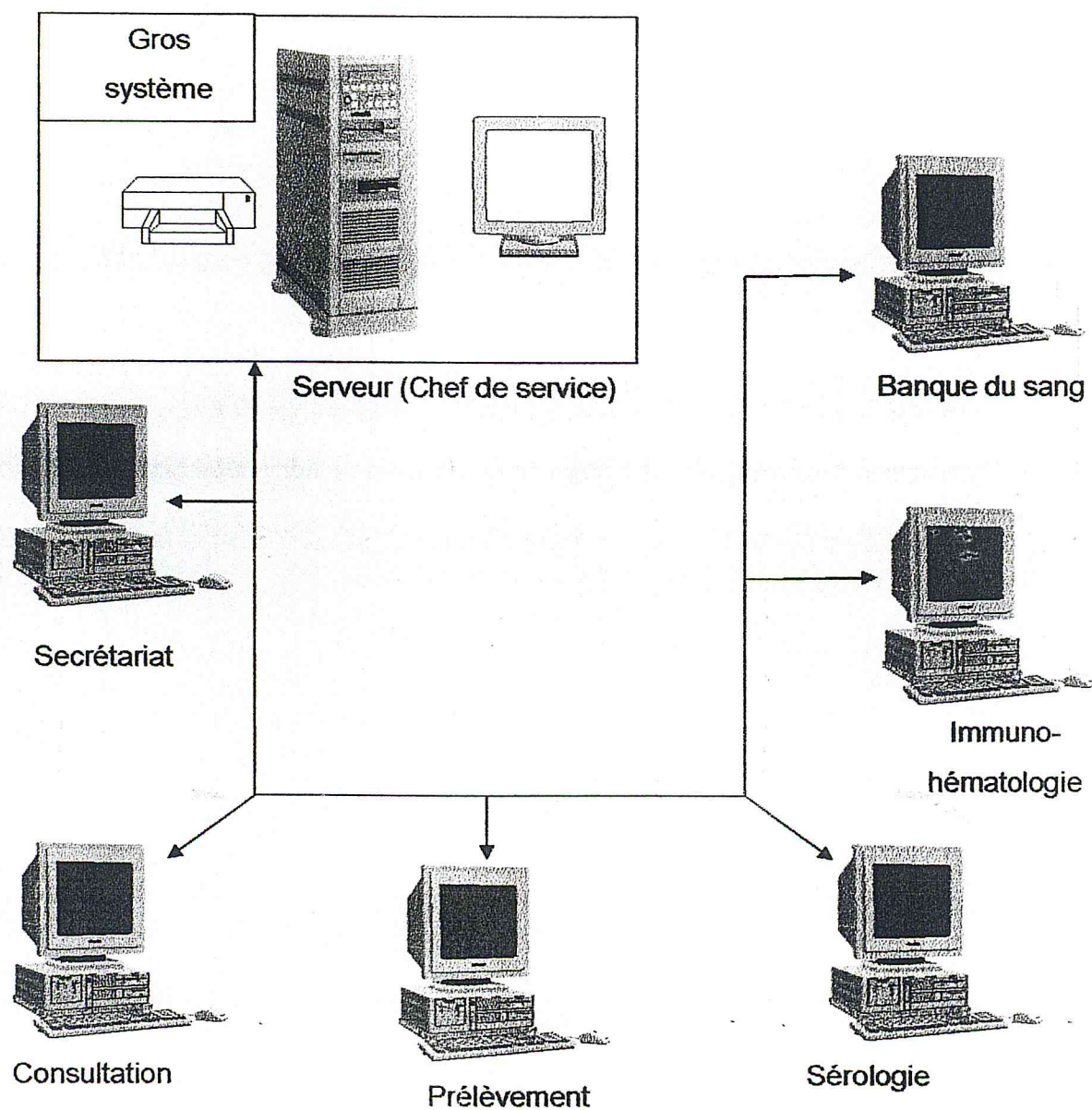


Figure3 : Architecture du réseau client/serveur.

❖ Rôle de chacun des écrans a installé :**➤ Le chef de service :**

- ✓ Saisie des nouveaux médecins internes.
- ✓ Saisie des nouveaux techniciens internes.
- ✓ Supprimer des techniciens internes.
- ✓ Supprimer des médecins internes.

➤ Secrétariat :

- ✓ Saisie des nouveaux donneurs.
- ✓ Saisie des nouveaux patients.
- ✓ Saisie des nouveaux flacons.
- ✓ Supprimer des donneurs.
- ✓ Supprimer des patients.
- ✓ Imprimer les différents documents.

➤ Consultation :

- ✓ Saisie des nouvelles consultations.
- ✓ Consulter les informations des donneurs.
- ✓ Consulter les consultations des donneurs.
- ✓ Consulter les dons des donneurs.

➤ Sérologie :

- ✓ Saisie des nouveaux tests réalisés de sérologie.
- ✓ Saisie des nouveaux les résultats de tests réalisés de sérologie.

➤ Immuno-hématologie :

- ✓ Saisie des nouveaux tests réalisés de groupage des donneurs.
- ✓ Saisie des nouveaux tests réalisés de groupage des patients.
- ✓ Saisie des nouveaux les résultats de tests réalisés des donneurs.
- ✓ Saisie des nouveaux les résultats de tests réalisés des patients.

➤ Banque du sang :

- ✓ Saisie des nouveaux distributions du sang.
- ✓ Saisie des nouveaux demandes du sang.
- ✓ Consulter les flacons du sang.
- ✓ Consulter les demandes du sang.
- ✓ Consulter les distributions du sang.

❖ Avantages et inconvénients de la solution :**➤ Avantages :**

- + Rapidité et efficacité dans l'échange de l'information entre les différentes structures du service.
- + Saisie des informations en temps réel.
- + Les informations seront prises au lieu même de leurs naissance ce qui évite leur encombrement.
- + Les consultations et les mises à jour seront réalisables à tout moment.
- + Amélioration des travaux de recherche et d'enseignement au sein du service.
- + Motivation du personnel.

➤ **Inconvénients :**

- Solution très coûteuse et non réalisable à court terme vue l'indisponibilité du matériel.
- Retour à l'application des procédures manuelles en cas de panne du matériel.
- Une formation du personnel est nécessaire d'où une charge supplémentaire à prendre pour le service.

2^{ème} Solution : multipostes :

La solution est représentée par la configuration suivante :

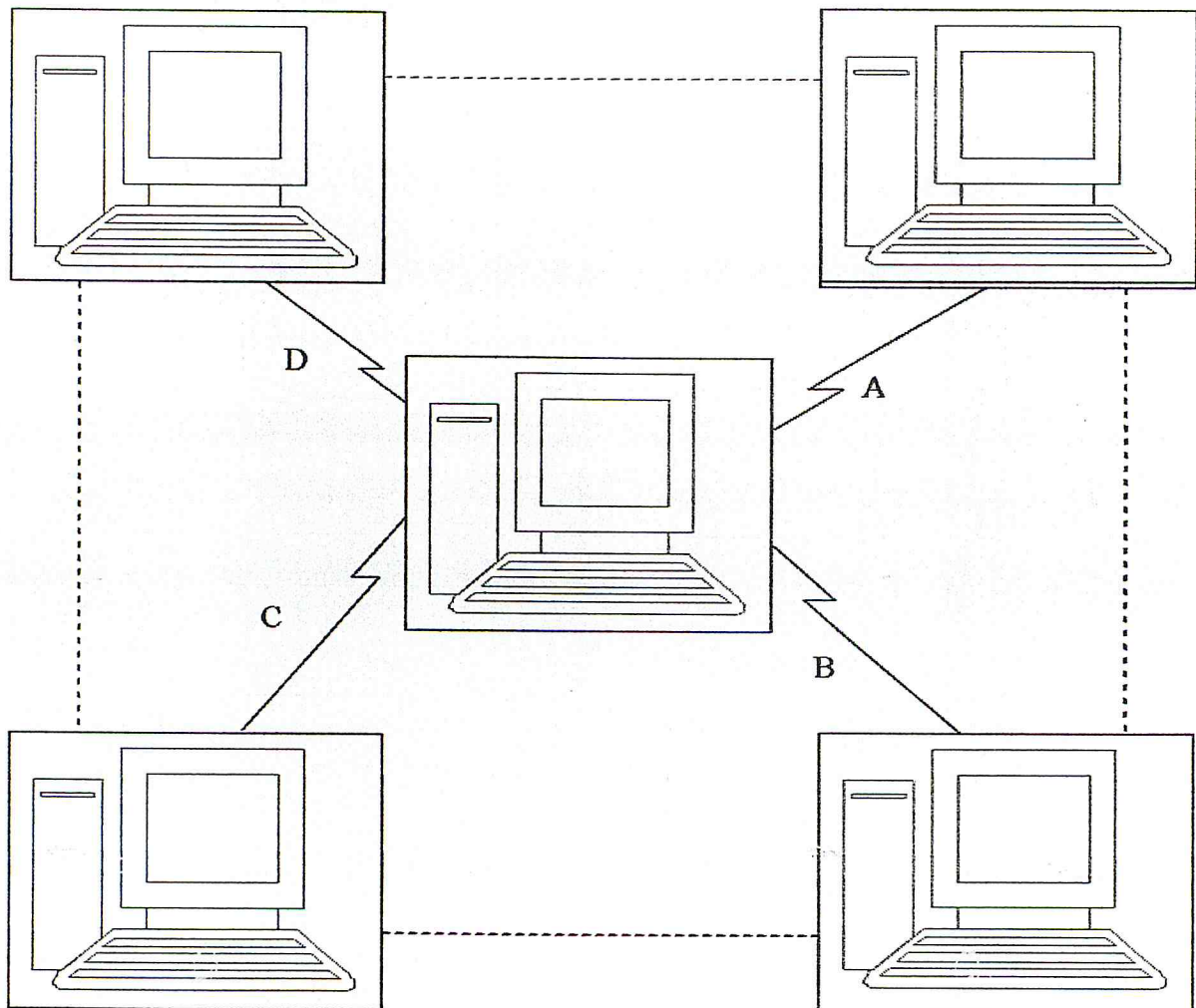


Figure4 : Architecture multipostes.

A, B, C, D Transfert de disquettes.

Répartition des micro-ordinateurs :

Le principe de cette solution consiste à installer un micro-ordinateur avec une imprimante pour chaque service.

Le micro N°1 sera placé au service de secrétariat servira comme point de saisie les donneurs, les dons, les consultations et les flacons.

Le micro N°2 sera placé au service de sérologie servira comme point de saisie les tests de sérologie.

Le micro N°3 sera au niveau de service d'immuno-hématologie servira comme point de saisie les tests de groupage.

Le micro N°4 sera au niveau de service de banque du sang, servira comme point de saisie les distributions du sang et les demandes de sang.

Le micro N°5 sera placé au service de consultation servira comme point de saisie des consultations et les flacons.

Fonctionnement :

Le service de secrétariat et le point de départ de cheminement de l'information. Toutes les informations primordiales seront enregistrées et archivées au même poste (micro N°1), ceci avant chaque consultation.

Les informations concernant les autres services seront saisie dans leurs services.

Chaque fin de journée, on mémorise toutes les informations concernant les tests des flacons sur disquettes qu'on transmettra au service de banque du sang, au service de secrétariat et au service de consultation pour les requêtes de recherche et consultations.

Avantages et inconvénients :**Avantages :**

- ✓ En cas de panne l'arrêt total du système est à écarter.
- ✓ Amélioration des travaux de recherche et d'enseignement.
- ✓ Décentraliser les taches de saisir.
- ✓ Chaque poste effectué son travail indépendamment aux autres.
- ✓ Facilite de consultation.
- ✓ Rapidité de la recherche d'information par apport aux premières solutions.

Inconvénient :

- ✓ Les retards dans la transmission des données peuvent donner naissance à des problèmes concernant les distributions du sang.
- ✓ Le coût élevé.

3^{ème} Solution : mono poste :

Cette solution est représentée par la configuration suivante :

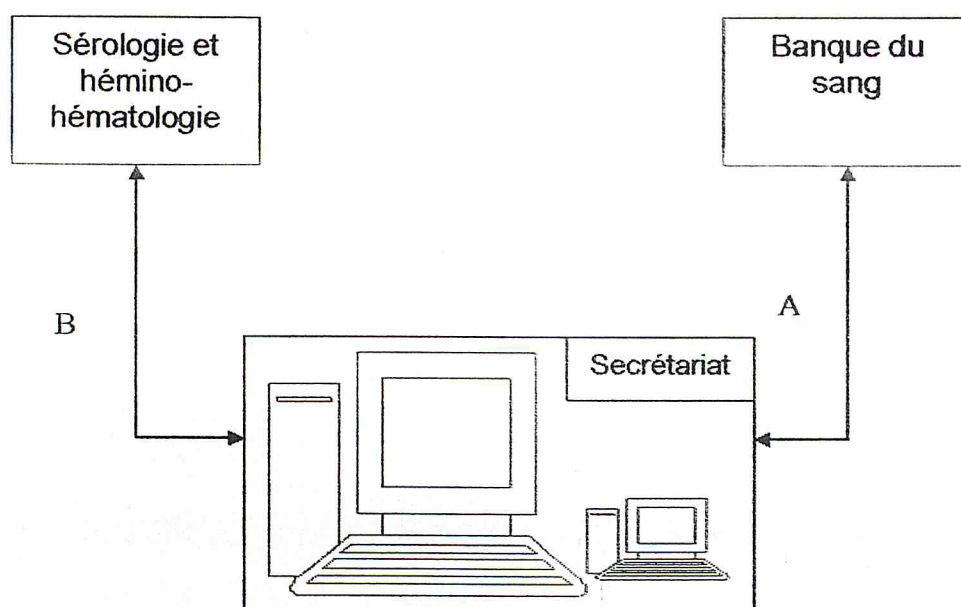


Figure 5 : Architecture mono poste.

A, B transfère des données sur des registres et des documents.

Avantages et inconvénient :**Avantages :**

- ✓ Solution non coûteuse et simple à réaliser.

Inconvénients :

- ✓ Circulation très énorme des registres et documents se qui pose le risque de perte d'informations.
- ✓ Non disponibilité de l'information en temps réel d'où des déplacements fréquents vers le secrétariat seront nécessaires.
- ✓ En cas de panne alors paralysie totale du système.
- ✓ Les tâches de saisir seront centraliser dans un seul service.

Solution retenue :

Après la discussion des différentes solutions avec les responsables de CRTS, et après l'analyse des divers objectifs à atteindre, et surtout les améliorations que la solution choisie devrait apporter à la situation actuelle, nous avons choisi la solution N°1.

9. Conclusion :

A la fin de ce chapitre nous possédons maintenant une certaine connaissance des modalités du fonctionnement de CRTS, cette connaissance nous a permis de pouvoir ressortir les principaux problèmes rencontrés par ces différents utilisateurs, problèmes que nous essayerons de résoudre ou du moins d'atténuer, d'ailleurs cette étude préalable de l'existant va certainement nous aider dans la suite de notre travail et nous permettre d'orienter l'étude que nous menons.

CHAPITRE II

DESCRIPTION DE LA METHODE "OIT"

1. Introduction :

Les systèmes d'informations ont connu des générations successives de méthodes d'analyse. Dans notre étude nous avons optés pour la méthode OMT (Object Modeling Technique, Technique de modélisation par objet), qui est une méthode d'approche orientée objet.

La méthode OMT appartient à la dernière génération des méthodes orientées objet, contrairement aux approches antérieures, OMT permet de couvrir l'ensemble des processus d'analyse et de conception en utilisant le même formalisme. Cette continuité assure une meilleure cohérence entre les différentes étapes du développement. Plus qu'une simple technique de programmation, OMT est essentiellement une façon d'appréhender un problème à l'aide de concepts du monde réel plutôt que de concepts purement informatiques.

La méthode OMT permet de modéliser un système selon trois points de vue complémentaires, chacun capturant des aspects essentiels du système, tous requis pour une description complète :

- le modèle objet (statiques) est le point de vue des données
- le modèle dynamique est le point de vue du contrôle, des comportements
- le modèle fonctionnel est le point de vue des transformations apportées aux données

2. Analyse :

Première étape de la méthode OMT, elle a pour objectif d'examiner les besoins, analyses leur implication et les restituer rigoureusement afin de décrire de manière précise et compréhensible un modèle réel. Ce modèle consiste en un modèle objet, un modèle dynamique et un modèle fonctionnel.

A partir d'un énoncé du besoin, l'analyste construit (avec le demandeur) un modèle du monde réel qui fasse apparaître ses propriétés pertinentes, le modèle issu de l'analyse est une description précise, concise, de ce que le client demande, non de comment on peut le faire les objets du modèle sont des objets du domaine, jamais des objets issus de l'informatique une bonne analyse peut être comprise et critiquée par des experts du domaine qui ne sont pas informaticiens et ne comporte aucune décision d'implantation.

Le modèle d'analyse répond à plusieurs nécessités : il clarifie les besoins, il fournit une base d'accord entre le client et le développeur, il devient le cadre pour la conception et l'implémentation future.

2.1. Le modèle objet :

Ce modèle décrit la structure des objets d'un système :

- leur identité
- leurs relations
- leurs attributs
- leurs opérations

Le modèle objet est représenté par des diagrammes décrivant les différentes classes, hiérarchisées, et comportant les attributs et les opérations utiles.

Le but en construisant un modèle objet est de saisir dans le monde réel les concepts qui sont importants pour une application. Il est représenté graphiquement par des diagrammes d'objets dont les nœuds sont les classes d'objets et les arcs sont les relations entre les classes.

Le modèle objet fournit le cadre majeur dans lequel les modèles dynamique et fonctionnel peuvent être placés.

1.1. L'objet : On définit un objet comme une entité identifiable, ayant des limites connues et précises, et une pertinence particulière pour le problème considéré.

Les objets ont deux vertus :

- ❖ ils permettent la communication
- ❖ ils servent de base pratique à la programmation

Il n'existe pas pour un problème de découpage en objets meilleur que tous les autres, c'est une affaire de jugement

Les objets sont les éléments sur lesquels les autres modèles s'appuient. Le but de la modélisation par objets est de décrire des objets. Un objet est un concept, une abstraction ou une chose ayant des limites très claires et un sens précis dans le contexte du problème étudié, chaque objet a une identité et peut être distingué des autres.

Les objets servent à améliorer la compréhension du monde réel et proposent une base pratique pour l'implémentation informatique.

1.2. La classe : une classe désigne un ensemble d'objets ayant en commun :

- ❖ les attributs
- ❖ le comportement
- ❖ les liens avec d'autres classes
- ❖ la signification ou sémantique

Deux objets distincts d'une même classe peuvent avoir des attributs identiques.

Deux classes distinctes peuvent décrire les mêmes schémas d'attributs et de comportement sans pour autant avoir des sémantiques comparables.

Chaque objet connaît sa classe : la classe est une propriété implicite de l'objet c'est l'utilisation des classes qui permet d'abstraire un problème ce qui peut être décrit au niveau de la classe n'a pas à être répété au niveau des objets instances.

La classe d'objets décrit un groupe d'objets similaires (attributs), un comportement commun (opération), des relations communes avec les autres objets ainsi qu'une même sémantique.

1.3. L'attribut : un attribut est une donnée maintenue par les instances d'une classe, un attribut reçoit une valeur pour chaque instance, un attribut est une valeur d'un type fondamental, pas un objet contrairement aux objets, les attributs n'ont pas d'identité. Le nom d'un attribut peut être suivi par différents descripteurs (à volonté), principalement :

- ❖ le type de la donnée
- ❖ la valeur par défaut

Exemple : nom, âge, poids, marque d'un véhicule.

Les attributs sont répertoriés dans la deuxième partie de la boîte de la classe, chaque nom d'attribut pourra être accompagné de détails facultatifs tels que : le type ou la valeur par défaut.

1.4. Le diagramme d'objet : les diagrammes objets fournissent une notation graphique pour les concepts de classes et d'objets, on dispose de deux types de diagrammes :

- ❖ les diagrammes de classes : les diagrammes de classes décrivent des ensembles d'objets, leurs caractéristiques et les liens inter classes.

- ❖ les diagrammes d'instances : les diagrammes d'instances décrivent des objets effectifs avec leurs liens, ils sont utiles pour documenter des jeux de tests (scénarios), et pour discuter des exemples on utilise principalement les diagrammes de classe. Les diagrammes d'instances servent à documenter des diagrammes de classe complexes.

1.5. Les opérations et les méthodes : une opération est une transformation qui peut être appliquée à ou par des objets d'une classe, une opération possède comme paramètre implicite l'objet pour lequel elle est invoquée, une opération utilise et/ou modifie des attributs de l'objet, une opération peut être définie pour plusieurs classes (polymorphisme), l'implantation effective d'une opération pour une classe s'appelle une méthode (ou encore fonction membre).

Toutes les méthodes qui implantent une opération doivent avoir :

- ❖ le même rôle symbolique.
- ❖ la même signature.
- ❖ le même type de valeur de retour.

La même opération peut s'appliquer à de nombreuses classes différentes. Une méthode est l'implémentation d'une opération pour une classe. Les opérations sont énumérées dans la troisième partie de la boîte de la classe.

La figure ci-dessous résume la notation de modélisation par objet pour les classes.

Nom de classe
Nom d'attribut 1 : type de donnée 1 = valeur par défaut 1. Nom d'attribut i : type de donnée i = valeur par défaut i.
Nom d'opération 1(liste d'arguments 1) : type de résultat 1. Nom d'opération j (liste d'arguments j) : type de résultat j.

Figure 6 : Notation des classes

1.6. Les liens et les associations : les liens et associations permettent de définir des relations entre objets.

❖ **les liens :** un lien est une connexion physique entre instances d'objets. Un lien est une instance d'association.

Exemple : James Bond *travaille pour* l'intelligence service.

Mathématiquement, un lien est un n-uple : une liste ordonnée d'objets.

❖ **les associations :** une association est un groupe de liens avec une structure commune et une sémantique identique.

Exemple : un agent secret *travaille pour* un service secret.

Mathématiquement, une association est une relation. Une association est bi-directionnelle la direction induite par le nom est dite direction normale, l'autre direction est dite direction inverse l'association inverse possède parfois un nom, mais ce n'est pas nécessaire.

Exemple : un service secret *emploie* un agent secret.

Les associations peuvent être binaires, ternaires ou d'autre ordre.

Le symbole OMT pour les associations ternaires et n-aires est un losange avec des lignes le reliant aux classes concernées par l'association.

Le nom de l'association est optionnel et est écrit près du losange.

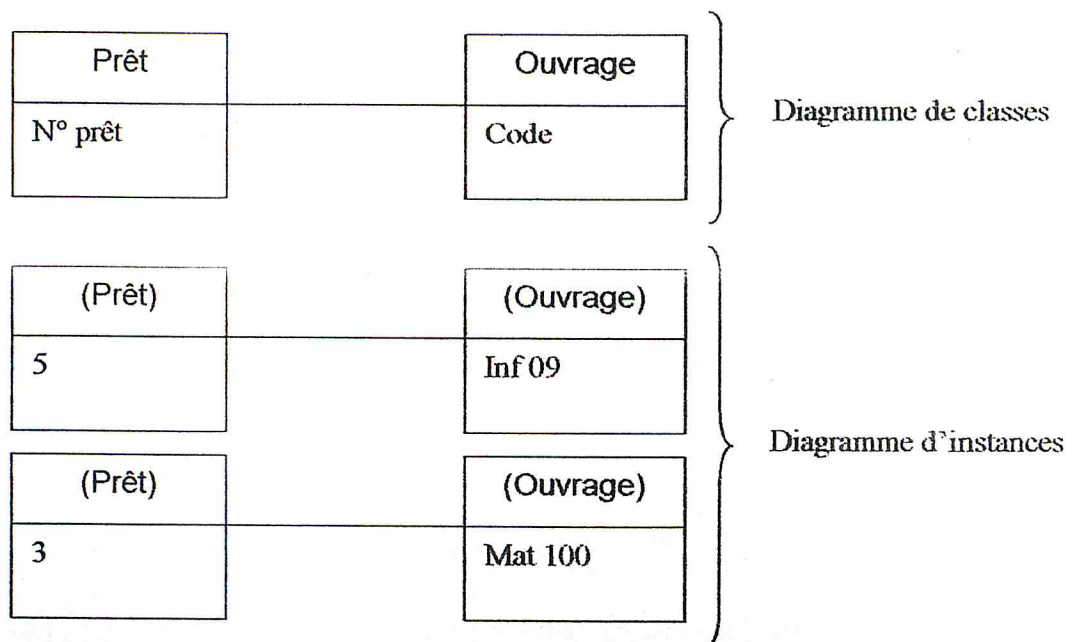


Figure 7 : Association un à un et lien.

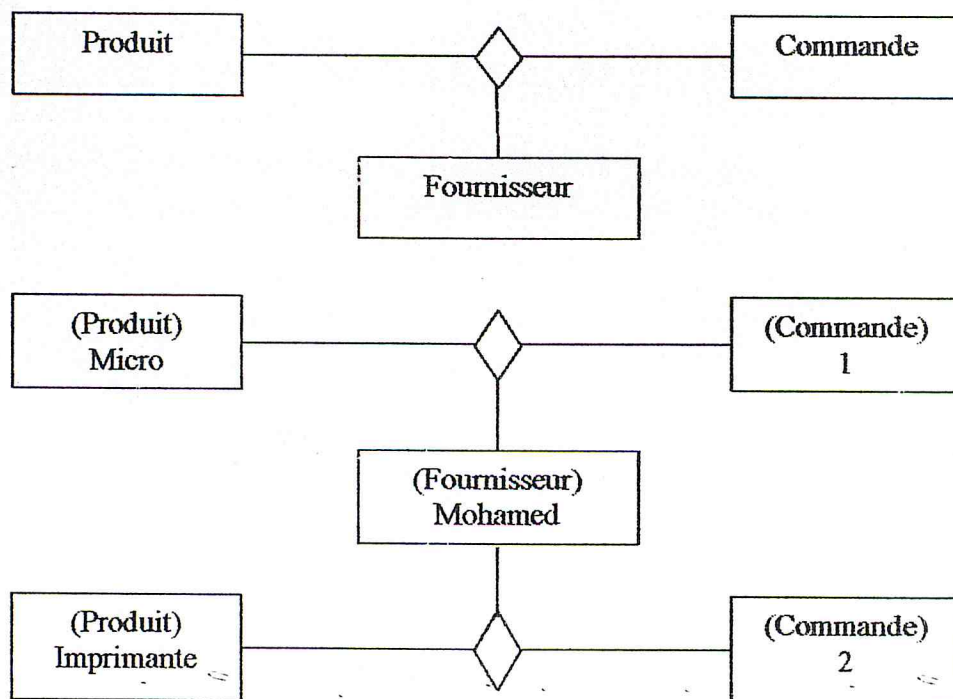


Figure 8 : Associations ternaires et liens

1.7. Multiplicité : la multiplicité désigne le nombre d'instances d'une classe qui peuvent être liées à une instance d'une classe associée, la multiplicité est souvent d'ordre "1", ou d'ordre "plusieurs", mais en fait toutes les situations sont possibles.

- ❖ un rond noir signifie "zéro ou plus".
- ❖ un rond blanc signifie "zéro ou un".
- ❖ la multiplicité peut toujours être précisée : 2+ à côté d'un rond noir signifie "2 ou plus".
- ❖ la multiplicité est écrite du côté de la classe qui porte la multiplicité, c'est à dire dont les instances sont comptées.

Exemple : une station de travail affiche zéro ou une fenêtre "console".



Figure 9 : Exemple d'associations zéro ou un

❖ **Choix des multiplicités :** la détermination précise des multiplicités se fait en général lorsque le modèle est assez avancé et bien compris, le choix des multiplicités dépend souvent de conditions importantes sur le domaine.

➤ Sous estimer la multiplicité peut conduire à une mauvaise flexibilité de l'application.

Exemple : une personne a-t-elle un ou plusieurs numéro de téléphone?

➤ Sur estimer la multiplicité conduit à des coûts inutiles.

1.8. Concepts évolués de liens et d'associations :

❖ **Attributs de liens :** un attribut de lien est une propriété des liens d'une association.

Exemple : "droit d'accès" est une propriété du lien "accessible par" entre un utilisateur et un fichier.

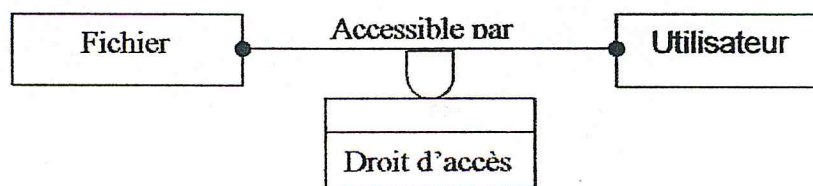


Figure 10 : Attributs de lien pour association à plusieurs.

❖ **Modéliser une association comme une classe :** il est parfois utile de modéliser une association comme une classe :

- quand un lien peut participer à des associations avec d'autres classes.
- quand un lien peut comporter des opérations.

Dans ce cas, chaque lien devient une instance de la classe. (Un lanceur peut jouer pour plusieurs équipes la même année)

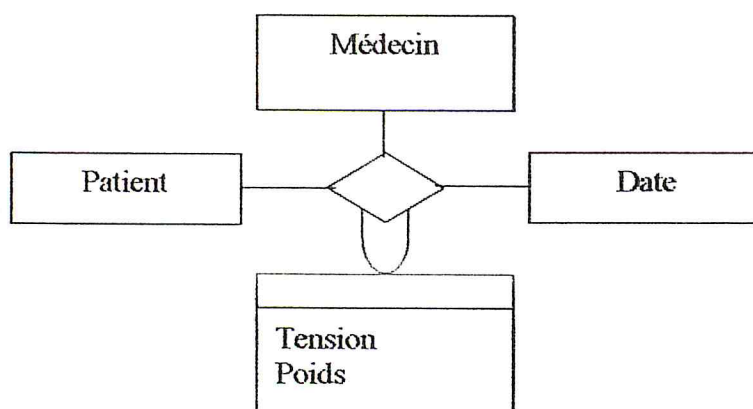


Figure 11 : Modéliser l'association « autorisation » comme une classe

➤ **Le nom de rôle** : un rôle est une extrémité d'association, une association binaire a deux rôles, et chacun peut recevoir un nom, un rôle permet de voir une association comme un moyen de naviguer d'un objet à un ensemble d'objets liés, le nom de rôle est le nom que l'on donnerait au champ de la classe opposée qui pourrait implanter le lien.

Un nom de rôle est un attribut dérivé de la classe opposée dont la valeur est un ensemble d'objets liés (et donc accessibles) parfois le nom de la classe est suffisant et permet l'omission des noms de rôles, notamment lorsque deux classes ne participent ensemble qu'à une seule association.

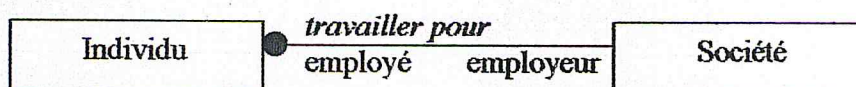


Figure 12 : Exemple de nom de rôle

Les noms de rôles sont nécessaires pour des associations liant des objets d'une même classe, les noms de rôles au bout des associations attachées à une classe doivent être uniques, ils ne peuvent non plus être en conflit avec des noms d'attributs.

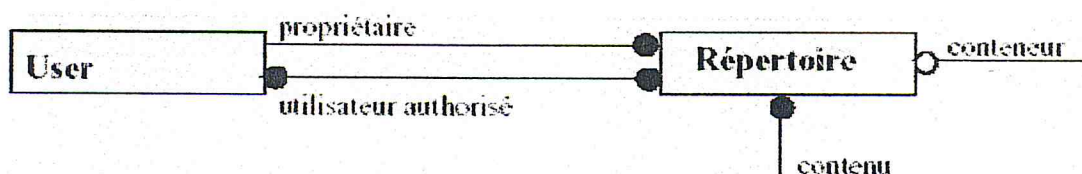


Figure 13 : Exemple des noms de rôles dans une association attachée à même classe

➤ **L'ordre** : Habituellement les objets du côté « plusieurs » d'une association n'ont pas d'ordre explicite et peuvent être considérés comme des ensembles. Parfois, ils sont explicitement ordonnés.

Un ensemble ordonné d'objets du côté « plusieurs » de l'association est indiquée par {ordonné} près du point de multiplicité de rôle. C'est un type particulier de contraintes.

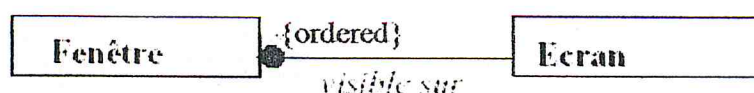


Figure 14 : Une association ordonné.

➤ **Qualification** : une association qualifiée fait intervenir deux classes et un qualifieur, le qualifieur est un attribut particulier qui permet de réduire la multiplicité d'une association les associations 1-n et m-n peuvent être qualifiées, le qualifieur distingue parmi les objets d'un côté "n", une association qualifiée peut être vue comme une forme d'association ternaire.

Exemple : un répertoire contient de nombreux fichiers, distingués par leur nom

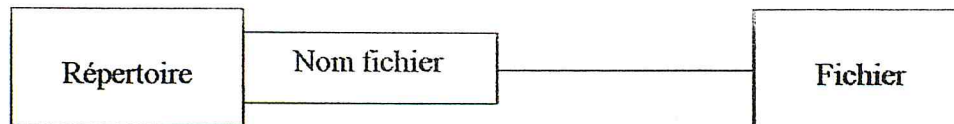


Figure 15 : Une association qualifiée.

La qualification ne réduit pas toujours à l'unité. Souvent, la qualification réduit à l'unité, c'est à dire que le qualifieur identifie exactement un objet ce n'est pas toujours le cas.

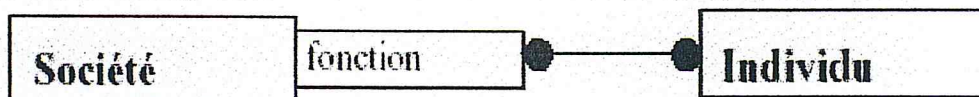


Figure 16 : Une association qualifiée sans réduire la multiplicité.

Une même personne peut être directeur de plusieurs sociétés. Une société peut avoir plusieurs directeurs. En connaissant la fonction dans la société, on réduit cependant le nombre des individus éligibles pour le lien.

Le qualifieur peut parfois être vu comme un attribut. On peut parfois modéliser un qualifieur par un attribut.

Exemple : les sociétés sont identifiées sur chaque place boursière par une abréviation télex, toujours la même.

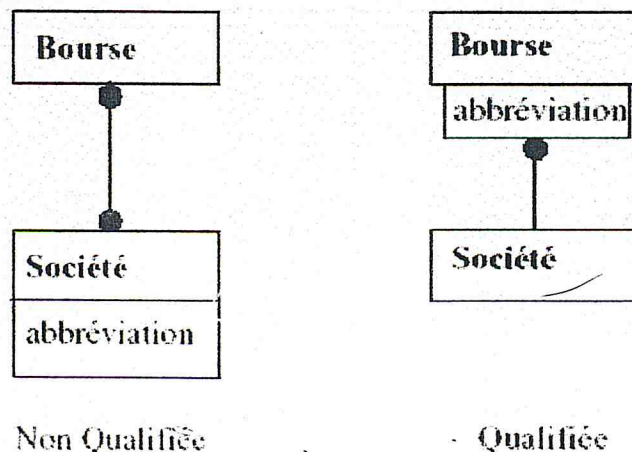


Figure 17 : L'association qualifiée est préférable de l'autre .

La version qualifiée, car elle résisterait au fait qu'une même société possède des abréviations télex différentes sur des places boursières distinctes.

➤ **Agrégation** : l'agrégation est la relation de la partie au tout, des composants à l'assemblage.

Exemples : Un document consiste en plusieurs paragraphes, chacun consistant en plusieurs phrases.

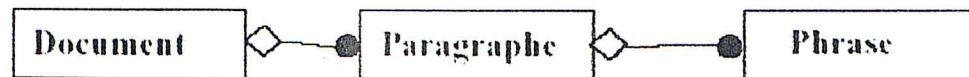


Figure18 : Exemple d'agrégation.

Une relation d'agrégation lie une classe assemblage à une classe composant. Un assemblage avec plusieurs composants correspond à plusieurs relations d'agrégation. On peut décrire la multiplicité des composants dans l'assemblage.

L'agrégation est une forme d'association finement couplée avec une sémantique additionnelle. L'agrégation est dessinée comme une association à l'exception de petit losange qui indique l'extrémité assemblage de la relation.

Une agrégation peut comporter plusieurs niveaux.

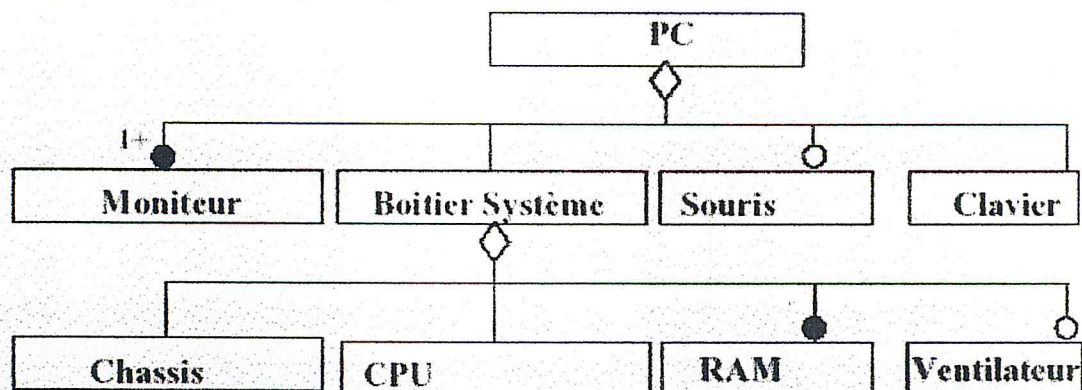


Figure 19 : Exemple d'agrégation à plusieurs niveaux.

1.8. Héritage et généralisation :

- **Héritage** : héritage et généralisation sont utiles pour partager des concepts entre plusieurs classes tout en préservant les différences. Chaque sous-classe hérite les mêmes propriétés de sa super-classe et possède ses propres attributs et opérations spécifiques.

Les attributs et opérations décrits pour la super classe sont hérités par les sous classes. L'héritage est le mécanisme par lequel des caractéristiques décrites pour une super classe sont rendues disponibles dans une sous classe.

- **Généralisation** : la généralisation est la relation entre une classe et une ou plusieurs versions raffinées de la même classes. La classe précisée est la super-classe et les autres sont des sous classes.

La notation pour la généralisation est triangle reliant une super-classe et est relié par une ligne ou sommet de triangle.

La relation de généralisation est souvent appelée relation « est un », ou bien « est une sorte de » (en anglais « is a »).

Les sous-classes sont reliées par des lignes à une barre horizontale. Elle-même reliées à la base de triangle.

- **Propriétés d'héritage et de généralisation** :

- ❖ héritage et généralisation sont transitives.
- ❖ on utilise les termes de (classe) **ancêtre** et de (classe) **descendante** pour dénoter la fermeture transitive de la généralisation.
- ❖ une instance de sous classe est aussi instance de toutes ses classes ancêtre.

- ❖ une instance d'une sous classe comporte des données pour tous les champs de ses ancêtres.
- ❖ toute opération ancêtre (non privée) peut être appliquée à une classe.

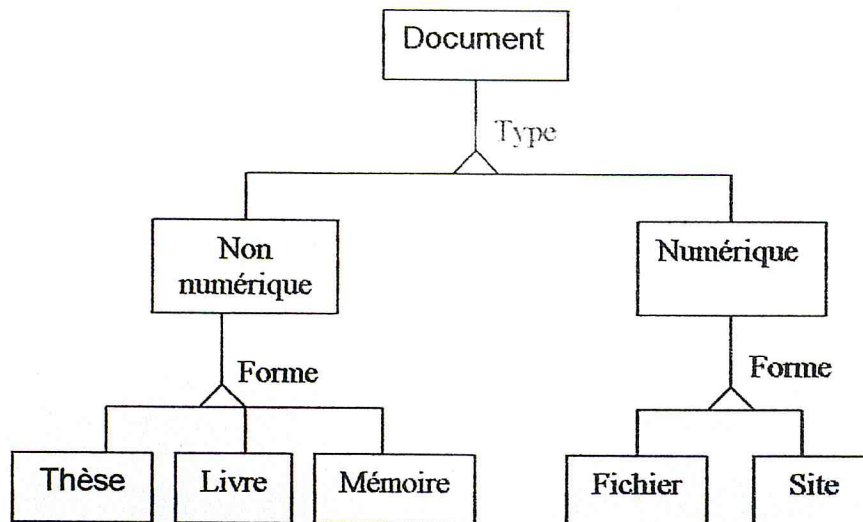


Figure 20 : Héritage et généralisation en plusieurs niveaux.

➤ **Discriminateurs :** un discriminateur est un attribut de type énuméré qui indique quelle propriété est abstraite par la relation de généralisation. Les mentions « type de » sont des discriminateurs. Seulement une propriété peut être généralisée par niveau.

Exemple : la classe véhicule peut être discriminée de deux façons :

- ❖ par le mode de propulsion (humaine, à cheval, à charbon, diesel ...).
- ❖ par le milieu (eau, air, terre, espace ...).

Le discriminateur n'est qu'un nom donné à la généralisation. Les valeurs de discriminateurs sont dans une correspondance bijective (un-un) avec les sous classes.

Les discriminateurs ne sont pas forcément implantés dans les programmes.

Les discriminateurs ne sont pas impératifs dans les diagrammes objet.

1.10. Les classes abstraits : une classe abstraite ne possède pas d'instances, mais décrit des attributs et opérations communs à plusieurs sous classes. Une classe concrète peut avoir des sous classes abstraites.

La classe « Objet » dans un éditeur graphique est typiquement abstraite toutes les opérations non définies, mais simplement nommées dans une classe abstraite doivent recevoir une implémentation dans les sous classes concrètes.

Une classe abstraite décrit (une partie de) l'interface de programmation de ses sous classes concrètes sans être capable de produire une implantation effective de la plupart des opérations.

Une classe concrète peut généralement être affiné en plusieurs sous classes, ce qui la rend abstraite.

En contre partie une classe abstraite peut devenir concrète dans une application ou la différence entre ces sous classes est sans importance.

Il faut noter que la nature abstraite d'une classe est toujours provisoire et dépend du point de vue pris en compte.

1.11. Les contraintes : une contrainte est une condition logique qui s'applique à des entités du modèle : objets, classes, attributs, liens, associations.

Une contrainte limite les valeurs que peuvent prendre certains attributs, soit à tout moment (contrainte universelle), soit au cours du temps (contrainte dynamique) les contraintes sont exprimées de façon déclarative de préférence un bon modèle décrit de nombreuses contraintes.

Dans le model objet les contraintes sont délimités par des accolades et positionnées près de l'entité concernée par la contrainte.

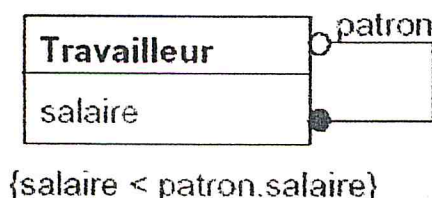


Figure 21 : Exemple de contraintes

1.11. Les contraintes sur les liens : la multiplicité contraint une association donc elle restreint le nombre d'objets donnée.

La notation « {ordonné} » indique que les éléments de l'extrémité d'une association ont un ordre explicite qui doit être préservé.

Les contraintes générales lient plusieurs classes ou associations. Ce lien est mentionné par des flèches pointillées assorties d'un énoncé commenté de la contrainte, elles doivent être exprimées sous forme naturelle ou en équations (meilleur) lorsque trop de classes sont liées par la contrainte, il vaut mieux la décrire à part que de ruiner le modèle.

1.12. Clefs minimales - candidate keys :

Une clef minimale ("candidate key" en anglais) est un ensemble minimal d'attributs qui identifient uniquement un objet ou un lien.

Une classe ou une association peut avoir plusieurs clefs minimales, l'identifiant d'un objet est toujours une clef minimale.

Les clefs minimales sont utiles pour expliciter les associations ternaires et plus : savoir si on doit disposer de deux ou trois objets pour identifier le lien est une information très importante.

Exemple : {candidate key : (Projet),(Individu),(Langage)}

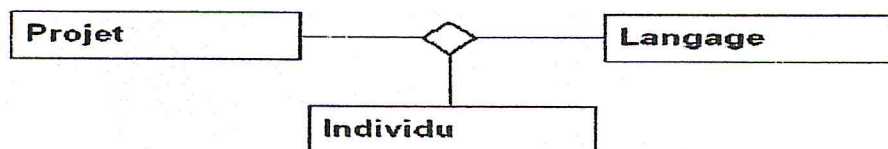


Figure 22 : Exemple de clefs minimales dans une association ternaire.

1.13 : Grouper les constructions :

➤ **Les modules :** le module est une unité logique qui groupe des classes, associations et généralisations, c'est une vue du problème (exemple : électricité, plomberie ...). un modèle objet consiste en un ou plusieurs modules, les noms de classes et d'associations doivent être uniques dans un module, il est préférable d'avoir des conventions de nommage uniformes et cohérentes entre les modules. le nom du module est habituellement mentionné en haut de chaque feuille mentionner la même classe dans plusieurs modules est le mécanisme par lequel on lie les modules entre eux

raisonnablement, les liens externes entre modules doivent être bien moins nombreux que les liens internes.

➤ **Les feuilles** : la feuille permet de découper le modèle objet complet en une série de pages chaque feuille consiste en une unique page un module consiste de une ou plusieurs pages. Mais jamais plusieurs modules sur une page.

Chaque feuille possède un titre et un numéro. Chaque association ou généralisation apparaît sur une seule feuille.

Une classe peut apparaître sur plusieurs feuilles, créant de ce fait le lien entre les feuilles on peut (non obligatoire) mentionner dans un cercle auprès d'une classe les nom et/ou numéros des feuilles qui complètent la description de la classe.

2.2. Le modèle dynamique :

Le modèle dynamique décrit les aspects du système qui se modifient avec le temps, il décrit la structure de contrôle des objets.

Le contrôle est l'aspect de système qui décrit des séquences d'opérations activées par des stimuli externes sans tenir compte de l'activité de ces opérations, de leurs champs d'exécution ou de leurs mode d'implémentation.

Le modèle dynamique est représenté graphiquement par un diagramme d'état dont les noeuds sont les états et les arcs sont les transitions entre les classes provoquées par des événements.

1. L'événement : un événement se produit à un instant donné dans le temps, et ne possède pas de durée. Les événements sont séquencés dans le temps, deux événements sans lien de causalité sont dits concurrents, ou parallèles (sans qu'ils doivent nécessairement se produire simultanément).

Chaque événement est unique, mais on décrit des classes d'événements, munies d'une organisation hiérarchique, certains événements sont de simples signaux, mais d'autres portent des informations transmises d'un objet à un autre, les informations associées à un événement sont des attributs de sa classe.

Exemples :

- ❖ décollage d'avion : avion, n° de vol, ville.
- ❖ appui sur la souris (n° de bouton, coordonnées).
- ❖ on décroche le téléphone.

2. Scénarios et traces d'événements : un scénario est une séquence réelle d'événements qui se produit pendant une exécution du système, les scénarios sont comparables aux instances dans leur relation aux classes d'objets : des instances.

Exemple : les événements successifs lors d'une communication téléphonique.

L'étape qui suit le listing d'un scénario est l'identification des objets impliqués, une trace associe les événements et les objets source et destination des informations associées dans le cas du téléphone on aurait :

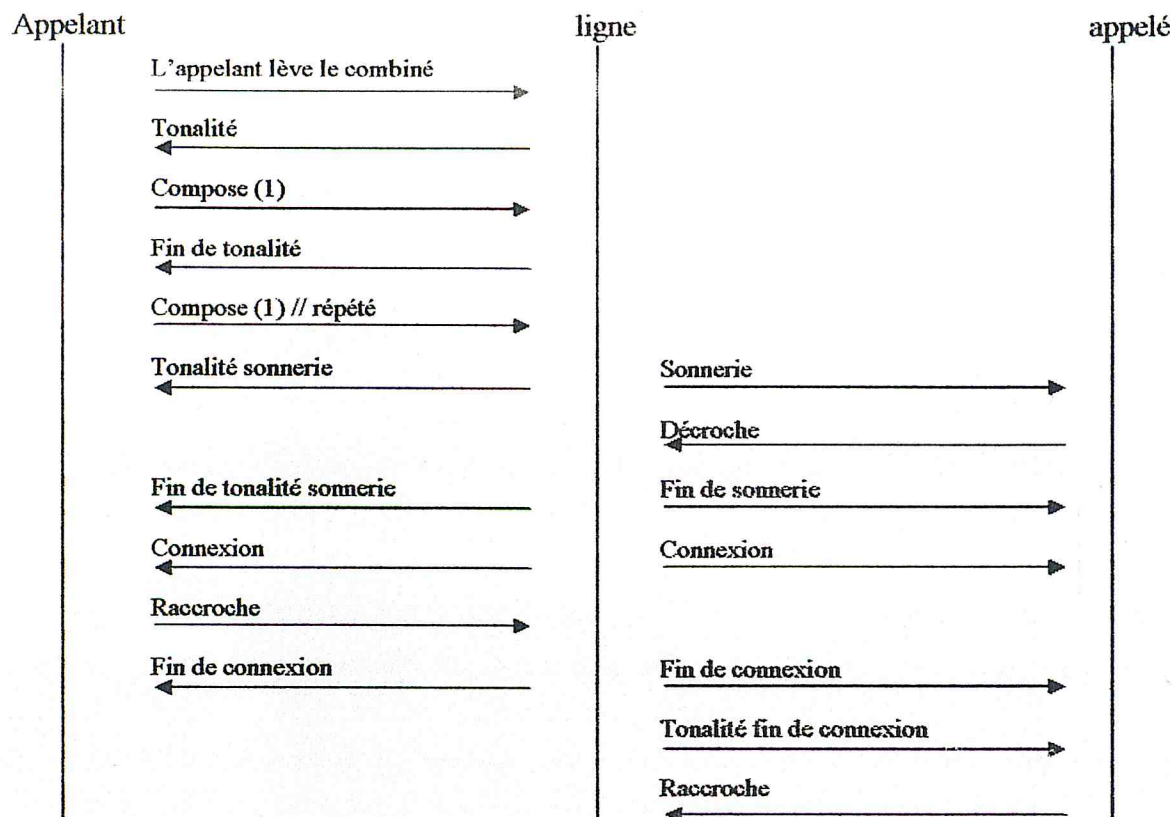


Figure 23 : Scénario de téléphone.

3. Etats : un état est une abstraction des valeurs d'attributs et de liens d'un objet et décrit un ensemble de valeurs. Un état est stable entre deux événements reçus par un objet et possède une durée l'ensemble des valeurs d'un attribut correspondant à un état est souvent décrit en intention par une condition logique ou une contrainte.

La description des événements et états dépend du niveau d'abstraction auquel on se place : un seul événement dans un modèle peut être décomposé en de nombreux événements et états dans un modèle plus précis.

La description des états ignore les attributs non pertinents.

Exemple : l'alarme sonne

4. Diagrammes d'états : un diagramme d'état associe des événements et des états quand un événement est reçu, l'état suivant dépend de l'état courant et de l'événement. Un changement d'état du fait d'un événement est appelé une transition. Un diagramme d'état est un graphe dont les noeuds sont des états et les arcs des transitions marquées par leur événement associé. Quand les conditions d'une transition sont réunies, on dit que celle ci "déclenche".

Un diagramme d'états décrit le comportement d'une seule classe d'objets, chaque instance obéit au même diagramme, mais se trouve dans un état "personnel", un diagramme d'états peut représenter une boucle, ou bien un traitement unique.

Les diagrammes de traitement unique possèdent un état initial et un état final de tels diagrammes peuvent être vus comme des sous programmes pour un diagramme complet.

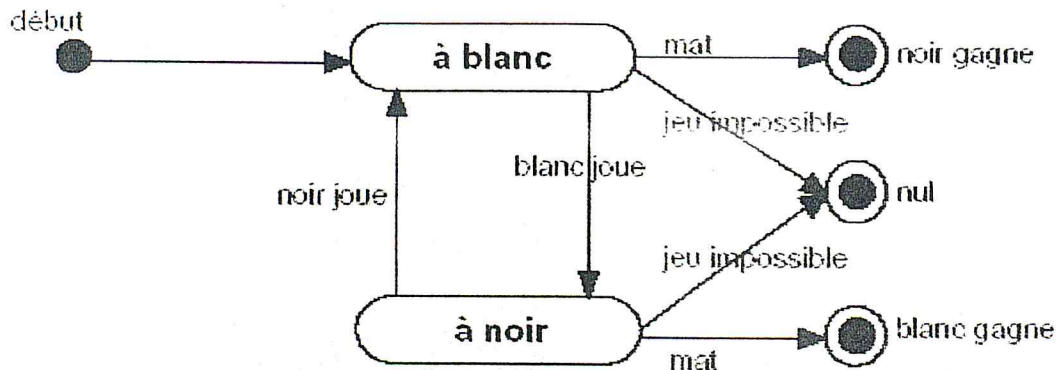


Figure 24 : Exemple d'un diagramme de traitement unique

5. Conditions : une condition est une fonction booléenne de l'état d'un objet, valide sur un intervalle de temps. Un état peut être défini par une condition être dans un état est une condition. Les conditions peuvent être utilisées pour garder des transitions, qui ne peuvent être déclenchées que si la condition est satisfaite.

6. Opérations :

➤ **contrôle des opérations :** le diagramme d'état comme tel est de peu d'utilité s'il ne permet pas de décrire comment les objets réagissent aux événements.

- ❖ Les opérations attachées aux états et aux transitions sont effectuées en réponse aux stimulations associées.
- ❖ Une activité est une opération qui "dure" un certain temps, et est associée à un état on compte comme activités les opérations continues : fonctionnement d'un moteur par exemple et les opérations non instantanées (exécution d'un programme, déplacement du bras d'un robot).
- ❖ La notation "do: A" dans une boîte d'état traduit l'exécution d'une activité A associée à l'état, dès l'instant d'entrée dans l'état.
- ❖ Une action est une opération sans durée, associée à une transition, ou à l'entrée dans un état traduire une activité par une action est légitime dès lorsque l'on ne s'intéresse pas à la structure de contrôle de l'opération, Sinon c'est impossible.
- ❖ La notation /A est utilisée pour associer un action A à une transition.

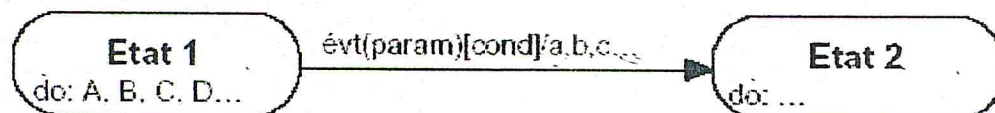


Figure 25 : résumé de la notation de diagramme d'état.

7. Relation entre les modèles objet et dynamique : un modèle dynamique décrit tout ou partie du comportement des objets d'une classe donnée.

- ❖ les états sont des classes d'équivalence d'attributs et de valeurs de liens pour des objets.
- ❖ les événements sont des opérations du modèle objet.

- ❖ chaque sous état d'un état contraint plus les valeurs des attributs, ainsi que le fait la restriction.
- ❖ une hiérarchie d'états est équivalente à une hiérarchie de restriction pour la classe
- ❖ les modèles et langages orientés objet ne permettent pas habituellement de décrire la généralisation par restriction dans l'héritage, d'où l'intérêt d'un modèle dynamique.
- ❖ héritage de restriction et héritage d'états sont deux points de vue sur la même chose : l'un statique (un objet ne peut pas changer de classe), et l'autre dynamique (il peut changer d'état dans sa classe).
- ❖ le modèle dynamique d'une classe est hérité par les sous classes, qui possèdent les mêmes transitions et les mêmes états.
- ❖ la hiérarchie des événements est indépendante de celle des classes en pratique.
- ❖ les événements sont fondamentaux, partagés par plusieurs classes en général.
- ❖ chaque état est défini indirectement par les objets et les événements.
- ❖ les transitions sont souvent implantées par des transitions sur les objets, de même nom.

2.3. Le modèle fonctionnel :

Le modèle fonctionnel décrit les calculs effectués par un système. Un calcul est une transformation de données. On décrit les résultats des transformations des données, sans dire quand ces transformations se produisent ce modèle décrit la signification - le rôle - des opérations et des actions des modèles objet et dynamique. Le modèle fonctionnel est central pour certains systèmes (compilateurs, feuilles de calcul électronique, le code des impôts).

1. Les diagrammes de flot de données : un diagramme de flots de données (DFD) décrit les relations fonctionnelles qui existent entre, les différentes valeurs calculées par un système.

Les diagrammes à flot de données sont particulièrement utiles pour représenter la fonction de plus haut niveau d'un système et son éclatement en unités fonctionnelles plus petites. Un traitement peut être développé en un nouveau diagramme à flot de données, chaque entrée ou sortie du nouveau diagramme.

Un DFD est un graphe qui relie essentiellement des sources de données et des processus. Normalement, un tel graphe ne mentionne pas le flot de contrôle.

Les éléments d'un DFD sont :

- ❖ Des process : qui transforment des données.
- ❖ Des flots de données : qui déplacent des données.
- ❖ Des acteurs : qui produisent et consomment des objets.
- ❖ Des mémoires (banques de données) : qui stockent et rendent disponibles des informations.

Imbriquer un diagramme à flot de données permet à chaque niveau d'être cohérent et compréhensible et pourtant la fonctionnalité globale peut être d'une complexité quelconque.

2. Les process : un process est une opération qui transforme des données, ceux de plus bas niveau correspondent tout simplement à des fonctions sans effets de bord.

Un DFD complet peut être vu comme un process de haut niveau : par exemple, le DFD précédent devient :



Figure 26 : Un DFD se présente comme un process.

Le diagramme montre les différents chemins suivis par les données, sans préjuger de celui qui sera effectivement utilisé. Un process a plusieurs entrées, et souvent plusieurs sorties :

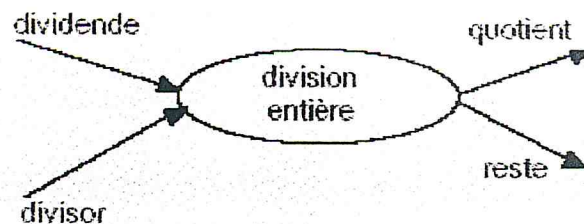


Figure 27 : Les chemins des données.

Un diagramme plus précis décrit ce qui est à l'œuvre dans un process de haut niveau.

3. Les flots de données : un flot relie une sortie d'un objet ou process ou mémoire à l'entrée d'un autre de ces éléments, il représente une donnée inchangée dans son transport, la même valeur peut être transmise à plusieurs consommateurs.

Une donnée peut être décomposée en éléments (attributs).

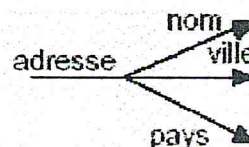


Figure 28 : Décomposition d'une donnée en éléments.

4. Les acteurs : un acteur est un objet actif qui produit ou consomme des valeurs de tels objets sont liés aux entrées et aux sorties d'un DFD. On les appelle aussi terminateurs, du fait de leur position périphérique.

Les opérations des acteurs sont sans objet pour le DFD, mais doivent être décrites par les diagrammes dynamiques (recevoir une donnée en "entrée" est forcément traduit en un événement). Un acteur est représenté avec la notation usuelle pour les classes : c'est un objet.

5. Les banques (réservoirs) de données : une banque de donnée (data store) est un objet passif, qui stocke des données ou des objets pour une utilisation future. Contrairement aux acteurs, elles n'implémentent pas d'autres opérations que celles de stocker et rendre des informations.

Le statut habituel d'une base de données est une banque de données dans un DFD, une banque de données peut restituer un élément isolé mémorisé ou bien l'intégralité de son contenu. Les banques de données sont des objets du point de vue OMT.

A la différence d'un acteur, une banque de données n'engendre pas d'opération pour lui-même mais répond simplement à des requêtes pour stocker les données et y accéder, une banque de données peut être implémenté comme un fichier et un acteur comme un dispositif externe.

On représente une banque de données par une paire de lignes parallèles contenant le nom de la banque de données.

Les flèches sortantes indiquent les informations récupérées à partir du réservoir. Il peut s'agir de la récupération de la valeur entière ou d'un de ses composants.

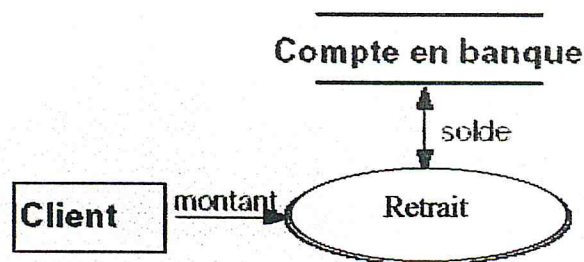


Figure 29 : Retrouver et stocker par le même process dans un réservoir.

6. Le flot de contrôle : le flot de contrôle est une valeur booléenne qui permet de savoir si un traitement est évalué ou non, il est compris comme une donnée booléenne (en vrai/faux) qui intervient en entrée et/ou en sortie de processus, et représenté par une flèche pointillée.

Un diagramme à flots de données montre tous les chemins possibles de calcul de valeurs et ne montre pas quels chemins sont exécutés ni dans quel ordre.

La décision et séquences sont des problèmes de contrôle et en tant que tels font partie du modèle dynamique.

Une décision permet de savoir si une ou plusieurs fonctions sont exécutées plutôt que de fournir une valeur aux fonctions. Même si les fonctions n'ont pas de valeurs d'entrées à partir de ces fonctions de décision, il est parfois utile de les englober dans le modèle fonctionnel afin de ne pas les oublier et de faire apparaître leurs dépendances de données.

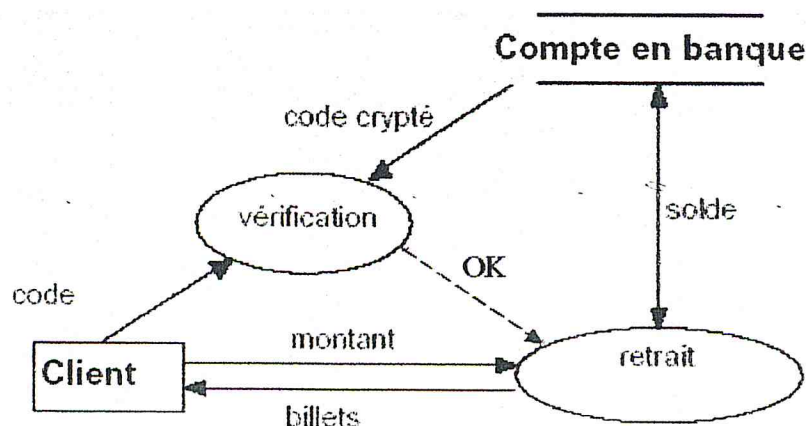


Figure 30 : la flèche (OK) est un flot de contrôle.

7. La spécification d'opérations :

Spécifier une opération requiert de définir sa signature (en conformité avec ce qui est connu par le modèle objet) et la transformation qu'elle opère sur ses arguments, dont l'objet cible lui-même ainsi que la donnée qu'elle retourne.

La spécification externe d'une opération ne nécessite pas de décrire des changements temporaires à son exécution.

Tout processus de bas niveau est une opération.

Les opérations sont triviales entrent dans trois catégories :

- les requêtes (ou accesseurs)
- les actions
- les activités

➤ **Les requêtes** : une requête ne possède pas d'effet de bord. Une requête sans paramètre est un attribut dérivé.

➤ **Les actions** : une action possède des effets de bord, et n'a pas de durée. Elle procède par des modifications d'attributs et de liens sur l'objet lui-même ou d'autres. Une action peut être définie en termes de l'état du système avant et après son exécution, de façon formelle ou non, mais en tout cas sans ambiguïté. On peut aussi décrire son algorithme, par exemple en pseudo code.

➤ **Les activités** : une activité possède une durée, "donc" des effets de bord, elles ne sont pertinentes que pour les objets de type "acteur", et pas pour les mémoires ou banques de données.

Les activités sont spécifiées par le modèle dynamique et par le modèle fonctionnel dans beaucoup de situations, une activité correspond à un diagramme d'états du modèle dynamique.

8. Les contraintes :

Les contraintes portant sur des valeurs permises d'attribut d'objets sont appelées des invariants. Les contraintes sont spécifiées à l'aide de fonctions, qui implantent toutes les possibilités de propagation rendues nécessaires par l'exploitation du système celles du modèle objet spécifient que certains objets dépendent entièrement ou partiellement d'autres objets.

Les contraintes dynamiques montrent les relations entre des états et des événements.

Les contraintes fonctionnelles spécifient des restrictions sur les opérations.

2.4. Relation entre les trois modèles : les modèles objet, dynamique et fonctionnel impliquent tous les mêmes concepts, c'est-à-dire les données, les séquencements et les opérations, mais chaque modèle se concentre sur un aspect particulier et laisse les autres aspects non interprétés.

Le modèle fonctionnel montre ce qui doit être par un système, le modèle objet montre les agissant, c'est-à-dire les objets et le modèle dynamique montre les séquences d'exécution des opérations.

❖ Par rapport au modèle fonctionnel, le modèle objet montre la structure des acteurs, des réservoirs de données et des flots du modèle fonctionnel. le modèle dynamique montre la séquence dans laquelle les traitements sont exécutés.

❖ Par rapport au modèle objet, le modèle fonctionnel montre les opérations sur les classes, et les arguments de chaque opération, il montre par conséquent la relation client-fournisseur entre classes. Le modèle dynamique montre les états de chaque objet et les opérations qui sont exécutés quand il reçoit des événements et changent d'état.

❖ Par rapport au modèle dynamique : le modèle fonctionnel montre les définitions des actions et des activités feuilles qui ne sont pas définis dans le modèle dynamique, le modèle objet montre les entités qui changent d'état et subissent les opérations.

3. Conseils pratiques à utiliser :

- Garder le modèle simple.
- Choisir les noms significatifs pour les classes avec attention.
- Éviter les associations ternaires ou plus.
- Utiliser des associations qualifiées autant que possible.
- Éviter les généralisations trop profondes.
- Ne pas nécessairement utiliser toutes les constructions d'OMT.
- Toujours documenter son modèle : cela guide le lecteur sur la compréhension des choix faits et explique les noms utilisés.
- Ne pas placer les attributs de liens dans une classe.
- Ne fournir un modèle dynamique que pour les classes qui le nécessitent.
- Utiliser des scénarios pour mettre au point.
- Rendre les diagrammes des sous classes indépendants de ceux des super classes.
- Ne définir un état que sur la base des attributs pertinents.
- Attention aux situations de priorité qui se produisent quand un objet attend des événements en provenance de plusieurs autres.

4. Règles de passage de modèle objet vers le modèle relationnel :

Ces règles ont été créées en réponse aux analystes qui font de l'analyse orientée objet et implémentent leur base de données en relationnel. Notons qu'il n'est possible de traiter que les attributs, c'est-à-dire enregistrer que l'état des objets.

Règle N° 1 : toute classe devient une relation dans laquelle :

- ❖ Les attributs de la relation traduisent les attributs de la classe.
- ❖ Si aucun des attributs de la classe ne peut jouer le rôle d'un identifiant, on ajoute un qui sera la clé primaire de la relation.

Règle N° 2 : une association 1 vers 1 se traduit en :

Portant dans une relation parmi les deux (le choix de la relation est facultatif) la clé primaire de l'autre relation, l'attribut ainsi ajouté s'appelle clé étrangère.

Règle N° 3 : une association 0 vers 1 se traduit en :

Portant dans la relation fille (la relation créée à partir de la classe ou la multiplicité de son côté égale 1) la clé primaire de la relation mère (l'autre), l'attribut ainsi ajouté s'appelle clé étrangère.

Règle N° 4 : une association N vers 1 se traduit en :

Portant dans la relation fille (la relation créée à partir de la classe ou la multiplicité de son côté égale N) la clé primaire de la relation mère (l'autre), l'attribut ainsi ajouté s'appelle clé étrangère.

Règle N° 5 : une association N vers N se traduit comme suit :

- ❖ Créant une relation particulière qui contient comme attributs les identifiants des deux classes associées (après traduction de la règle n°1).
- ❖ Ces attributs constituent à eux 2 la clé primaire de la relation.
- ❖ Ils sont individuellement clés étrangères.
- ❖ Ajoutant la ou les éventuelles propriétés de l'association (le cas d'une classe association).

Règle N° 6 : la relation d'héritage :

Les relations d'héritage dans le modèle objet ne sont pas interprétées dans le modèle relationnel, néanmoins il existe deux solutions :

- ❖ Toutes les classes fils sont soit regroupées dans une même relation et des attributs sont rajoutées pour arriver à distinguer entre les fils.

- ❖ Soit chaque fils sera interprété par une relation, et dans ce cas il faut prendre en compte les attributs et relations que la classe mère possède.

Remarque : les agrégations et compositions sont des cas particuliers de l'association et ils sont traités comme une simple association, notons seulement que la cardinalité cotée composée toujours égale à 1.

Ces règles doivent être appliquées en itérations, pour passer d'un modèle objet vers le modèle relationnel. Commençons d'abord par l'héritage, puis les classes et en fin les associations.

5. Conclusion :

Dans cette partie on a présenté les concepts de la méthode OMT et tout particulièrement les concepts des modèles objet, dynamique et fonctionnel.

Nous allons maintenant nous concentrer dans la partie suivante sur le processus des trois modèles.

CHAPITRE III

ANALYSE & CONCEPTION

1. Introduction :

Dans ce chapitre nous présentons l'approche de conception de notre système d'information en adoptant la méthode OMT. Nous exposons les différents diagrammes OMT obtenus tout au long du processus de conception.

2. Analyse :

2.1. Modélisation des objets :

La première étape de l'analyse des besoins est de construire un modèle objet, celui-ci fournit la structure statique des données du système réel et les organise en parties fonctionnelles.

2.1.1. Identification des classes des objets : la première étape dans la conception d'un modèle objet consiste à identifier les classes d'objets à partir du domaine d'application.

Légende de type de données :

Lettre utilisée	Type
A	Alphabétique
N	Numérique
D	Date

L'étude de notre problème conduit à la liste des classes suivantes :

Donneur	Flacon	Demande	Distribution
Paramètre	Test	Maladie	Patient
Groupe sanguin	Type de donneur	Type de flacon	Date de production
Technicien	Médecin	Tube	Hôpital
Service	Anticorps	Phénotype	Médecin externe
Banque de sang	Produit sanguin	Technicien interne	Médecin interne
Technicien externe	Donneur bénévole	Donneur non bénévole	

Donneur : contient les donneurs du sang.

Donneur bénévole : contient les donneurs du sang non demandé.

Donneur non bénévole : contient les donneurs du sang demandé par des patient.

Flacon : contient les poches plaines de produit sanguine.

Demande : contient les demandes du sang des patients.

Distribution : contient les distributions des flacons du sang.

Paramètre : contient les paramètres utilisés pour les analyses et le groupage.

Test : contient les tests de sérologie.

Maladie : contient les maladies dangereuses.

Patient : contient les patients qui demande du sang ou demande des analyses.

Groupe sanguine : contient les groupages du sang.

Type de donneur : contient les types des donneurs.

Type de flacon : contient les types des flacons du sang.

Date de production : contient les dates de production des paramètres des analyses.

Technicien : contient les techniciens ayant des relations avec CRTS.

Technicien interne : contient les techniciens de CRTS.

Technicien externe : contient les techniciens des hôpitaux responsables de déplacer les flacons du sang de CRTS vers l'hôpital.

Médecin : contient les médecins ayant de relations avec CRTS.

Médecin interne : contient les médecins de CRTS.

Médecin externe : contient les médecins qui traitent les patients dans les hôpitaux.

Tube : contient les tubes prélevés par des donneurs ou par des patients.

Hôpital : contient les hôpitaux qui demande du sang pour les patients.

Service : contient les services qui demande le sang dans l'hôpital.

Anticorps : contient les anticorps des patients ou des donneurs.

Phénotype : contient les phénotypes des patients ou des donneurs.

Banque de sang : contient les flacons des produits sanguines prélevés par des donneurs.

Produit sanguine : contient les produits sanguines.

2.1.2. Identification des associations :

N°	Association	Classe d'objet	Multiplicité	Attributs	Type
1	Contient	Flacon Produit	Un à plusieurs		
2	Stocké	Flacon Banque du sang	Un à plusieurs		
3	Séparer	Flacon Flacon	Un à plusieurs		
4	Concerne	Flacon Demande	Un à plusieurs		
5	Concerne	Flacon Distribution	Un à plusieurs		
6	Don	Flacon Donneur	Plusieurs à plusieurs	Type_d Date-d	A D
7	Utiliser	Test Paramètre	Un à plusieurs		
8	Avoir	Paramètre Date	Plusieurs à plusieurs	Date_p Qte	D N
9	Tirer	Paramètre Technicien interne Date	Plusieurs à plusieurs	Qte_t	N
10	Réaliser	Tube Test Technicien interne	Plusieurs à plusieurs	Rst_t Obs_t	A A

N°	Association	Classe d'objet	Multiplicité	Attributs	Type
11	Réaliser	Flacon Test Technicien interne	Plusieurs à plusieurs	Rst_f Obsr_f	A A
12	Grouper	Groupe sanguine Donneur	Un à plusieurs		
13	Est de phénotype	Phénotype Donneur	Un à plusieurs		
14	Porte	Donneur Maladie	Plusieurs à plusieurs		
15	Consulter	Donneur Médecin interne Date	Plusieurs à plusieurs	TA Poids Obs	N N A
16	Hospitaliser	Patient Hôpital Date	Plusieurs à plusieurs	N_dos	A
17	Porte	Patient Maladie	Plusieurs à plusieurs		
18	Suivi dans	Patient Service Date	Plusieurs à plusieurs		
19	Grouper	Patient Groupe sanguine	Un à plusieurs		
20	Est de phénotype	Patient Phénotype	Un à plusieurs		
21	Concerne	Patient Demande	Un à plusieurs		
22	Prélevé	Patient Tube	Un à plusieurs		
23	Envoie	Médecin externe Demande	Un à plusieurs		
24	Travail pour	Médecin externe Hôpital	Un à plusieurs		
25	Concerne	Demande Groupe sanguin	Un à plusieurs		
26	Concerne	Demande Produit sanguine	Un à plusieurs	Qte_d	A
27	Travail pour	Technicien externe Hôpital	Un à plusieurs		
28	Concerne	Demande Phénotype	Un à plusieurs		
29	Suivi par	Distribution Technicien interne	Un à plusieurs		
30	Recevait par	Distribution Technicien externe	Un à plusieurs		

2.1.3. Dictionnaire de données : un dictionnaire est un paragraphe décrivant chaque classe d'objets, il décrit le cadre d'utilisation de la classe dans le problème courant, avec les hypothèses et les restrictions faites sur ces membres et son utilisation.

Le dictionnaire doit aussi inclure associations, attributs et opérations.

Classe	Attributs	Signification	Opérations	Type	Taille
Donneur	MAT_DN	Matricule de donneur.	- Ajouter	A	10
	NOM_DN	Nom de donneur.	donneur.	A	20
	PNOM_DN	Prénom de donneur.	- Supprimer	A	20
	DT_N_DN	Date de naissance de donneur.	donneur.	D	10
	L_N_DN	Lieu de naissance de donneur.	- exclure un	A	20
	SX_DN	Sexe de donneur.	donneur.	A	20
	ET_C_DN	Etat civil de donneur.	- MAJ donneur.	A	20
	ADRS_PE	Adresse personnelle de donneur.	- exclu au don	A	20
	ADRS_PO	Adresse professionnelle de donneur.	temporairement	A	50
	VIL_PE	Ville personnelle de donneur.	- exclu au don	A	50
	VIL_PO	Ville professionnelle de donneur.	définitivement	A	50
	W_PE	Wilaya personnelle de donneur.		A	20
	W_PO	Wilaya professionnelle de donneur.		A	20
	TEL_PE	Téléphone personnel de donneur.		A	20
	TEL_PO	Téléphone professionnelle de donneur.		A	20
Patient	MAT_PT	Matricule de patient.	- Ajouter	A	12
	NOM_PT	Nom de patient.	patient.	A	20
	NOM_J_PT	Nom de jeune fille patient.	- Supprimer	A	20
	PNOM_PT	Prénom de patient.	patient.	A	20
	D_N_PT	Date de naissance de patient.	- MAJ patient.	D	10
	L_N_PT	Lieu de naissance de patient.		A	20
	SX_PT	Sexe de patient.		A	20
	ET_C_PT	Etat civil de patient.		A	20
	ADRS_PT	Adresse de patient.		A	50
	TEL_PT	Téléphone de patient		A	20
Médecin	MAT_MD	Matricule de médecin.	- Ajouter	A	07
	NOM_MD	Nom de médecin.	médecin.	A	20
	PNOM_MD	Prénom de médecin.	- Supprimer	A	20
	ADRS_MD	Adresse de médecin.	médecin.	A	50
	TEL_MD	Téléphone de médecin.	- MAJ médecin.	A	20

Classe	Attributs	Signification	Opérations	Type	Taille
Distribution	N_DTB DATE	Numéro de distribution. Date de distribution.	- Ajouter et supprimer et MAJ.	A D	08 08
Paramètre	COD_PTRE DS_PTRE	Code de paramètre. Désignation de paramètre.	- Ajouter et supprimer et MAJ.	A A	03 30
Date	DATE	Date	- Ajouter et supprimer et MAJ.	D	10
Flacon	COD_FL CC_FL	Code de flacon. Volume de flacon.	- Ajouter et supprimer et MAJ. - Distribuer - Séparer - Jeter - Réserver	A N	13 06
Groupe sanguine	GP_S	Groupe sanguine.	- Ajouter et supprimer et MAJ.	A	03
Technicien	MAT_TH NOM_TH PNOM_TH ADRS_TH TEL_TH N_CP_TH	Matricule de technicien. Nom de technicien. Prénom de technicien. Adresse de technicien. Téléphone de technicien. Numéro de carte professionnelle de technicien.	- Ajouter et supprimer et MAJ.	A A A A A A	07 20 20 50 20 20
Phénotype	PHTYPE	Phénotype	- Ajouter et supprimer et MAJ.	A	20
Service	N_SV DS_SV	Numéro de service. Désignation de service.	- Ajouter et supprimer et MAJ.	A A	04 30
Médecin externe	L_TR_MD	Lieu de travail de médecin.	- Ajouter médecin. - Supprimer médecin. - MAJ médecin.	A	20
Hôpital	N_HP DS_HP	Numéro de l'hôpital. Désignation de l'hôpital.	- Ajouter et supprimer et MAJ.	A A	07 30
Banque	N_BQ	Numéro de banque.	- Ajouter et supprimer et MAJ.	A	03

Classe	Attributs	Signification	Opérations	Type	Taille
Tube	N_TB DADE	Numéro de tube Date de prélèvement.	- Ajouter et supprimer et MAJ.	A	07
Test	COD_TST DS_TST	Code de teste. Désignation de teste.	- Ajouter et supprimer et MAJ.	A A	03 30
Demande	N_DM DATE_D DATE_S	Numéro de demande. Date de demande. Date souhaitée.	- Ajouter et supprimer et MAJ.	A D D	08 08 08
Technicien externe	L_TR_TH	Lieu de travail de technicien.	- Ajouter et supprimer et MAJ.	A	20
Maladie	COD_ML DS_ML	Code de maladie. Désignation de maladie	- Ajouter et supprimer et MAJ.	A A	04 30
Produit sanguin	COD_PD DS_PD DL_PRT	Code de produit. Désignation de produit. Délais de préemption.	- Ajouter et supprimer et MAJ.	A A A	04 30 20

2.1.4. Utilisation d'héritage : L'héritage peut être ajoutée de deux manières :

En généralisant les aspects communs à plusieurs classes dans une super classe (généralisation) ou en affinant les classes existantes dans des sous classes (spécialisation).

❖ Généralisation des classes :

- **Flacon** : classe représentant les propriétés communes aux classes flacon simple, flacon double et flacon triple.
- **Donneur** : classe représentant les propriétés communes aux classes donneur bénévole et donneur demandé.

❖ Spécialisation des classes :

- **Technicien** : classe est affiné dans des sous classes technicien interne et technicien externe.
- **Médecin** : classe est affiné dans des sous classes médecin interne et médecin externe.

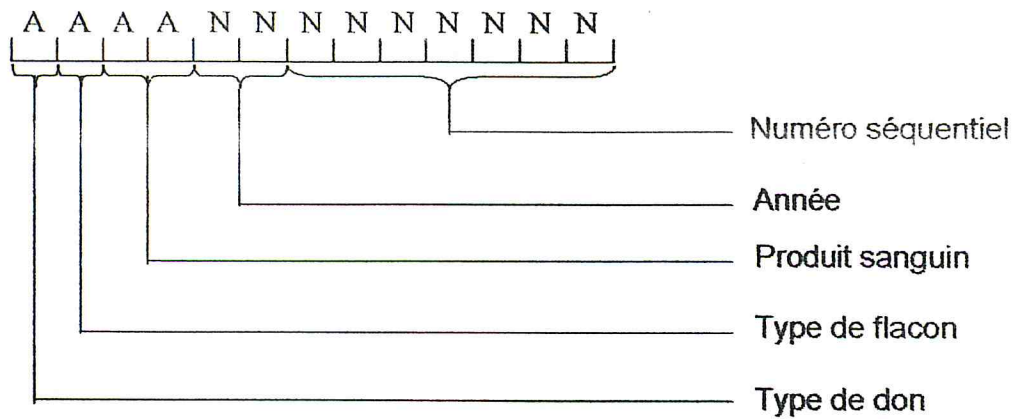
2.1.5. Codification proposée :

Répondant aux objectifs du système d'information qui consiste dans son ensemble à assurer de meilleures performances, j'ai proposé quelques codifications. Ces dernières vérifiées les règles suivantes :

- **L'unicité** : un code ne correspond qu'un seul objet.
- **L'efficacité** : identification et facilité de recherche.
- **L'extensibilité** : facilité de rajouter d'autres objets.

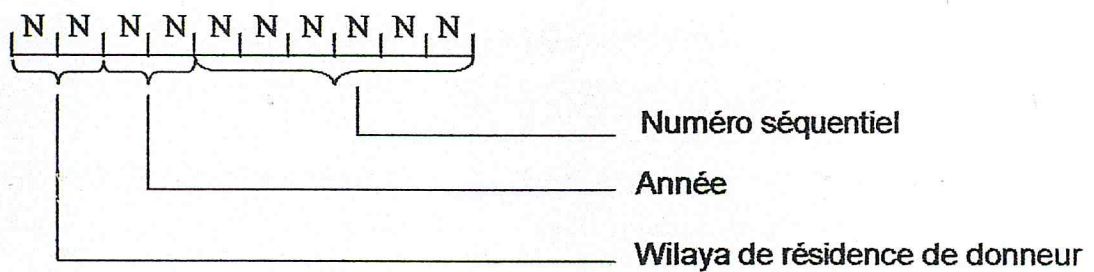
Description des codifications proposées:

1. Code de flacon :(COD-FL)



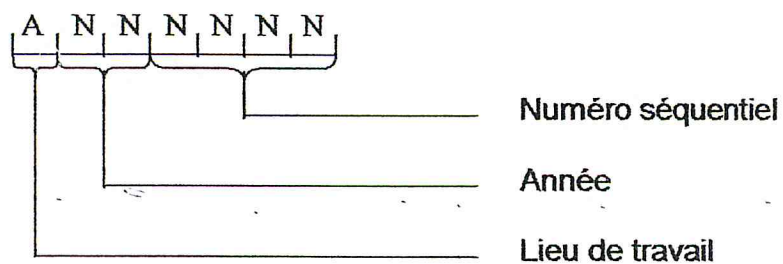
EXP : VDST001234412

2. Code de donneur :(MAT-DN)



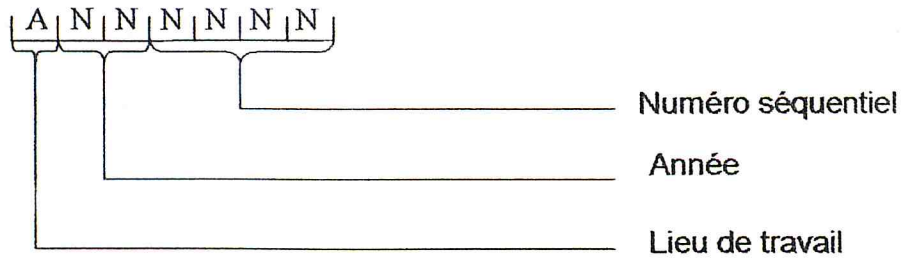
EXP : 0905000001

3. Code de médecin : (MAT-MD)



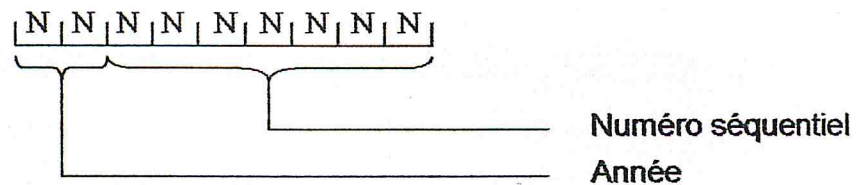
EXP : N00160145

4. Code de technicien : (MAT-TH)



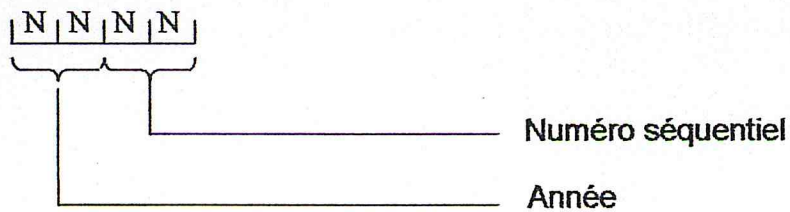
EXP : X9910023

5. Code de patient : (COD-PT)



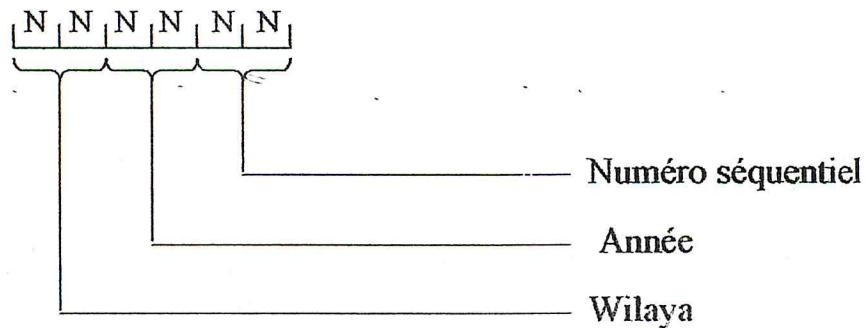
EXP : T99091000321

6. Code de service : (N-SV)



EXP : 0125

7. Code d'hôpital : (N-HP)



EXP : 090201

8. Code de tube : (N-TB)

N	N	N	N	N	N	N	N
---	---	---	---	---	---	---	---

Numéro séquentiel

Année

EXP : 0132105

9. Code de produit sanguin : (COD-PD)

A	A	A	A
---	---	---	---

Code alphabétique

EXP : SGTL

10. Code de maladie : (COD-ML)

N	N	N	N
---	---	---	---

Numéro séquentiel

Année

EXP : 0102

11. Code de distribution : (N-DST)

N	N	N	N	N	N	N	N	N
---	---	---	---	---	---	---	---	---

Numéro séquentiel

Année

EXP : 02000010

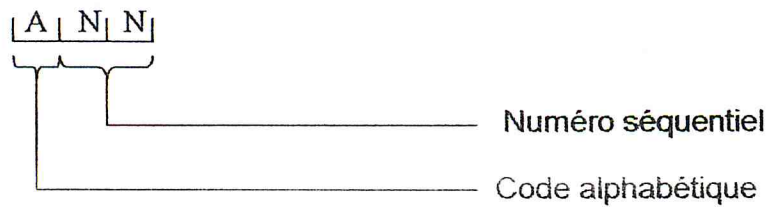
12. Code de demande : (N-DM)

N	N	N	N	N	N	N	N	N
---	---	---	---	---	---	---	---	---

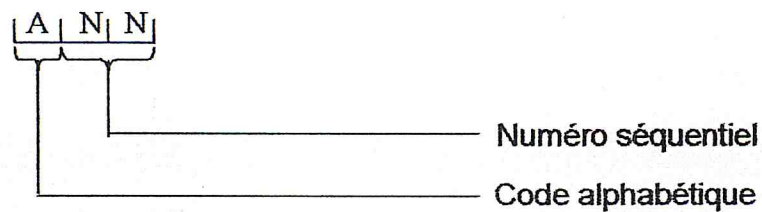
Numéro séquentiel

Année

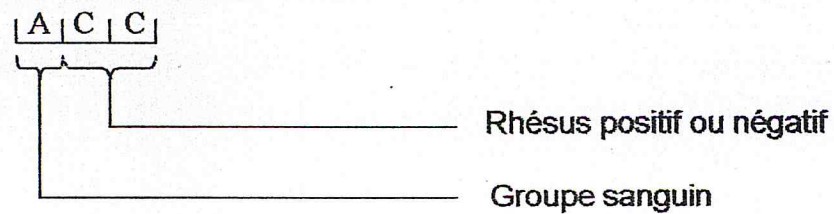
EXP : 01012516

13. Code de test : (COD-TST)

EXP : H01

14. Code de paramètre : (COD-PTRE)

EXP : S15

15. Code de groupe sanguin : (GP-S)

EXP : AB+

16. Code de banque : (N-BQ)

EXP : PM1

2.1.6. Diagramme de classes :

❖ Feuille 1 : le donneur du sang.

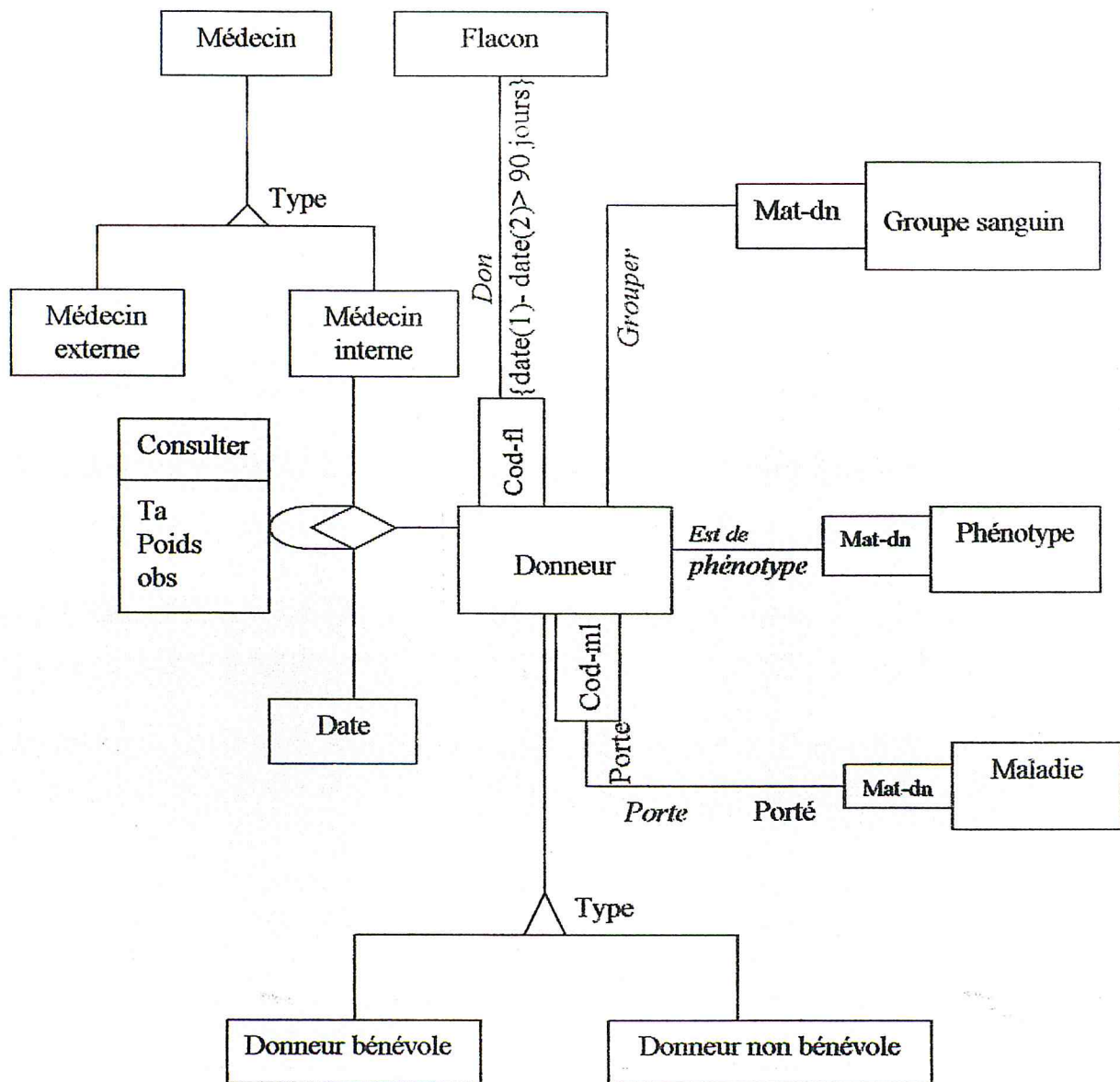


Figure 31 : Feuille (donneur du sang) de diagramme d'objet.

❖ Feuille 2 : le patient.

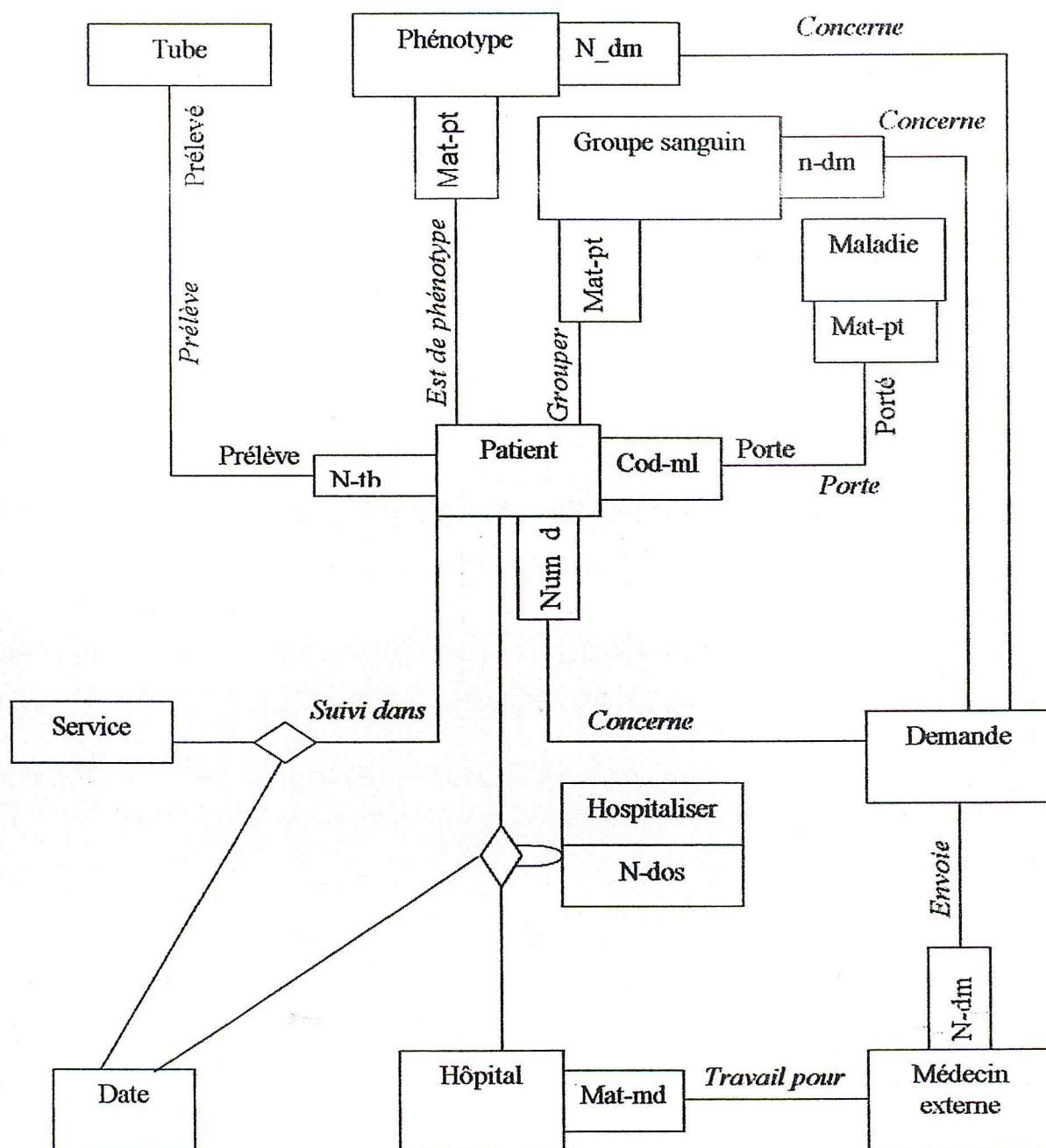


Figure 32 : Feuille (patient et la demande du sang) de diagramme d'objet.

❖ Feuille 3 : Analyse du sang.

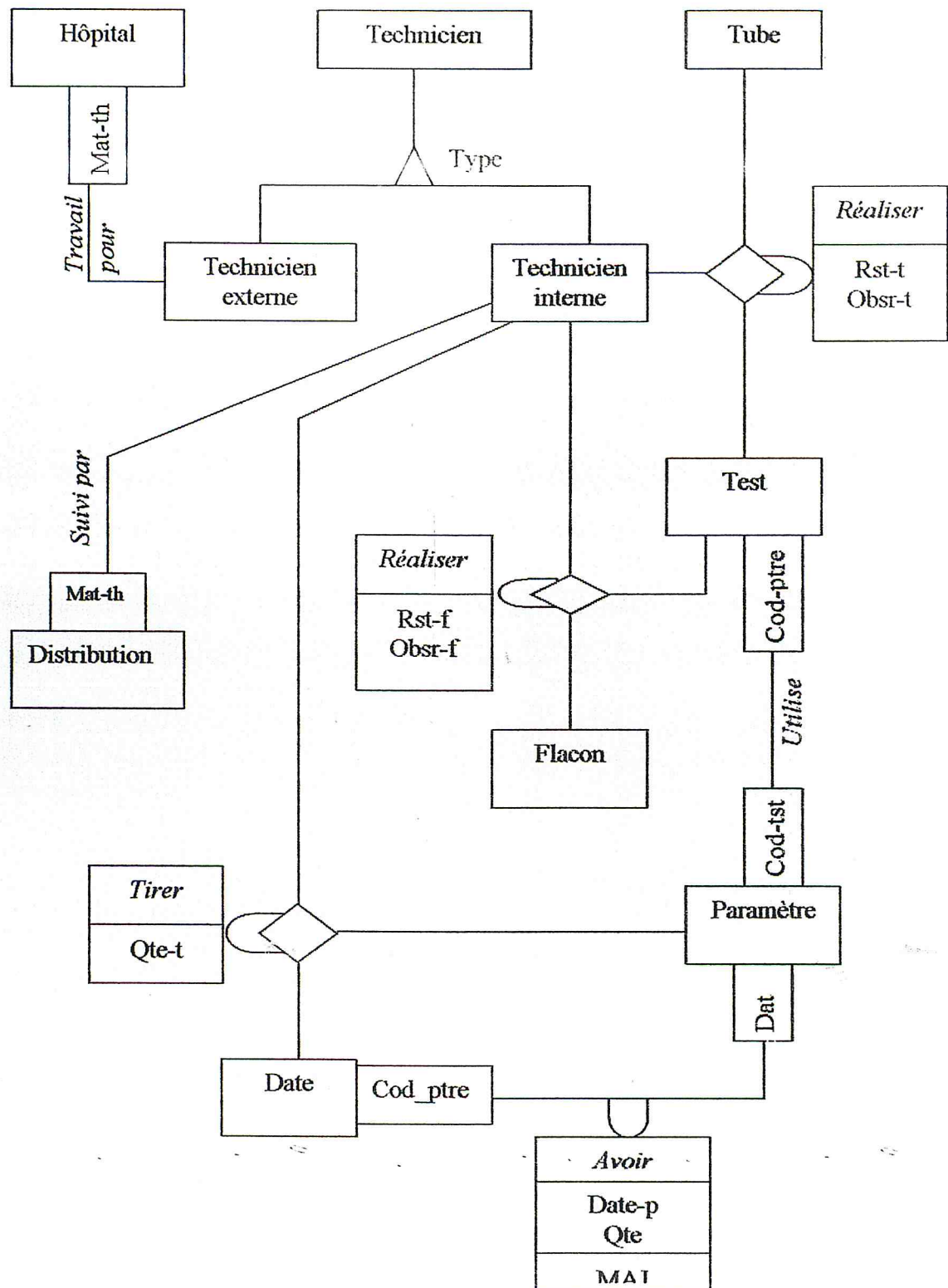


Figure 33 : Feuille (analyse du sang) de diagramme d'objet.

❖ Feuille 4 : distribution du sang.

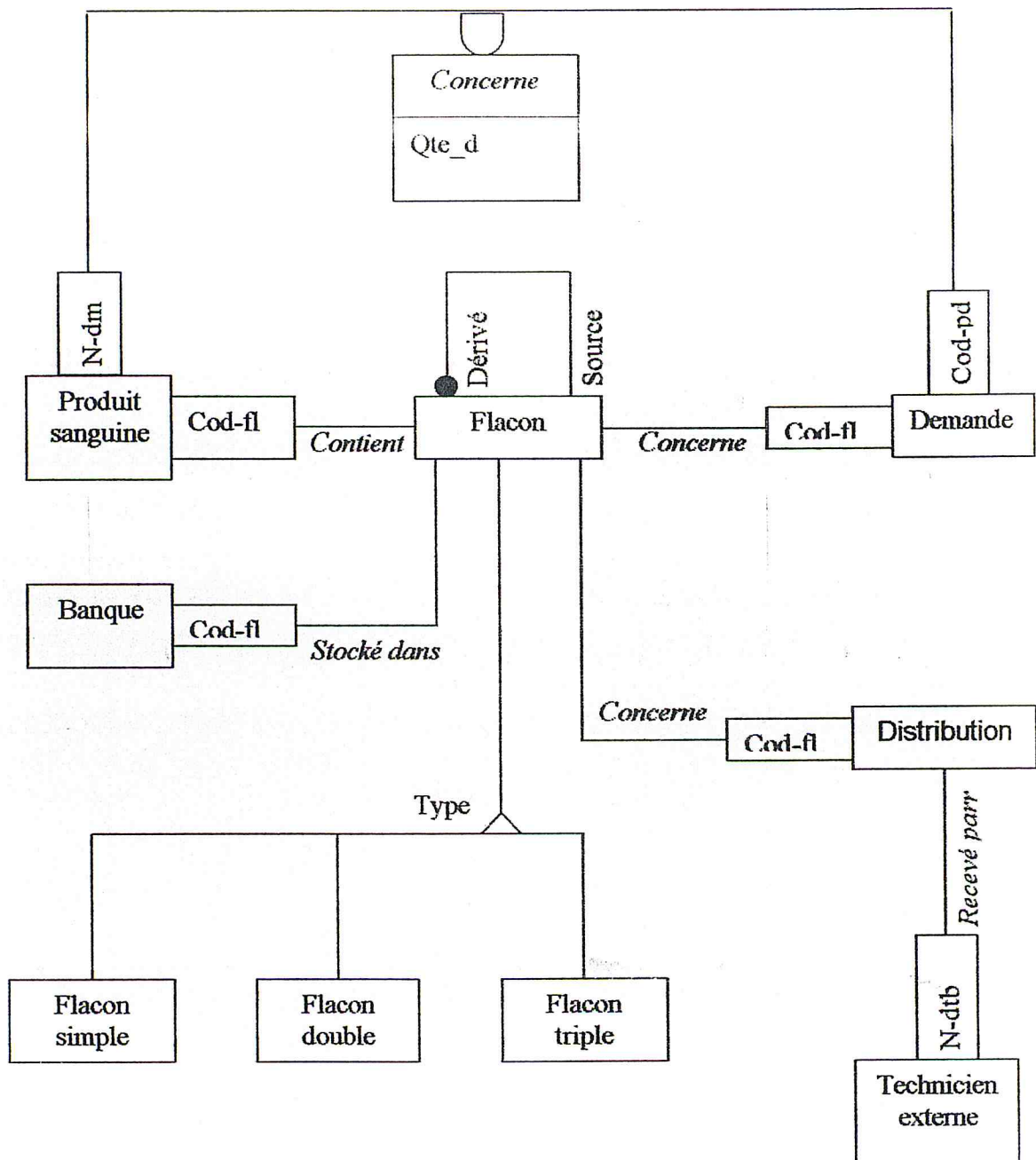


Figure 34 : Feuille (séparation et distribution du sang) de diagramme d'objet.

2.2. Le modèle dynamique :

Le modèle dynamique s'intéresse aux aspects temporels du système et aux objets à l'intérieur de ce système.

2.2.1. Scénario et suivis d'évènements :

❖ Scénario de consultation :

➤ Scénario normal d'une consultation :

- ✓ Mise à jour de fichier donneur.
- ✓ Mise à jour de fichier consulter.
- ✓ Edition d'un bon de prélèvement.

➤ Scénario d'évènement :

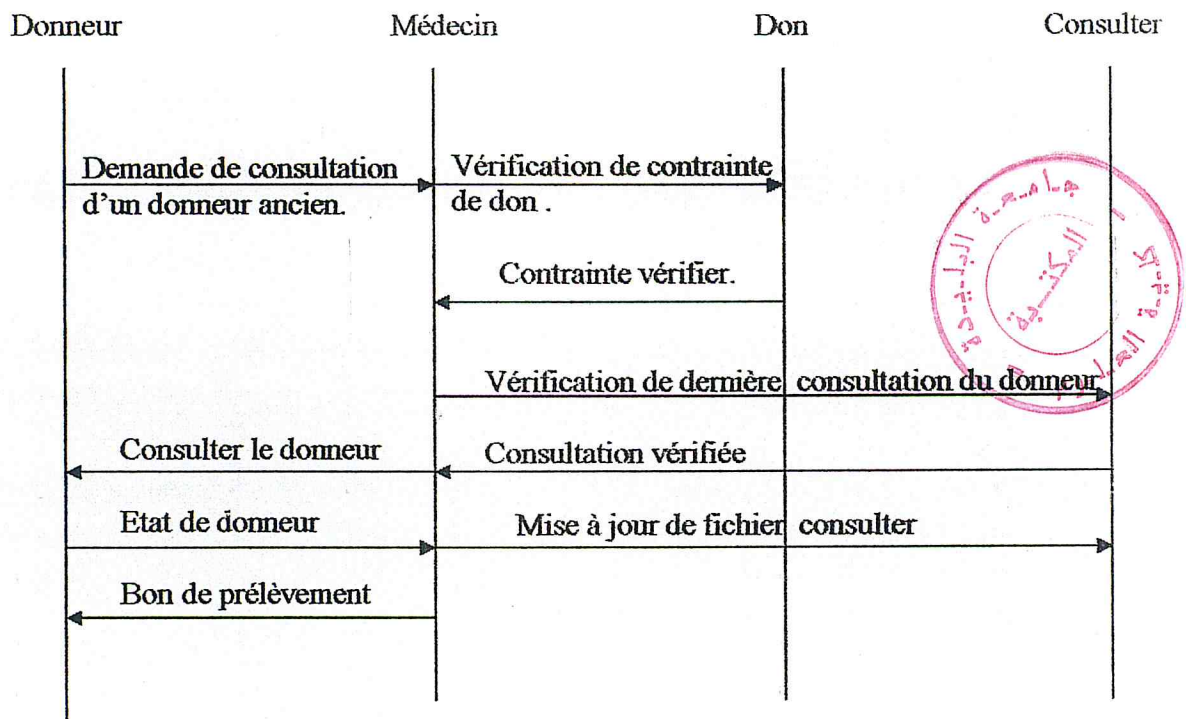


Figure 35 : Scénario de consultation.

- ❖ **Scénario de prélèvement :**
 - **Scénario normal d'un prélèvement :**
 - ✓ Mise à jour de fichier flacon.
 - ✓ Mise à jour de fichier don.
 - **Scénario d'évènement :**

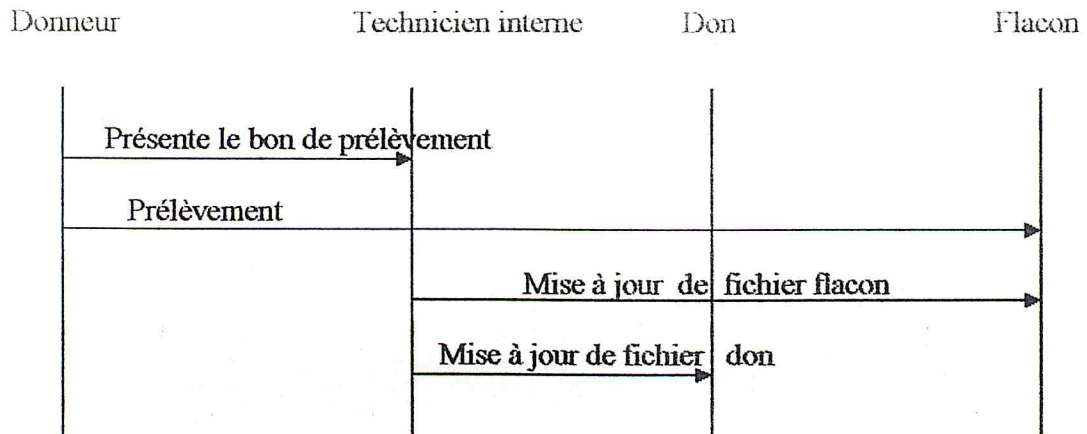


Figure 36 : Scénario de prélèvement.

- ❖ **Scénario de sérologie :**
 - **Scénario normal de sérologie :**
 - ✓ Mise à jour de fichier réaliser.
 - **Scénario d'évènement :**

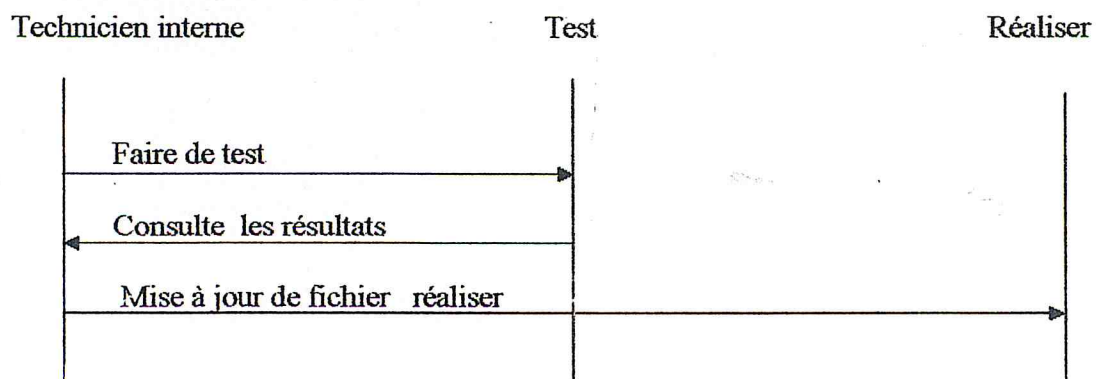


Figure 37 : Scénario de sérologie.

❖ Scénario de groupage :

➤ Scénario normal de groupage :

- ✓ Mise à jour de fichier réaliser
- ✓ Mise à jour de fichier flacon.
- ✓ Mise à jour de fichier donneur.

➤ Scénario d'évènement :

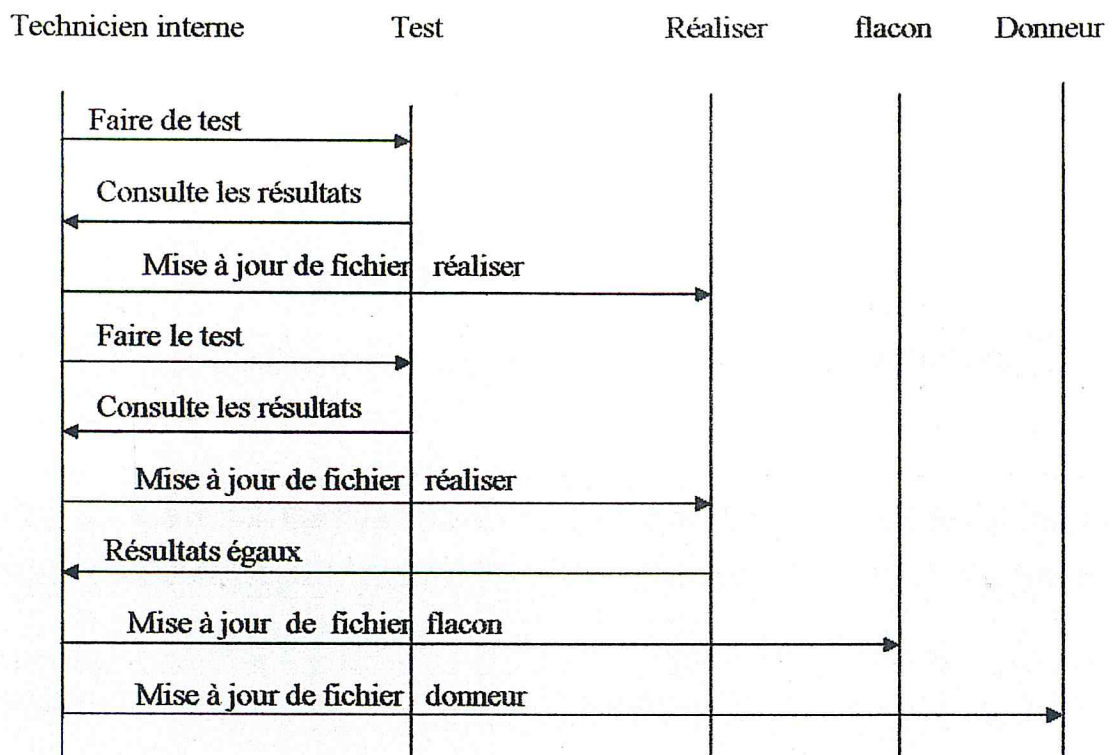


Figure 38 : Scénario de groupage.

❖ Scénario de distribution :

➤ Scénario normal de distribution :

- ✓ Mise à jour de fichier demande.
- ✓ Mise à jour de fichier patient.
- ✓ Mise à jour de fichier médecin externe.
- ✓ Mise à jour de fichier technicien externe.
- ✓ Mise à jour de fichier flacon.
- ✓ Mise à jour de fichier distribution.

➤ Scénario d'évènement :

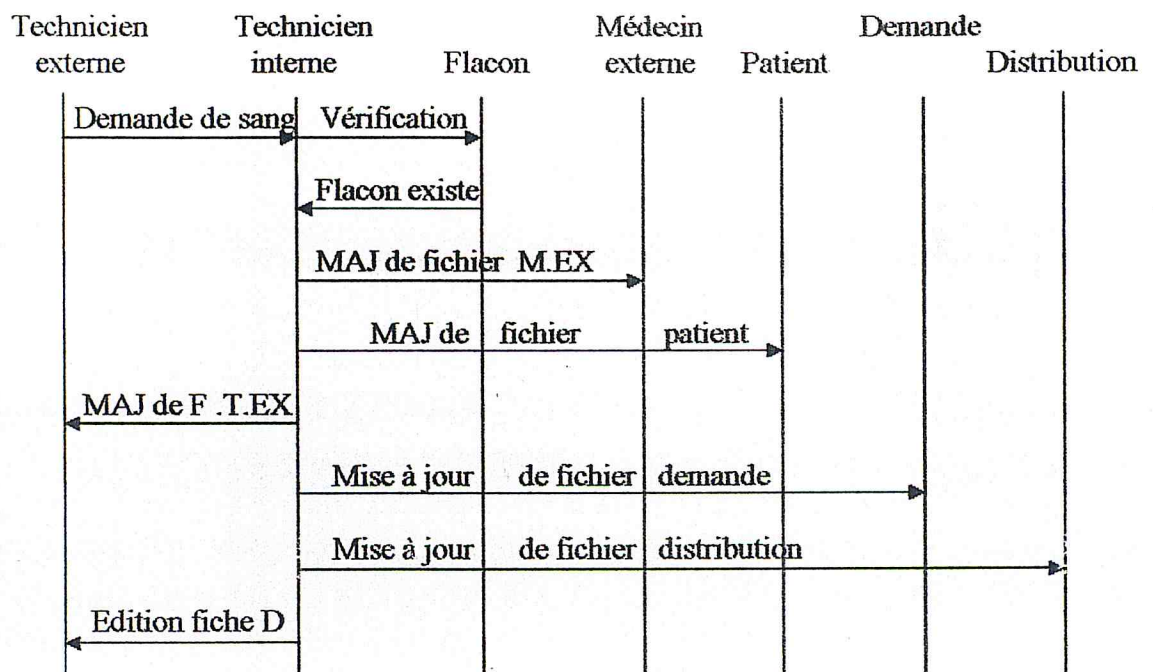


Figure 39 : Scénario de distribution.

❖ **Scénario de paramètre :**➤ **Scénario normal de paramètre :**

- ✓ Demande de paramètre.
- ✓ Vérification de stocke.
- ✓ Mise à jour de fichier tirer.
- ✓ Mise à jour de fichier avoir.

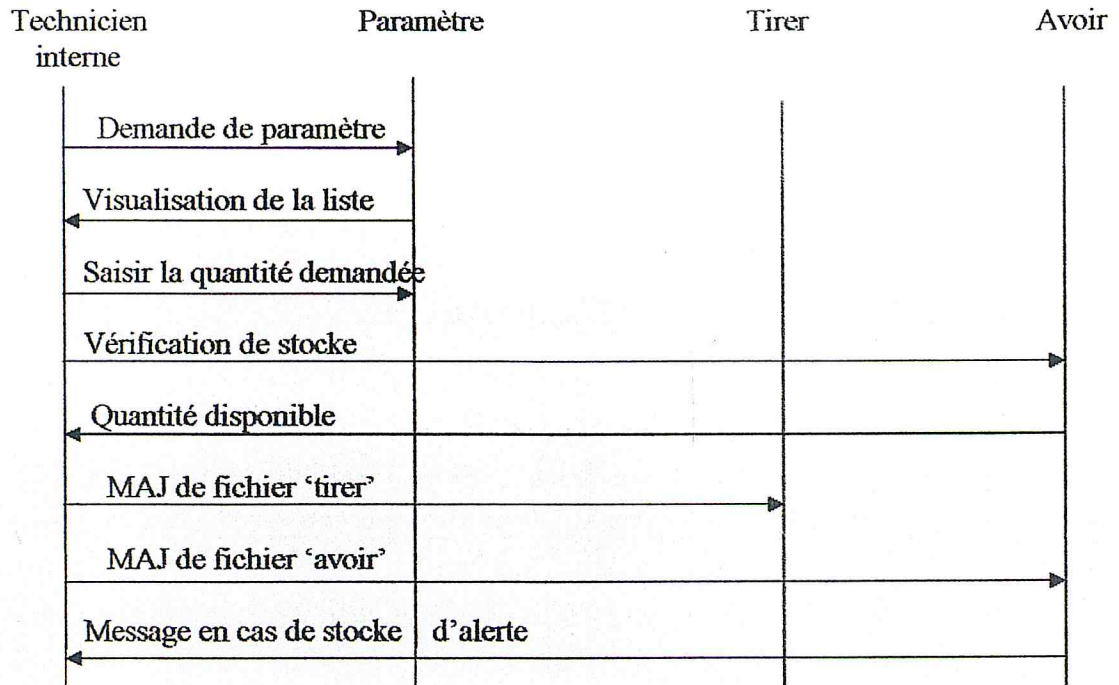
➤ **Scénario d'évènement :**

Figure 40 : Scénario de paramètre.

❖ Scénario des analyses :

➤ Scénario normal d'analyse :

- ✓ Mise à jour de fichier 'patient'.
- ✓ Mise à jour de fichier 'tube'.
- ✓ Mise à jour de fichier 'réaliser'.
- ✓ Edition de carte de groupage ou fiche de sérologie.

➤ Scénario d'évènement :

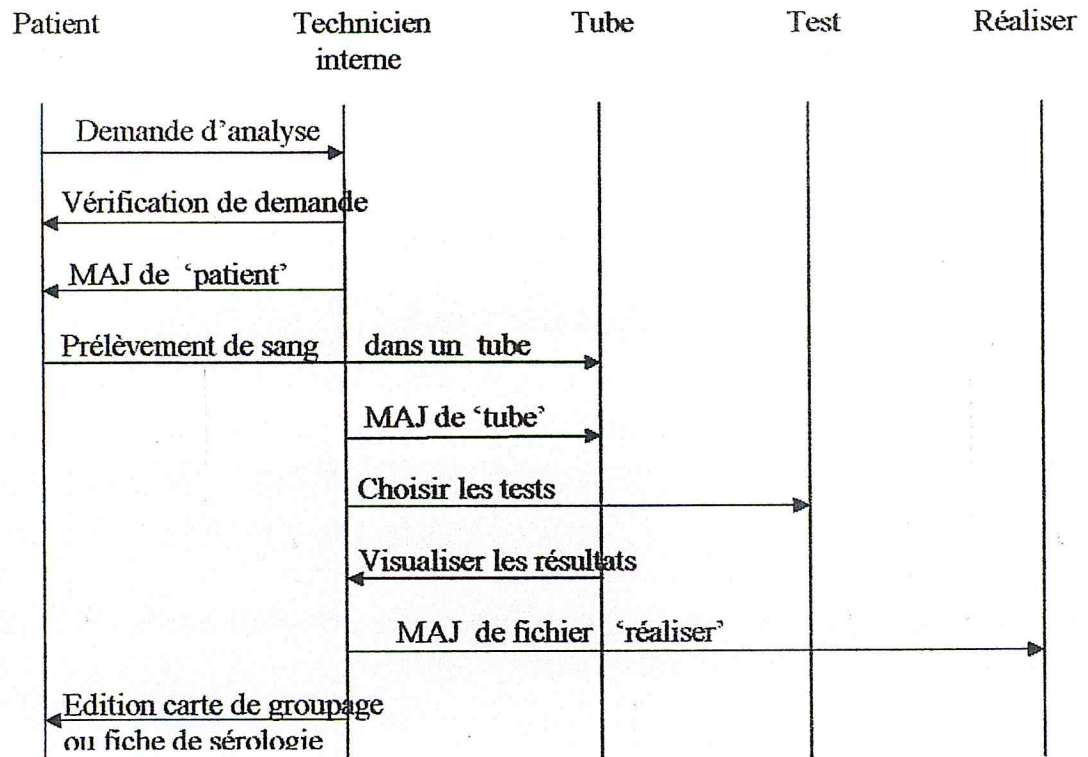


Figure 41 : Scénario des analyses.

2.2.2. Diagramme d'état :

➤ Diagramme d'état de donneur :

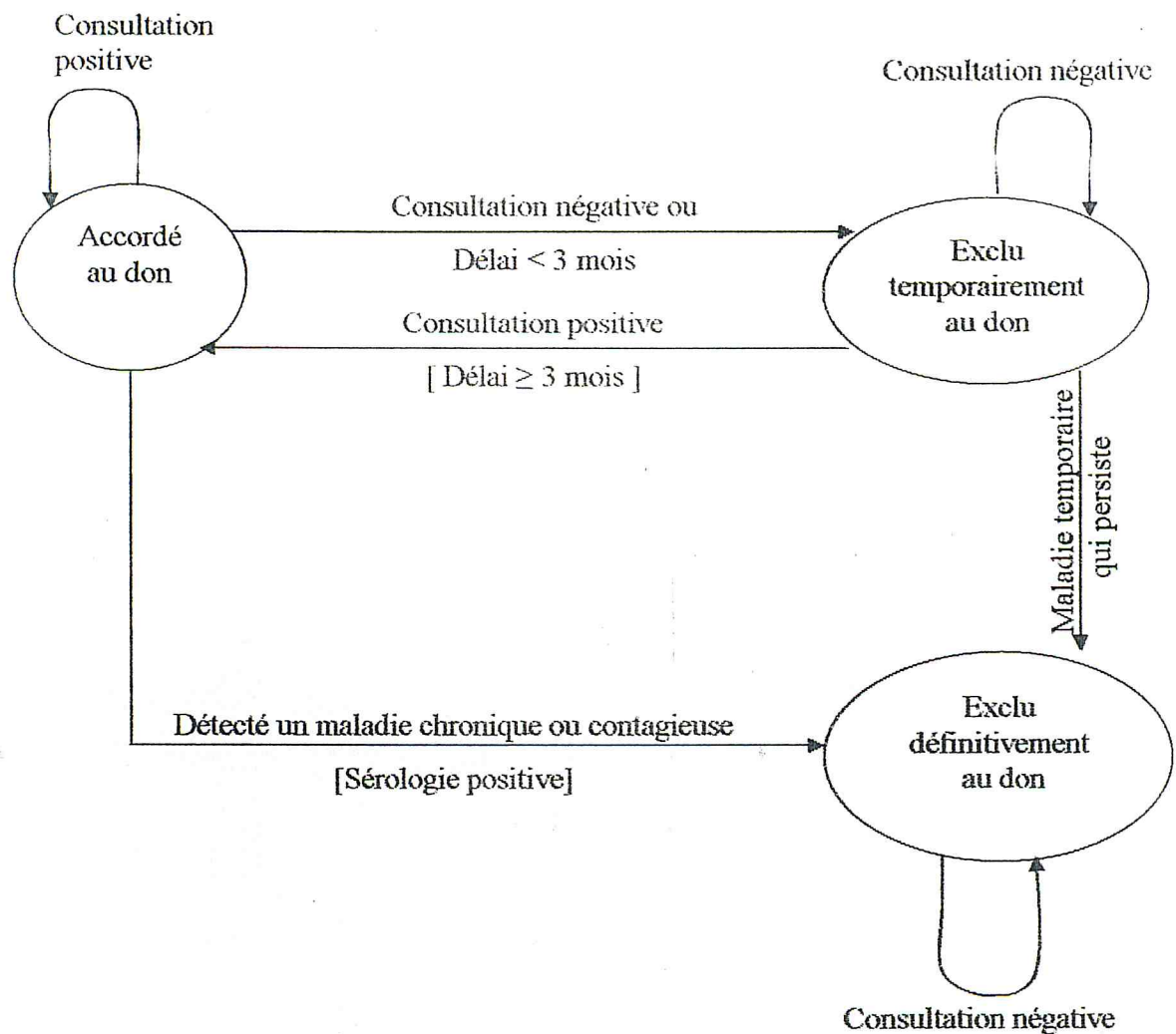


Figure 42 : Diagramme d'état de donneur.

➤ Diagramme d'état de flacon :

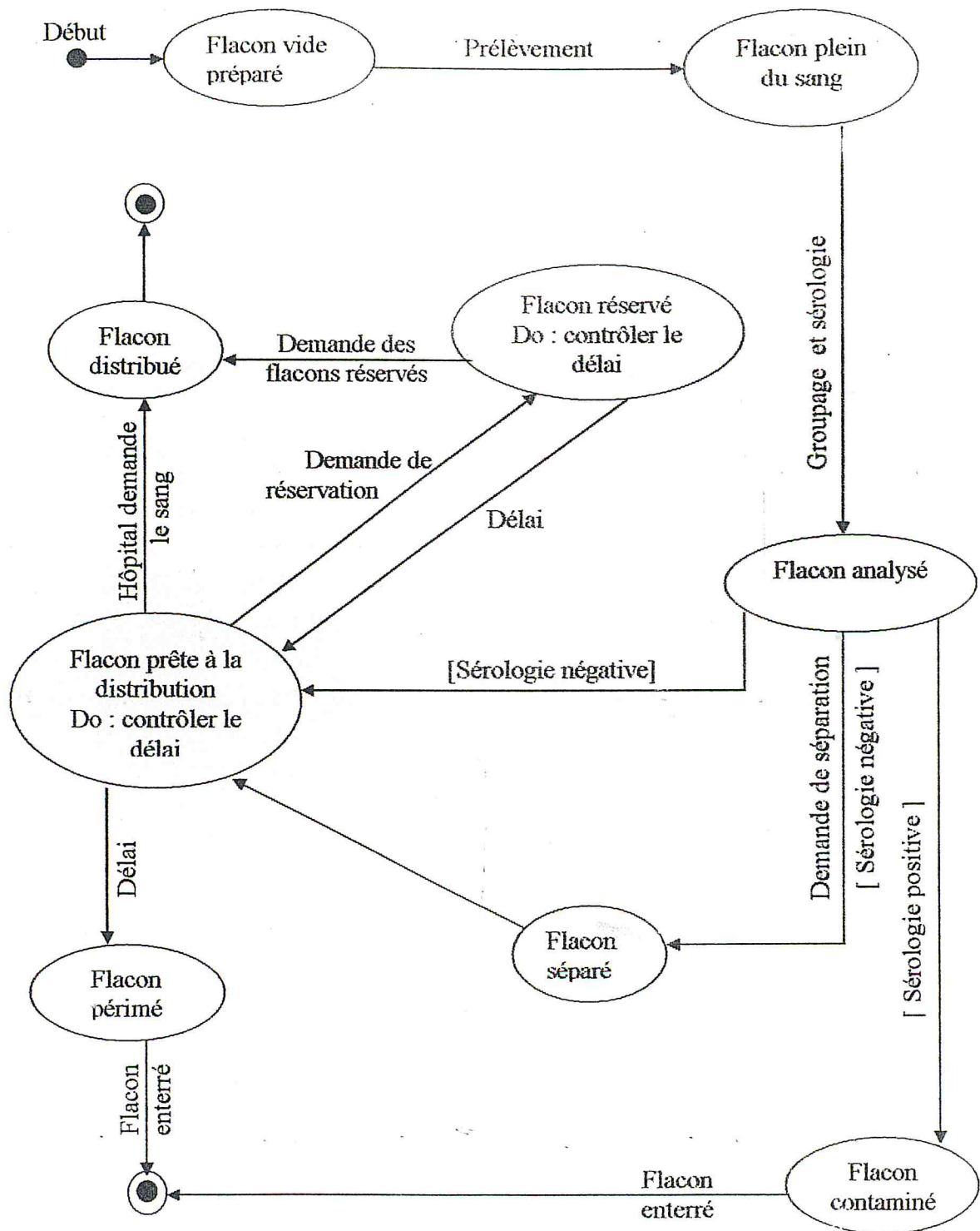


Figure 43 : Diagramme d'état de flacon.

➤ Diagramme d'état de stocke de paramètres :

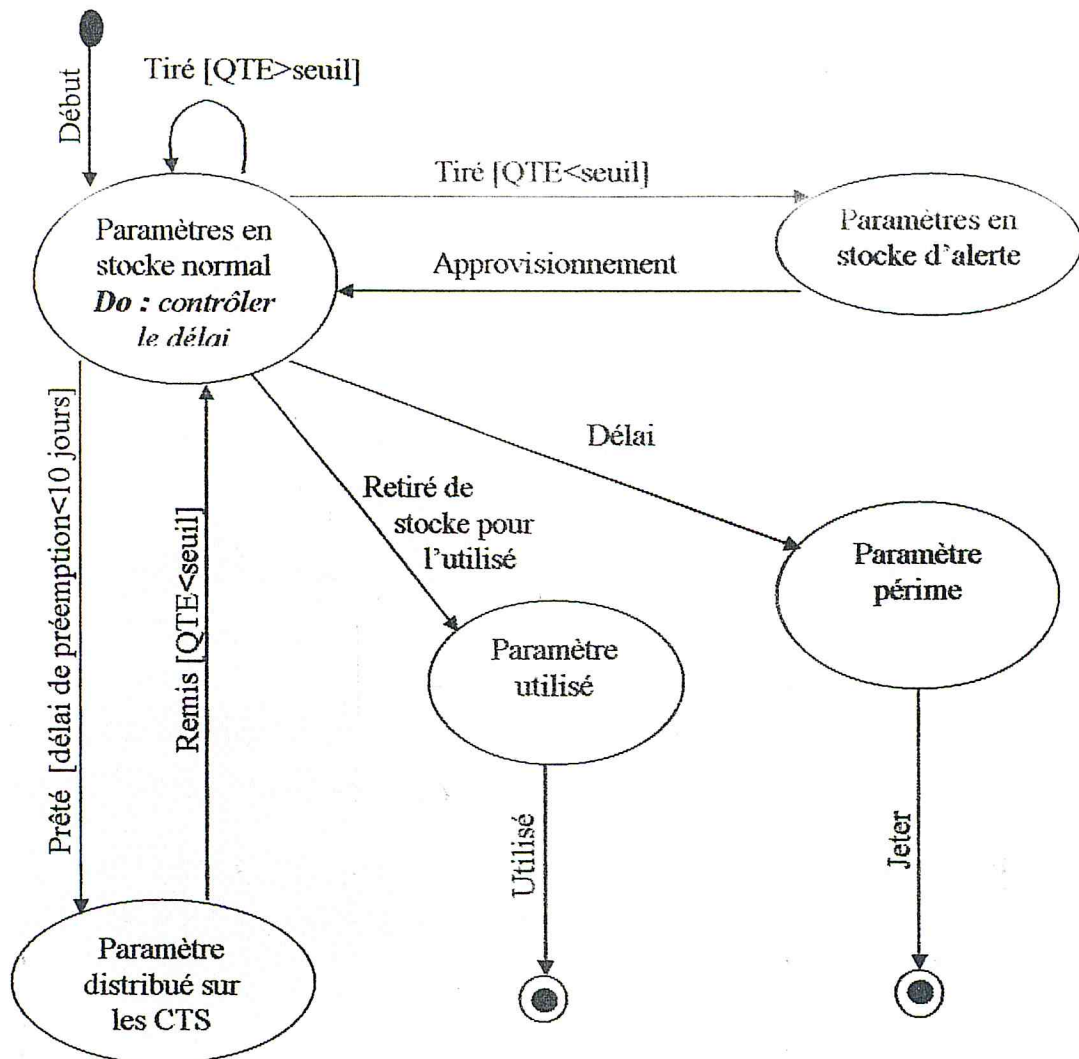


Figure 44 : Diagramme d'état de stocke de paramètre.

2.3. Le modèle fonctionnel :

1. Diagramme de flot de donnée du traitement « consultation » :

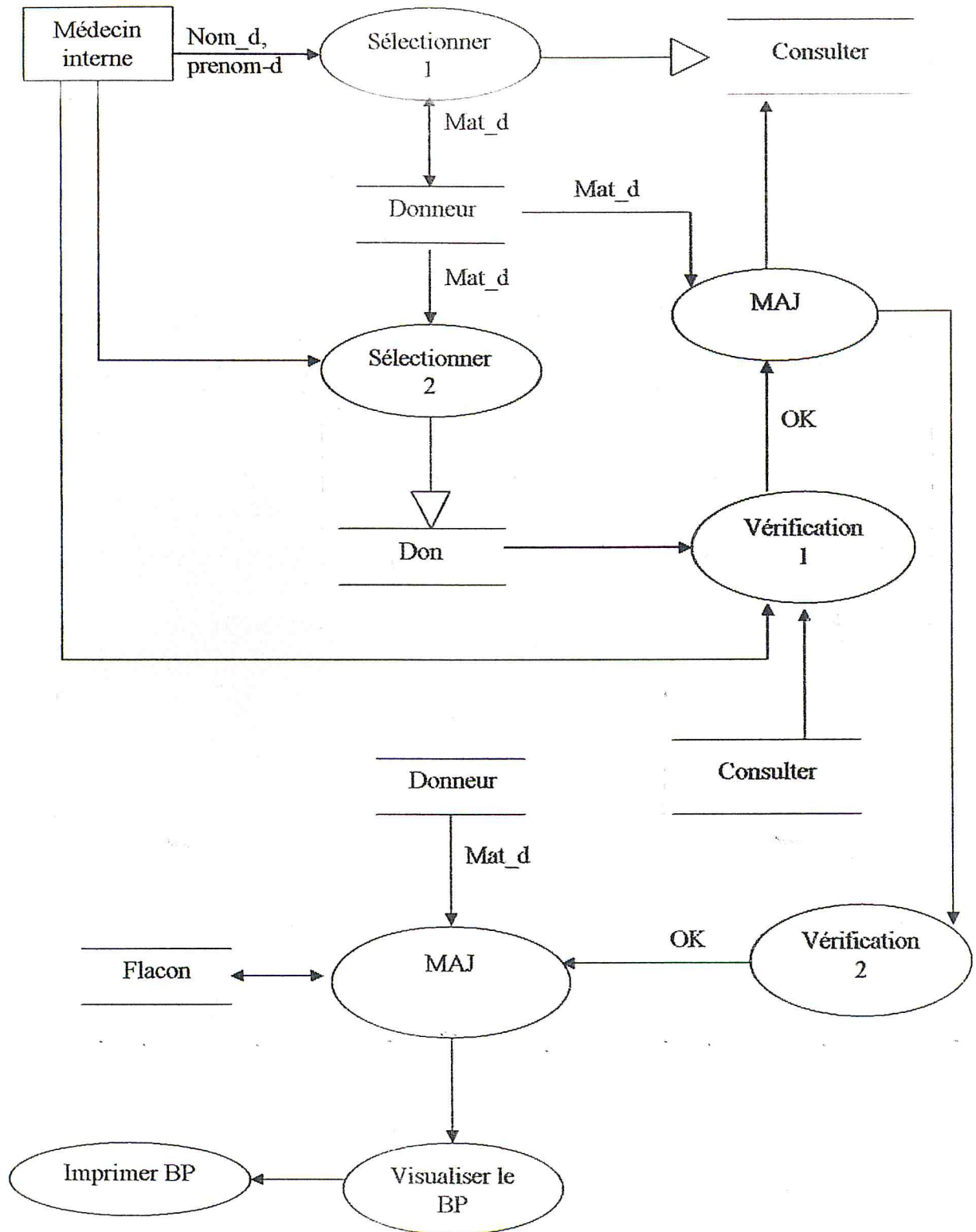
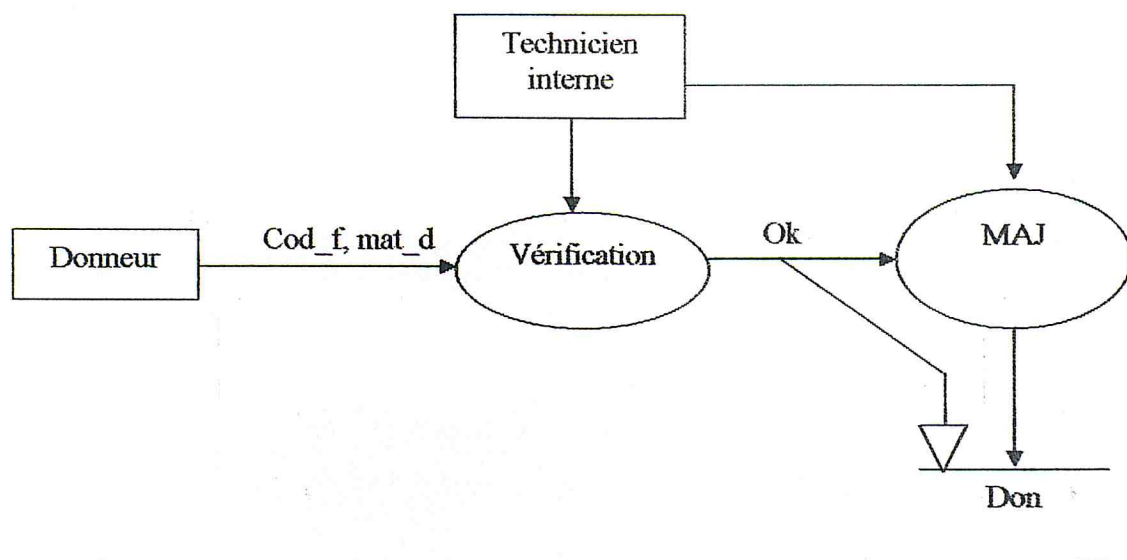


Figure 45 : DFD de consultation.

Description :

- **Créer consultation** : créer un nouveau objet dans le fichier consulter.
- **Vérification1** : vérifier si le donneur n'est pas exclu définitivement ou temporellement au don.
- **Vérification2** : vérifier si les résultats de consultation sont positifs.
- **Imprimer BP** : imprimer le bon de prélèvement.
- **Créer un objet flacon** : créer un nouveau objet dans le fichier flacon.

2. Diagramme de flot de donnée du traitement « prélèvement » :**Figure 46 : DFD de prélèvement.****Description :**

- **Créer un objet don** : créer un nouveau objet dans le fichier don.
- **Vérification** : vérifier le bon de prélèvement et le code de la poche de sang vide de donneur.

3. Diagramme de flot de donnée du traitement « sérologie » :

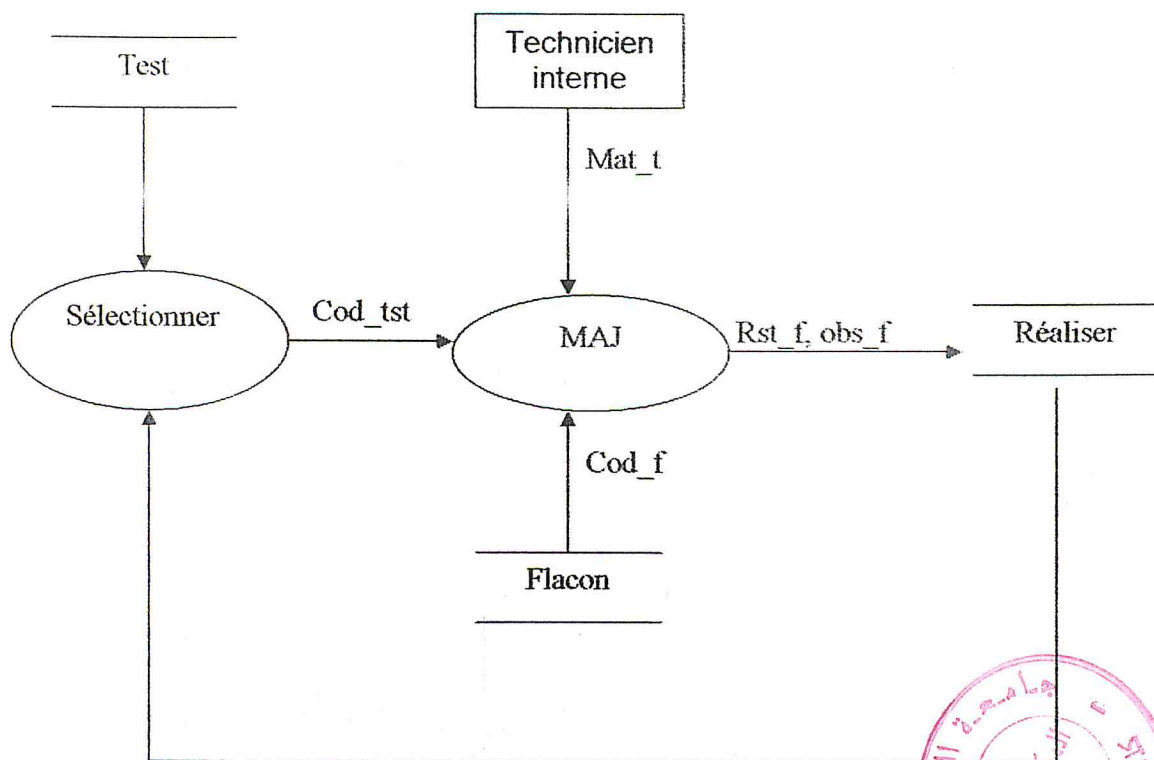


Figure 47 : DFD de sérologie.

Description :

- Créer un objet réaliser : créer un nouveau objet dans le fichier réaliser.
- Sélectionner : sélectionner une test à réaliser.

4. Diagramme de flot de donnée du traitement « groupage » :

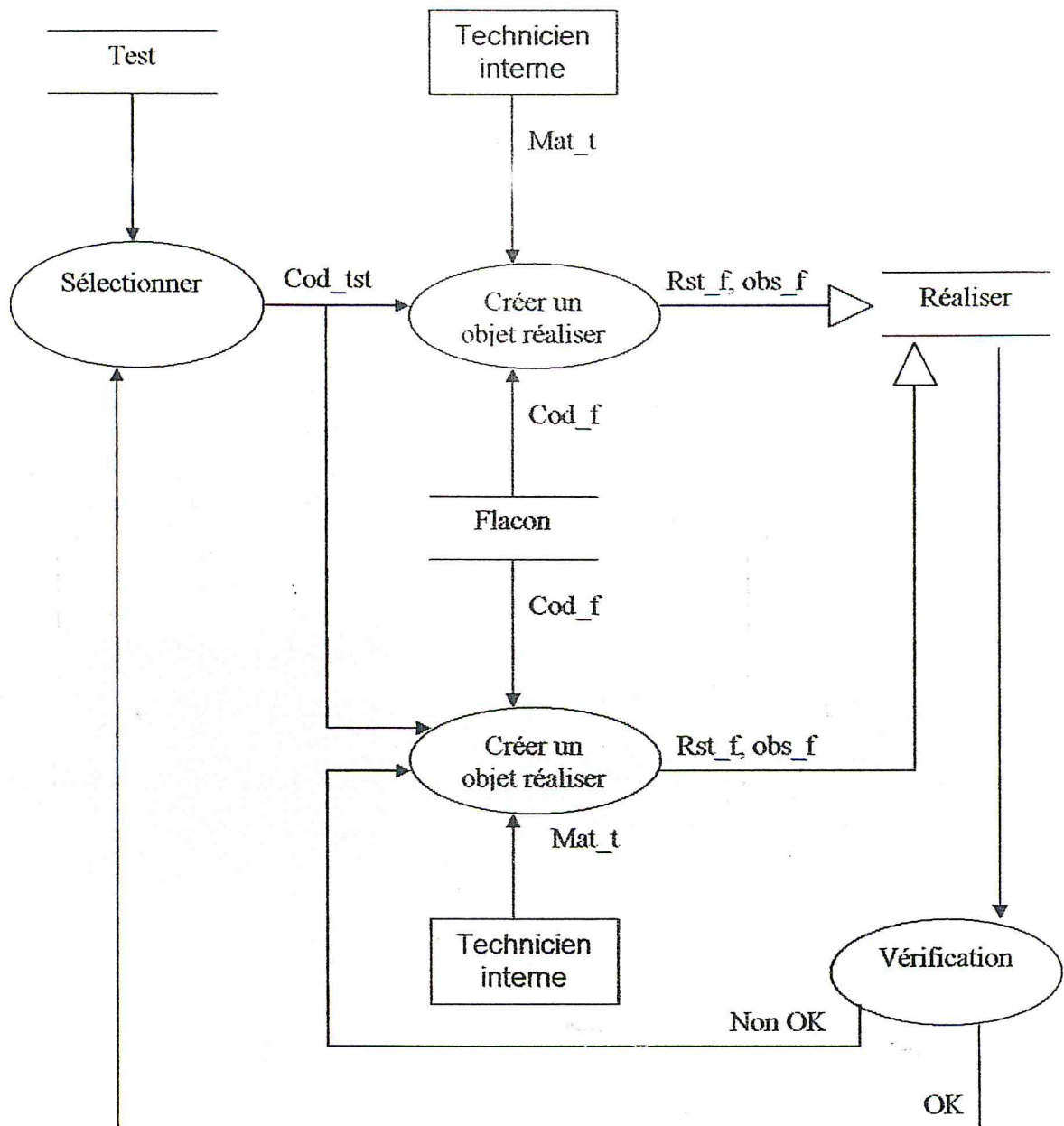


Figure 48 : DFD de groupage.

Description :

- **Créer un objet réaliser** : créer un nouveau objet dans le fichier réaliser deux fois pour la même test.
- **Sélectionner** : sélectionner une test à réaliser.
- **Vérification** : vérifier les résultats de même test sont égaux ou non, si non il faut refaire la même test pour la troisième fois.

5. Diagramme de flot de donnée du traitement «distribution» :

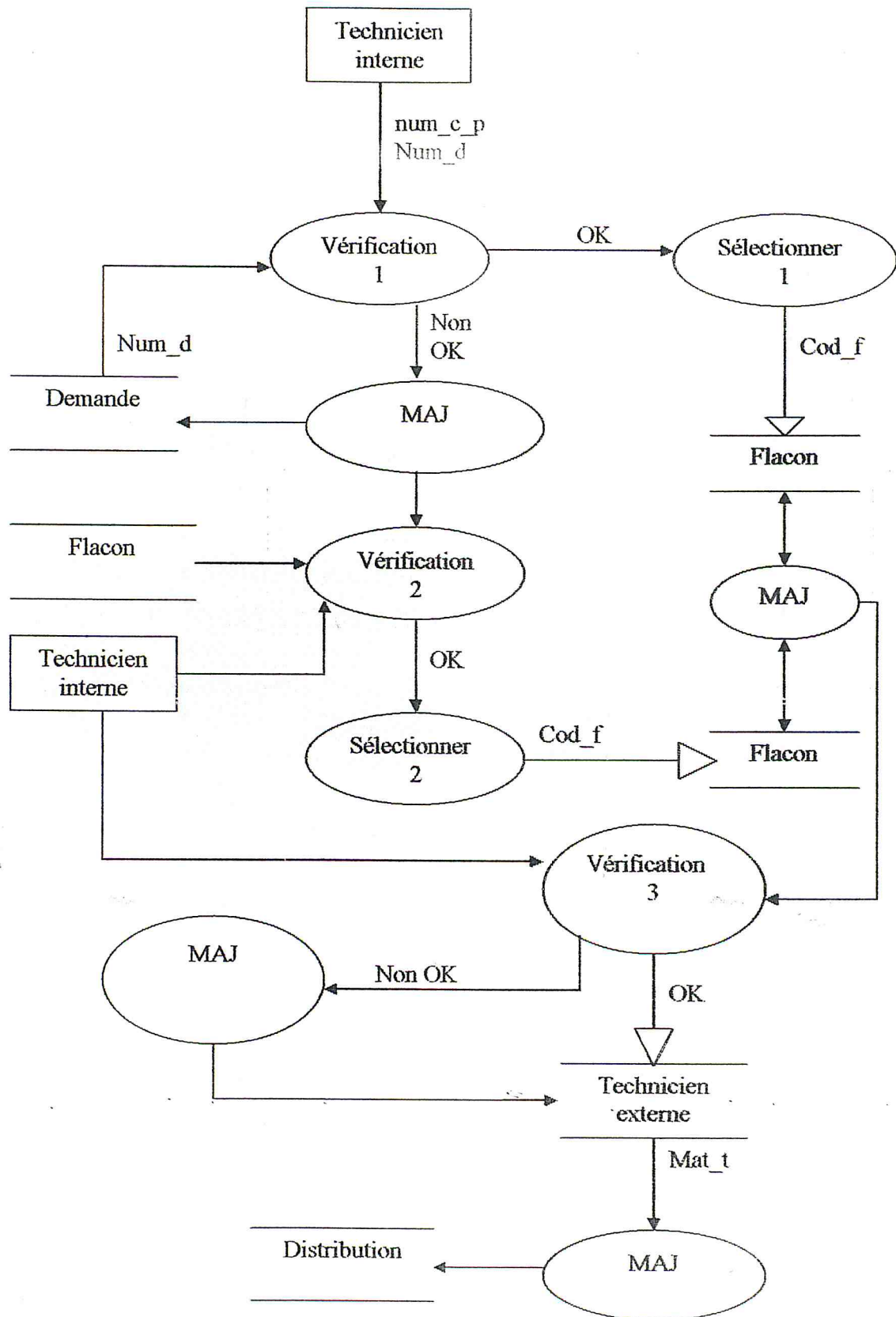


Figure 49 : DFD de distribution.

Description :

- **Vérification1** : vérifier si la demande existe déjà ou non.
- **Vérification 2** : vérifier s'ils existe des flacons non demandés ou pas.
- **Vérification 3** : vérifier si le technicien externe demandeur existe déjà ou non.
- **MAJ** : mise à jour des flacons à distribuées.
- **Créer un objet technicien externe** : créer un nouveau objet dans le fichier technicien externe.
- **Créer un objet distribution** : créer un nouveau objet dans le fichier distribution.
- **Sélectionner** : sélectionner des objets dans un fichier.
- **Créer un objet demande**: vérifier les résultats de même test sont égaux ou non, si non il faut refaire la même test pour la troisième fois.

3. Conception de base de données :**3.1. Le passage de modèle objet vers le modèle relationnel :**

On appliquant les règles de passage de modèle objet vers le modèle relationnel pour aboutit au schéma des tables relationnelles constituant la base de données.

- ❖ Toutes les classes de modèle objet seront traduites en relations telles que leurs attributs traduisent les attributs des classes (**Règle 1**).

Exemple :

Donneur (mat_dn , nom_dn, pnom_dn , adrs_pe, adrs_po, vil_pe, vil_po, w_pe, w_po, tel_pe, tel_po, dt_n_dn , l_n_dn , sx_dn , et_c_dn, tp_dn , phtype , gp_s, etat_d)

- ❖ Toutes les associations de type plusieurs à plusieurs (**N vers N**) seront traduites en relations, la clé primaire de la relation est la concaténation des clés des classes qui les reliés, et ajoutant la ou les propriétés de l'association comme des attributs de la relation (le cas d'une classe association) (**Règle 5**).

Exemple :

L'association « réaliser » entre les classes (test, technicien, flacon) se traduit comme suit :

Réaliser (cod_fl , mat_th , cod_tst , rst_f , obsr_f)

- ❖ Toutes les associations de type un à plusieurs (**1 vers N**) seront traduites en portant dans la relation fille de multiplicité N la clé primaire de la relation mère (l'autre) (**Règle 4**).

Exemple :

L'association « grouper » entre les classes « patient » et « groupe sanguine » se traduit comme suit : la clé de la relation de la classe « groupe sanguine » (**gp_s**) sera immigré dans la relation de la classe « patient ».

Patient (mat_pt , nom_pt, nom_j_pt, pnom_pt, dt_n_pt, adrs_pt, tel_pt, sx_pt, n_sv, mat, n_dos, gp_s, phtype).

- ❖ Toutes les associations de type un à plusieurs (**1 vers 1**) seront traduites en Portant dans une relation parmi les deux la clé primaire de l'autre relation. (**Règle 2**).

Exemple :

L'association « concerne » entre les classes « demande » et « distribution » se traduit comme suit : la clé de la relation de la classe « demande » (**n_dm**) sera immigré dans la relation de la classe « distribution ».

Distribution (n_dtb , mat_th_r , date_d , n_dm).

- ❖ Les relations d'héritage dans le modèle objet ne sont pas interprétées dans le modèle relationnel (Règle 6), néanmoins il existe deux solutions :
 - Toutes les classes fils sont soit regroupées dans une même relation et des attributs sont rajoutées pour arriver à distinguer entre les fils.

Exemple :

Les classes (donneur bénévole, donneur non bénévole) seront regroupées dans la relation de la classe « donneur » tel que le discriminateur « type » sera représenté comme attribut « tp_dn » dans la relation « donneur » pour séparé entre les objets des deux classes regroupées.

Donneur (mat_dn , nom_dn , adrs_pe, adrs_po, vil_pe, vil_po, w_pe, w_po, tel_pe, tel_po, pnomo_dn , dt_n_dn , l_n_dn , sx_dn , et_c_dn, tp_dn , phtype , gp_s, etat_d)

- Soit chaque fils sera interprété par une relation, et dans ce cas il faut prendre en compte les attributs et relations que la classe mère possède.

Exemple :

On peut interprété les classes fils (technicien externe, technicien interne) comme des relations et les attributs de la super classe seront hérité par les relations des classes fils.

Technicien interne (mat_th , nom_th , pnomo_th , adrs_th ,tel_th)

Technicien externe (mat_th , nom_th , pnomo_th , adrs_th ,tel_th , l_tr_th)

3.2. Liste des relations de modèle relationnel:

Produit sanguin (COD_PD, DS_PD, DL_PD)

Banque (N_BQ)

Paramètre (COD_PTRE , DS_PTRE)

Technicien (MAT_TH, NOM_TH, PNOM_TH, ADRS_TH, TEL_TH, TYPE, L_TR_TH, N_CP_TH)

Test (COD_TST, DS_TST, N_CP_TH)

Hôpital (N_HP, DS_HP)

Service (N_SV, DS_SV, N_HP)

Groupe sanguin (GP_S)

Phénotype (PHTYPE)

Maladie (COD_ML , DS_ML)

Porte_p (MAT_PT, COD_ML , OBSR_P)

Porte_d (MAT_DN , COD_ML , OBSR_D)

Donneur (MAT_DN , NOM_DN , ADRS_PE, ADRS_PO, VIL_PE, VIL_PO, W_PE, W_PO, TEL_PE, TEL_PO, PNOM_DN , DT_N_DN , L_N_DN , SX_DN , ET_C_DN, TP_DN , PHTYPE , GP_S, ETAT_D).

Patient (MAT_PT, NOM_PT, NOM_J_PT, PNOM_PT, DT_N_PT, ADRS_PT, TEL_PT, SX_PT, N_SV, MAT, N_DOS, GP_S, PHTYPE).

Médecin (MAT_MD, NOM_MD, PNOM_MD, ADRS_MD, TEL_MD, TYPE, L_TR_MD)

Distribution (N_DTB , MAT_TH_R , DATE_D , N_DM).

Consulter (MAT_DN, MAT_MD, DATE, POIDS, OBSR_C)

Flacon (COD_FL , CC_FL , TP_FL , COD_PD , N_DM , GP_S , PHTYPE , N_DTB, MAT_DN , TP_DON, COD_PTRE , N_PV, DATE_D, N_BQ, DATE_PR).

Concerne (N_DM, COD_PD, QTE_D)

Anticorps (ATCP)

Date (DATE)

Sépare (COD_FL_P , COD_FL_F)

Utilise (COD_TST, COD_PTRE)

Réaliser_t (N_TB, COD_TST, MAT_TH, RST_T, OBSR_T)

Réaliser_f (COD_FL, COD_TST, MAT_TH, RST_F, OBSR_F)

Demande (N_DM, GP_S, MAT_PT, PHTYPE, MAT_MD, DATE, DATE_S)

Tube (N_TB, DATE, MAT_PT)

Tirer (MAT_TH, COD_PTRE, DATE, QTE)

Avoir (COD_PTRE, DATE, QTE, DATE_P)

3.3. Liste des tables de base de données :

Produit sanguin :

Nom de champs	Type	Taille	Clé primaire
COD_PD	A	04	*
DS_PD	A	30	
DL_PD	A	10	

Banque :

Nom de champs	Type	Taille	Clé primaire
N_BQ	A	03	*

Paramètre :

Nom de champs	Type	Taille	Clé primaire
COD_PTRE	A	03	*
DS_PTRE	A	30	

Technicien :

Nom de champs	Type	Taille	Clé primaire
MAT_TH	A	07	*
NOM_TH	A	20	
PNOM_TH	A	20	
ADRS_TH	A	50	
TEL_TH	A	20	
L_TR_TH	A	1	
N_CP_TH	A	20	

Tube :

Nom de champs	Type	Taille	Clé primaire
N_TB	A	07	*
MAT_PT	A	12	
DATE_P	D	08	

Hôpital :

Nom de champs	Type	Taille	Clé primaire
N_HP	A	07	*
DS_HP	A	30	

Service :

Nom de champs	Type	Taille	Clé primaire
N_SV	A	04	*
DS_SV	A	30	
N_HP	A	07	

Groupe sanguin :

Nom de champs	Type	Taille	Clé primaire
GP_S	A	10	*

Médecin :

Nom de champs	Type	Taille	Clé primaire
MAT_MD	A	07	*
NOM_MD	A	20	
PNOM_MD	A	20	
ADRS_MD	A	50	
TEL_MD	A	20	
L_TR_MD	A	1	

Date :

Nom de champs	Type	Taille	Clé primaire
DATE	D	08	*

Maladie :

Nom de champs	Type	Taille	Clé primaire
COD_ML	A	04	*
DS_ML	A	30	

Porte_p:

Nom de champs	Type	Taille	Clé primaire
MAT_PT	A	12	*
COD_ML	A	04	*
OBSR_P	A	50	

Porte_d :

Nom de champs	Type	Taille	Clé primaire
MAT_DN	A	10	*
COD_ML	A	04	*
OBSR_D	A	50	

Séparé :

Nom de champs	Type	Taille	Clé primaire
COD_FL_P	A	13	*
COD_FL_F	A	13	*

Phénotype :

Nom de champs	Type	Taille	Clé primaire
PHTYPE	A	30	*

Donneur :

Nom de champs	Type	Taille	Clé primaire
MAT_DN	A	10	*
NOM_DN	A	20	
PNOM_DN	A	20	
DT_N_DN	D	08	
L_N_DN	A	20	
SX_DN	A	1	
ET_C_DN	A	20	
TP_DN	A	20	
PHTYPE	A	30	
GP_S	A	10	
ADRS_PE	A	50	
ADRS_PO	A	50	
VIL_PE	A	20	
VIL_PO	A	20	
W_PE	A	20	
W_PO	A	20	
TEL_PE	A	20	
TEL_PO	A	20	

Demande :

Nom de champs	Type	Taille	Clé primaire
N_DM	A	08	*
GP_S	A	10	
MAT_PT	A	12	
PHTYPE	A	30	
MAT_MD	A	07	
DATE_D	D	08	
DATE_S	D	08	

Utilise :

Nom de champs	Type	Taille	Clé primaire
COD_PTRE	A	03	*
COD_TST	A	03	*

Patient :

Nom de champs	Type	Taille	Clé primaire
MAT_PT	A	12	*
N_SV	A	04	
MAT	A	20	
N_DOS	A	20	
GP_S	A	10	
PHTYPE	A	30	
NOM_PT	A	20	
PNOM_PT	A	20	
DT_N_PT	D	08	
ADRS_PT	A	50	
TEL_PT	A	20	
SX_PT	A	1	
NOM_J_PT	A	20	

Distribution :

Nom de champs	Type	Taille	Clé primaire
N_DTB	A	08	*
MAT_TH_R	A	07	
DATE_D	D	08	
N_DM	A	08	

Concerne :

Nom de champs	Type	Taille	Clé primaire
N_DM	A	08	*
COD_PD	A	04	*
QTE_D	A	06	

Avoir :

Nom de champs	Type	Taille	Clé primaire
COD_Pd	A	04	*
DATE_A	D	08	*
DATE_P	D	08	
QTE	N	03	

Flacon :

Nom de champs	Type	Taille	Clé primaire
COD_FL	A	13	*
CC_FL	N	03	
N_DM	A	08	
GP_S	A	10	
PHTYPE	A	30	
N_DTB	A	08	
MAT_DN	A	10	
TP_DON	A	20	
DATE_D	D	08	
N_BQ	A	03	
TP_FL	A	02	
COD_PD	A	04	

Réaliser_f :

Nom de champs	Type	Taille	Clé primaire
COD_FL	A	13	*
COD_TST	A	03	*
MAT_TH	A	07	*
RST_F	A	20	
OBSR_F	A	50	

Teste :

Nom de champs	Type	Taille	Clé primaire
COD_TST	A	03	*
DS_TST	A	30	

Réaliser_t:

Nom de champs	Type	Taille	Clé primaire
N_TB	A	07	*
COD_TST	A	03	*
MAT_TH	A	07	*
RST_T	A	20	
OBSR_T	A	50	

Tirer :

Nom de champs	Type	Taille	Clé primaire
COD_P	A	03	*
MAT_TH	A	07	*
DATE	A	08	*
QTE_R	N	03	

Consulter :

Nom de champs	Type	Taille	Clé primaire
MAT_DN	A	10	*
MAT_MD	A	07	*
DATE_C	D	08	*
TA	A	05	
POIDS	N	05	
OBSR_C	A	50	

4. Conclusion :

Dans ce chapitre nous avons présenté les différents diagrammes de OMT objet, dynamique et fonctionnel, que nous avons utilisés lors de l'analyse et la conception du système. Puis après avoir identifié, progressivement, toutes les classes du domaine nous avons fait le passage, inévitable, vers le modèle relationnel.

Durant toute la démarche ce sont les cas d'utilisation qui nous ont guidé dans le processus d'implémentation.

CHAPITRE IV

REALISATION & MISE EN OEUVRE

1. Introduction :

Dans ce chapitre, nous présentons l'outil mis en œuvre dans le cadre de notre projet. Cet outil permet de saisir, sauvegarder, consulter, supprimer et modifier des données de CRTS à partir d'une base de données implémentée sous Oracle.

2. Environnement de développement :

Notre logiciel a été réalisé avec **JAVA 1.5** sous l'environnement **Windows XP**. Développer une application Windows avec JAVA consiste à concevoir l'interface utilisateur de manière particulièrement visuelle et interactive, spécifier de manière tout aussi interactive les caractéristiques, appelées propriétés, des divers composants (fenêtres, boutons de commande, zones d'édition, etc...).

3. Présentation de système de gestion de base de données utilisé:

Pour la création de notre base de données nous avons opté pour le SGBD Oracle version 8.1.5.0 vues les nombreuses fonctionnalités qu'il assure.

Ces fonctionnalités sont résumés en :

- La définition et la manipulation des données.
- La cohérence des données.
- La confidentialité des données.
- L'intégrité des données.
- La sauvegarde et la restauration des données.
- Offre des outils de contrôle d'accès et de sécurité pouvant gérer des logiques d'accès de données très complexes.
- La gestion des accès concurrents de plusieurs utilisateurs exploitant différentes applications accédant simultanément au même données..
- Client/serveur: Oracle permet de séparer le serveur de données et les applications clientes.
- Gestion des bases de données distribuées : Oracle permet de gérer une base de données distribuées sur différents sites du réseau comme s'il s'agissait d'une base de données unique.

Les étapes :

- ✓ Nous avons crée la base de donnée à travers l'interface graphique d'Oracle 8.i, que l'on a nommée « CRTS ».
- ✓ Puis nous avons passé à la création de toutes les tables de cette base de données, et on introduit les champs de chaque table.
- ✓ Ensuite, nous avons passé à l'étape suivant qui consiste à la création des relations entre les tables, les contraintes, et les indexes.

4. Présentation de langage de programmation utilisé:

Pour la programmation et la conception des interfaces de notre application on a utilisé le langage JAVA version 1.5, vues les nombreuses caractéristiques.

Caractéristiques de base de langage JAVA : Java est un langage :

1. simple,
2. orienté objet,
3. distribué,
4. interprété,
5. robuste, et sûr
6. possédant une riche API : le JDK.
7. indépendant de l'architecture (portable),
8. mobile,

9. performant (pas pour toutes les applications, mais ça va venir...)
10. multithreadé,
11. dynamique,
12. exécuté non pas au travers d'un interpréteur, mais au travers de l'émulation d'une machine virtuelle, ou d'une puce JavaOS.
13. véritable explosion depuis son lancement : nombreux produits commerciaux : IDEs, APIs spécialisées, etc...
14. Java propose un ensemble de classes pour la programmation réseau. Ces classes sont regroupées dans le paquetage *java.net*. On peut programmer très facilement des architectures client/serveur en mode TCP ou UDP à l'aide des classes proposées.

5. Sécurité :

Pour ne pas courir le risque d'une perte inattentionnées des informations par erreur de commande ou de manipulation du matériel, nous proposons la nécessité de prendre des mesures formelles et strictes vis à vis de la sécurité de notre système afin d'éviter une perte de temps et de capital.

Afin d'assurer l'intégrité, la fiabilité et la confidentialité des informations, on propose quelques mesures de sécurité qui doivent être prises en considération.

5.1. Sécurité de la base de données : La sécurité de la base de données recouvre l'ensemble des opérations qui permettent à certains utilisateurs ou groupes d'utilisateurs d'accéder à des données en consultation, en modification ou en édition.

- L'attribution de mot de passe aux différents utilisateurs leur définit le droit de consulter, de modifier et d'éditer la vue correspondante.
- Pour la protection de la base contre toute forme de destruction (incendie, vol, perte, erreur, de logiciel, défaillance du matériel...) des sauvegardes périodiques de la base de données en plusieurs copies placées dans des endroits différents.
- Conserver durant une certaine durée les fichiers manuels et les documents de la saisie pour les cas extrêmes (perte d'informations).
- Pour assurer une cohérence et une protection des données, Oracle dispose d'un ensemble de commandes pour verrouiller un enregistrement, une table ou toute la base de données.
- Sauvegarde périodique des données.
- Après toute saisie, archivage des fiches de donneurs et des demandes de sang pour pouvoir retrouver l'information en cas d'erreurs graves.
- Toute saisie doit être correcte et complète.

Pour une question de confidentialité des informations, on doit respecter l'attribution purement secrète des mots de passe pour l'accès aux informations.

5.2. Sécurité de l'application : Sauvegarde du logiciel de l'application sur plusieurs copies.

- En cas de panne du serveur, prévoir un serveur de secours pour le remplacer.
- Pour la protection du réseau et pour la prévention contre tout virus, prévoir l'installation d'un antivirus réseau.

5.3. Sécurité du matériel : Des mesures de sécurité sont prises en charge par le CRTS pour le bon fonctionnement du matériel.

- Relier l'équipement informatique à un onduleur pour sa protection contre les perturbations de la source d'alimentation (coupures de courant et les courts circuits).

- Un système de climatisation pour éviter l'échauffement du matériel.
- Prévoir un système d'extinction d'incendie.

6. Techniques d'archivages et de classifications proposées : pour éviter les problèmes de perte des documents on a proposé les techniques de classement suivantes :

- On a proposé d'utiliser des armoires protégées et fermer.
- Pour facilité, on a proposé de classer les fiches des donneurs selon le groupe sanguin et le nom de donneur.
- ❖ Classement des fiches des donneurs se fait en niveau de secrétariat comme suit :
Le classement des fiches des donneurs de même groupe sanguin se fait dans une étagère et doit être rangé avec ordre alphabétique des noms des donneurs.

Dans le cas où le donneur est exclu définitivement au don sa fiche doit être archivé selon la même méthode précédente dans une armoire spéciale pour les donneurs exclus.

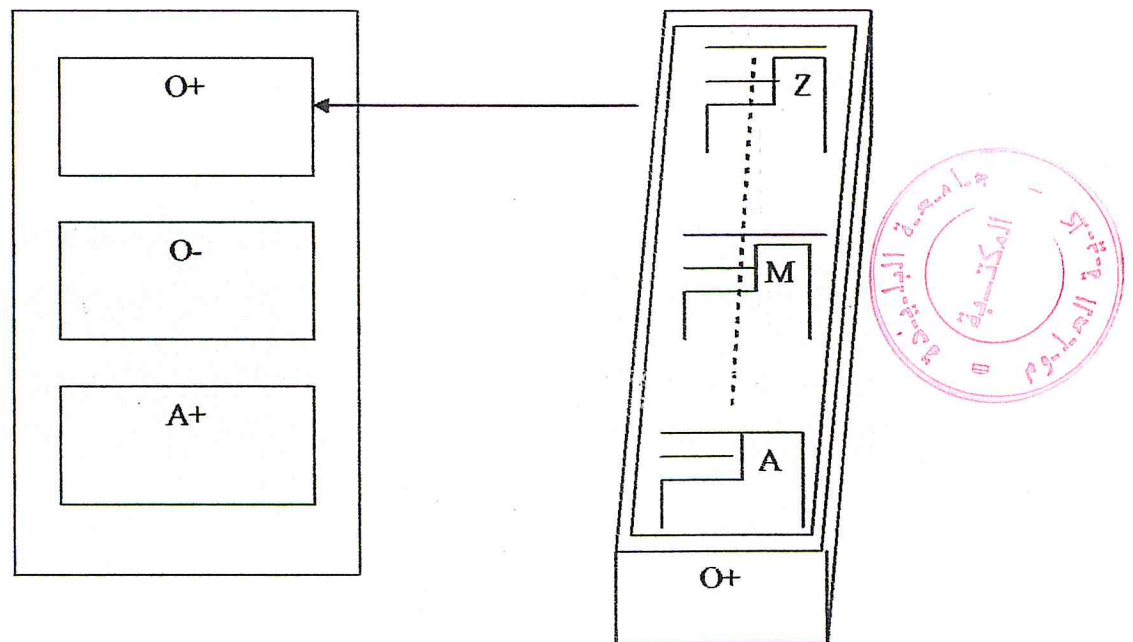


Figure 50 : Classement des fiches des donneurs.

- ❖ Archivage des demandes de sang et des fiches de distribution de sang se fait en niveau de service de banque du sang comme suit :
 - Les hôpitaux sont classés par leur wilaya.
 - Chaque demande de sang doit être fixée avec sa fiche de distribution.
 - L'archivage des demandes de sang avec la fiche de distribution de même hôpital dans un étagère et doit être rangé avec ordre de date de distribution et les hôpitaux sont rangés dans l'ordre alphabétique de leur nom.

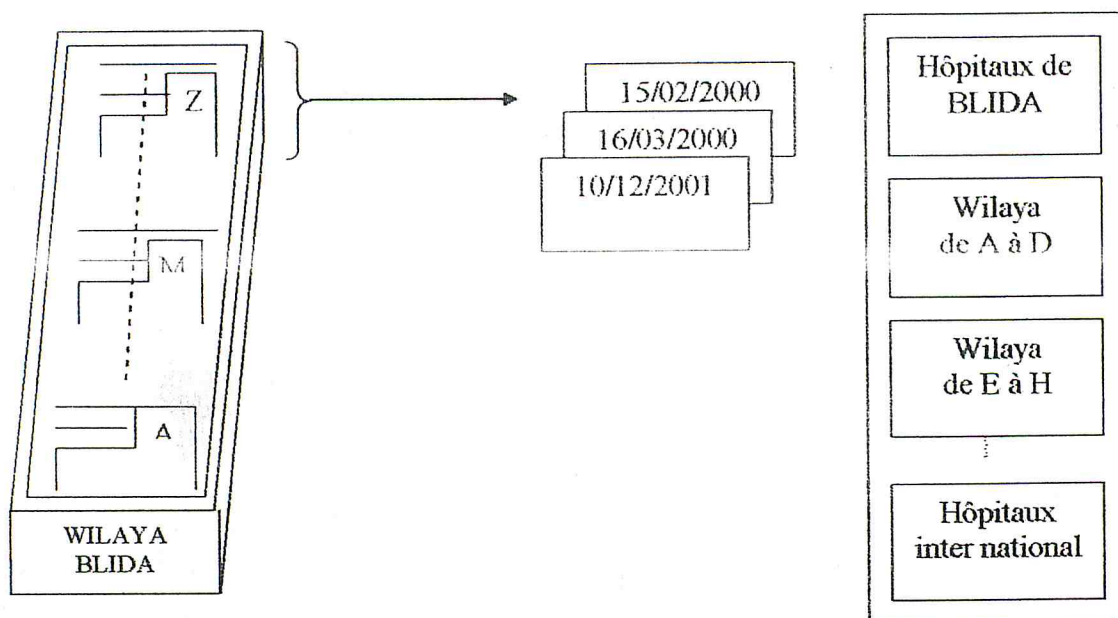


Figure 51 : Archivage des demandes de sang et fiches de distribution

7. Présentation du logiciel :

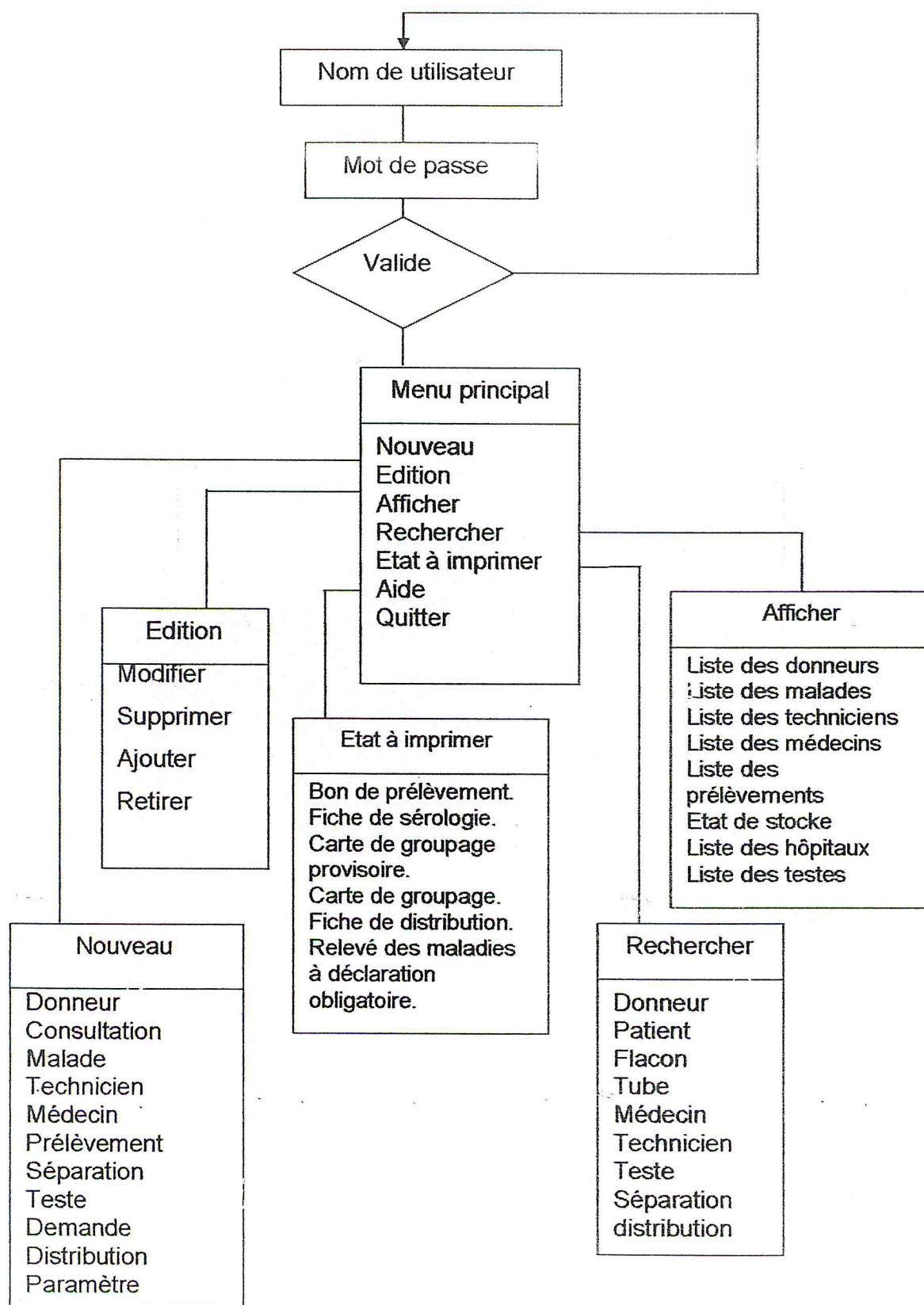
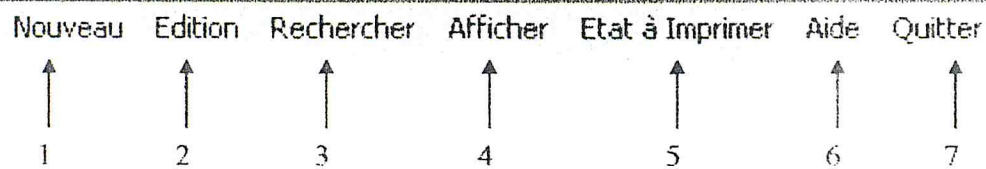


Figure 52 : Schéma global de fonctionnement de logiciel CRTS

Description de menu de CRTS :



1 : Le module « Nouveau » pour saisir les différents objets de la base de données comme des donneurs, patients,...

2 : Le module « Edition » pour faire des mises à jours comme des suppression ou des modifications des informations de donneur, ajouter ou retirer des paramètres de stocke,...

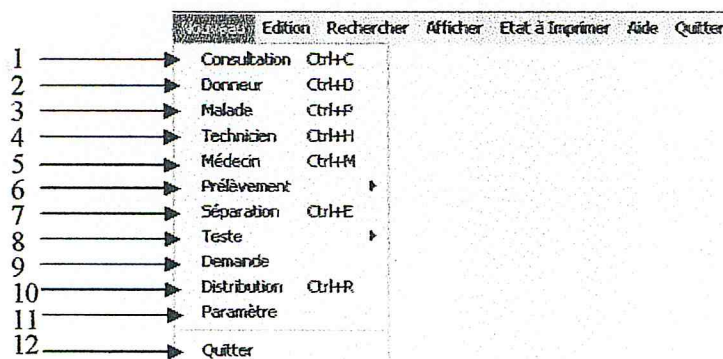
3 : Le module « Rechercher » pour faire des recherches dans la base de données comme recherche des donneurs, patients, flacons,...

4 : Le module « Afficher » pour afficher les listes des donneurs, des patients,...

5 : Le module « Etat à imprimer » pour imprimer les différents documents utiliser dans le centre de transfusion sanguine comme la carte de groupage, fiche de prélèvement,...

6 : Le module « Aide » pour consulter l'aide de CRTS, contacter les concepteurs de logiciel.

7 : Le module « Quitter » pour fermer le logiciel.



1 : Consultation : pour saisir des nouveau consultation des donneurs.

2 : Donneur : pour saisir des nouveaux donneurs.

3 : Malade : pour saisir des nouveaux patients demandeurs de sang ou demandeurs d'analyses.

4 : Technicien : pour saisir les nouveaux techniciens.

5 : Médecin : pour saisir les nouveaux médecins.

6 : Prélèvement : pour saisir les flacons et les tubes du sang.

7 : Séparation : pour saisir les flacons fils séparé de flacon père selon le produit sanguin.

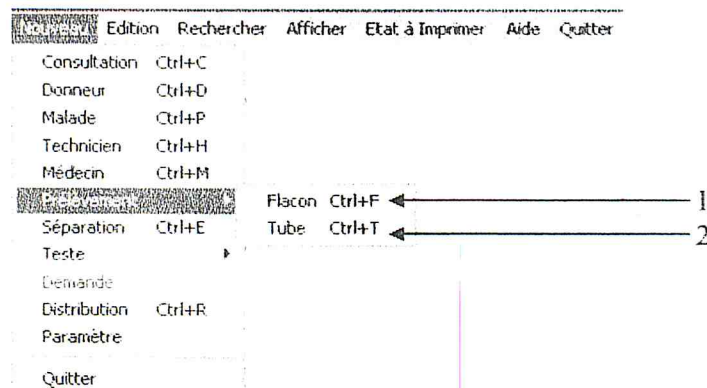
8 : Teste : pour saisir les résultats des analyses de sérologie ainsi que de groupage.

9 : Demande : pour saisir les demandes de sang.

10 : Distribution : pour saisir les distributions de sang réalisé.

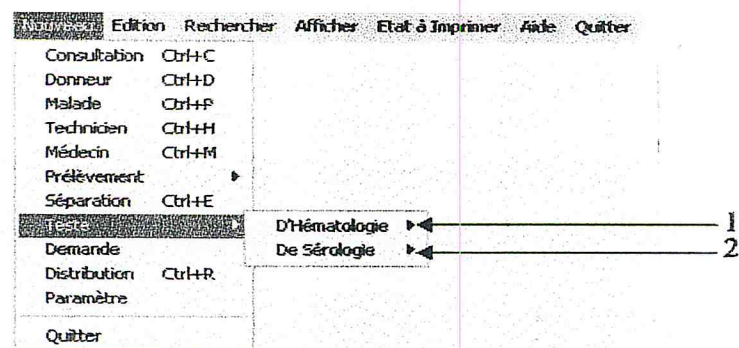
11 : Paramètre : pour saisir des nouveaux paramètres utiliser dans les analyses de sérologie et de groupage.

12 : Quitter : pour fermer l'application.



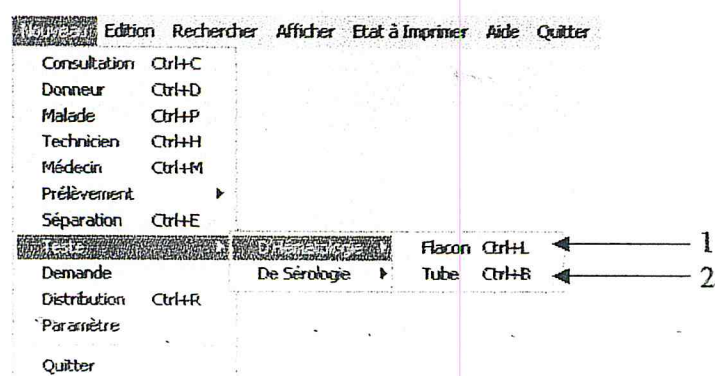
1 : Flacon : pour saisir les nouveaux flacons prélevé par des donneurs.

2 : Tube : pour saisir les nouveaux tubes de sang prélevé par des patients qui demande les analyses.



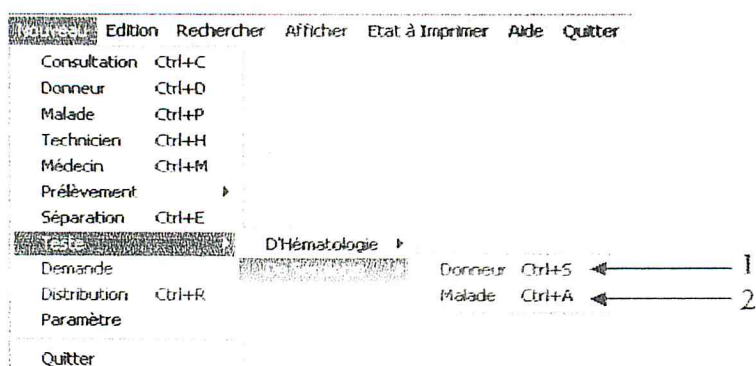
1 : Teste d'hématologie : pour saisir les résultats de groupage de sang.

2 : Teste de sérologie : pour saisir les résultats de sérologie.

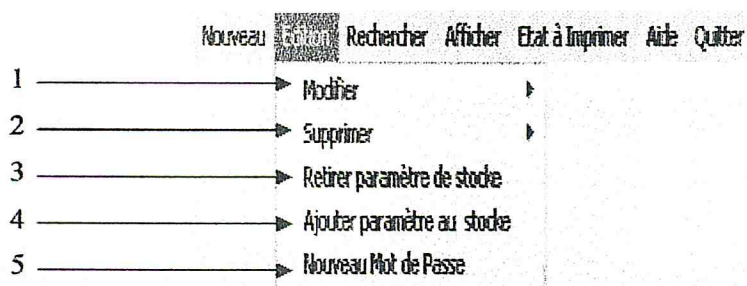


1 : Flacon : pour saisir les résultats des analyses de groupage des flacons.

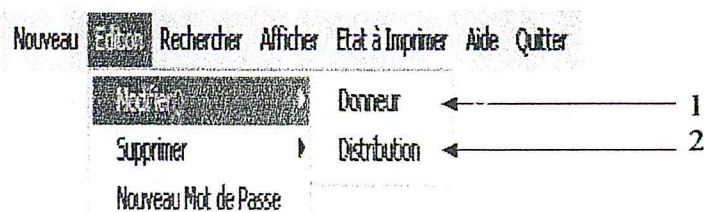
2 : Tube : pour saisir les résultats des analyses de groupage des tubes.



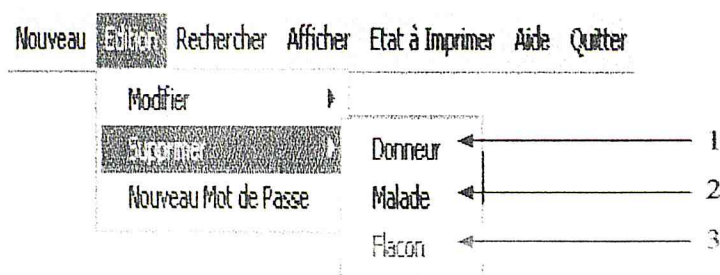
- 1 : Donneur : pour saisir les résultats de sérologie des donneurs.
 2 : Malade : pour saisir les résultats de sérologie des patients.



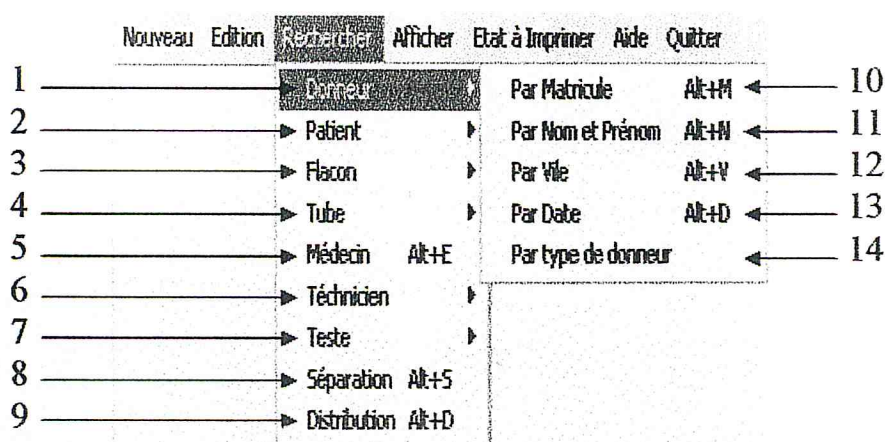
- 1 : Modifier : pour faire des modifications sur des informations des donneurs et des distributions.
 2 : Supprimer : pour supprimer des donneurs, patients,...
 3 : Retirer paramètre de stocke : pour saisir des quantités des paramètres sorties de stocke.
 4 : Ajouter paramètre au stocke : pour saisir les quantités des paramètres entrés au stocke.
 5 : Nouveau mot de passe : pour changer le mot de passe d'entrée à l'application.



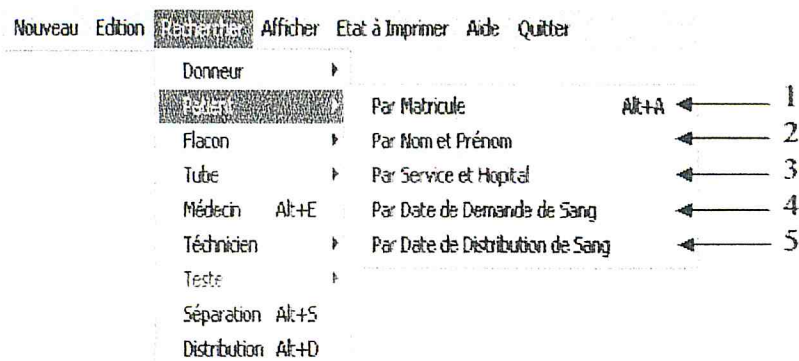
- 1 : Donneur : pour modifier des informations de donneur.
 2 : Distribution : pour modifier des informations de distribution.



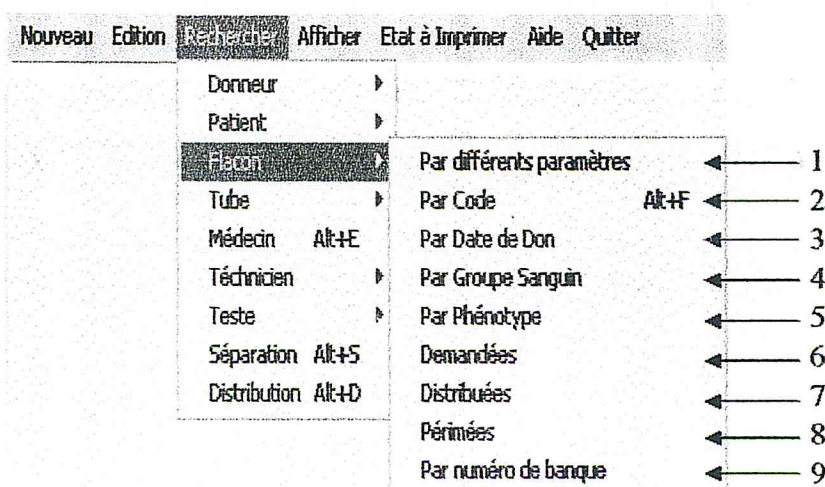
- 1 : Donneur : pour supprimer les donneurs.
 2 : Malade : pour supprimer les patients.
 3 : Flacon : pour supprimer les flacons.



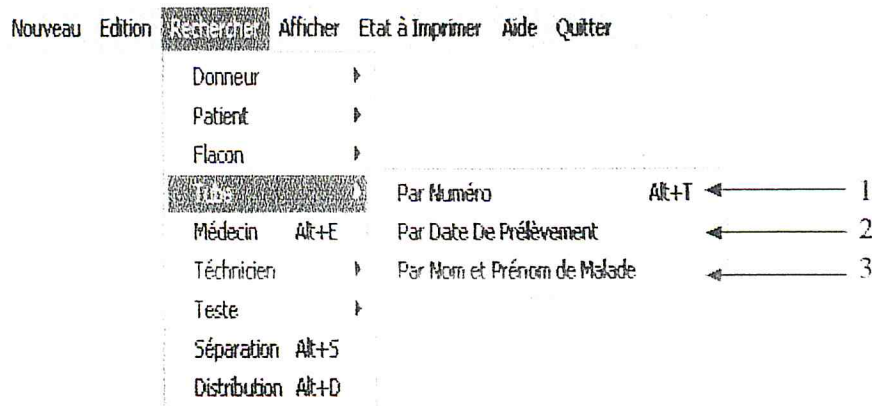
- 1 : Donneur : rechercher les donneurs.
 2 : Patient : rechercher les patients.
 3 : Flacon : rechercher les flacons.
 4 : Tube : rechercher les tubes.
 5 : Médecin : rechercher les médecins.
 6 : Technicien : rechercher les techniciens.
 7 : Teste : rechercher les résultats des testes.
 8 : Séparation : rechercher les séparations des flacons.
 9 : Distribution : rechercher les distributions de sang.
 10 : Par matricule : rechercher les donneurs par leur matricule.
 11 : Par nom et prénom : rechercher les donneurs par leur nom et prénom.
 12 : Par ville : rechercher les donneurs par leur ville de résidence.
 13 : Par date : rechercher les donneurs par la date de prélèvement.
 14 : Par type de donneur : rechercher les donneurs par leur type.



- 1 : Par matricule : rechercher les patients par leur matricule.
 2 : Par nom et prénom : rechercher les patients par leur nom et prénom.
 3 : Par service et hôpital : rechercher les patients par service et hôpital où il soigne.
 4 : Par date de demande de sang : rechercher les patients par la date de demande de sang.
 5 : Par date de distribution de sang : rechercher les patients par la date de distribution de sang.



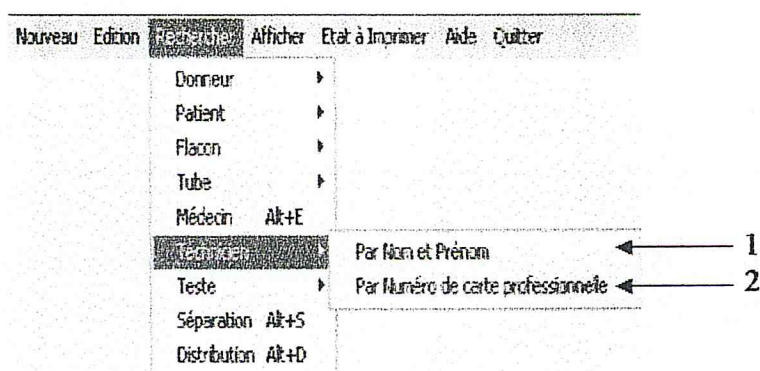
- 1 : Par différents paramètres : rechercher les flacons selon plusieurs paramètres.
 2 : Par code : rechercher les flacons par leur code.
 3 : Par date de don : rechercher les flacons par leur date de prélèvement.
 4 : Par groupe sanguin : rechercher les flacons par leur groupe sanguin.
 5 : Par phénotype : rechercher les flacons par leur phénotype.
 6 : Demandé : rechercher les flacons demandés selon le numéro de demande de sang.
 7 : Distribué : rechercher les flacons distribués selon de numéro de distribution.
 8 : Périmé : rechercher les flacons périmés.
 9 : Par numéro de banque : rechercher les flacons selon le numéro de banque.



1 : Par numéro : rechercher les tubes par leur numéro.

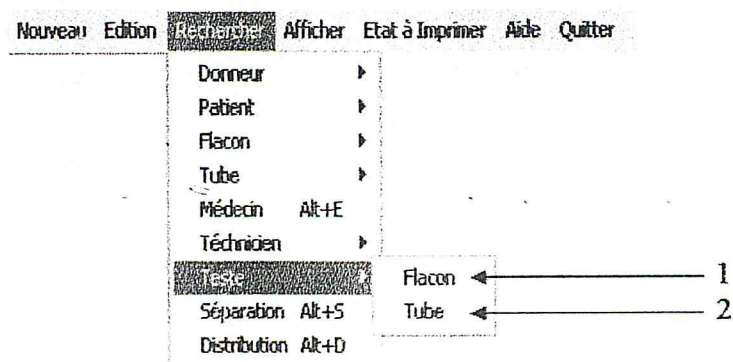
2 : Par date de prélèvement : rechercher les tubes par leur date de prélèvement.

3 : Par nom et prénom de malade : rechercher les tubes par nom et prénom de patient qui prélève le tube de sang.



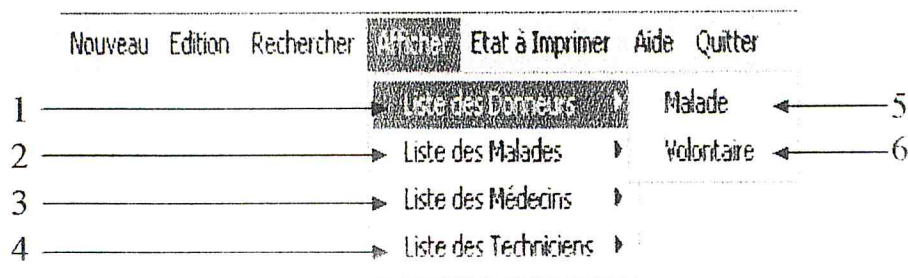
1 : Par nom et prénom : rechercher technicien par leur nom et prénom.

2 : Par numéro de carte professionnelle : rechercher technicien par numéro de leur carte professionnelle.



1 : Flacon : rechercher les résultats des testes d'un flacon selon leur numéro.

2 : Tube : rechercher les résultats des testes d'un tube selon leur numéro.



- 1 : Liste des donneurs : pour afficher ou imprimer la liste des donneurs.
 2 : Liste des malades : pour afficher ou imprimer la liste des patients.
 3 : Liste des médecin : pour afficher ou imprimer la liste des médecins.
 4 : Liste des techniciens : pour afficher ou imprimer la liste des techniciens.
 5 : Malade : pour afficher la liste des donneurs malades.
 6 : Volontaire : pour afficher la liste des donneurs bénévoles.



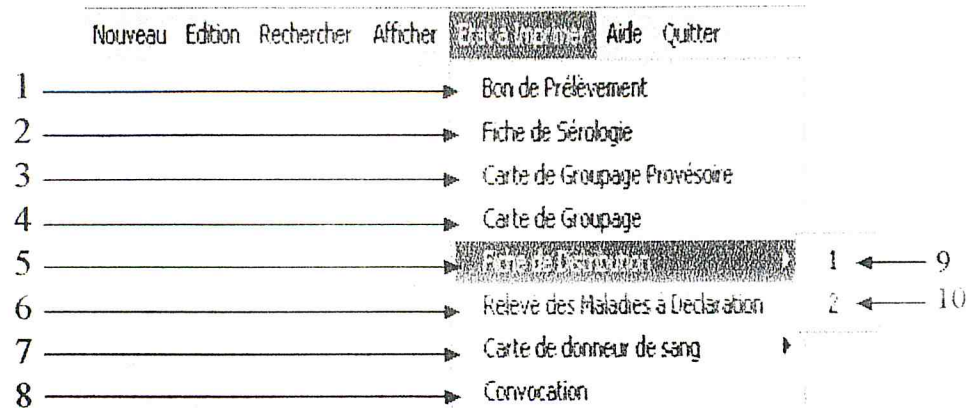
- 1 : A déclarer : pour afficher la liste des donneurs malades pour les déclarer au service de prévention.



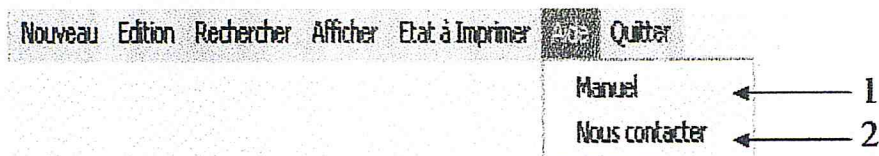
- 1 : Interne : pour afficher la liste des médecins de centre de transfusion sanguine.
 2 : Externe : pour afficher la liste des médecins qui demandent le sang pour les patients.



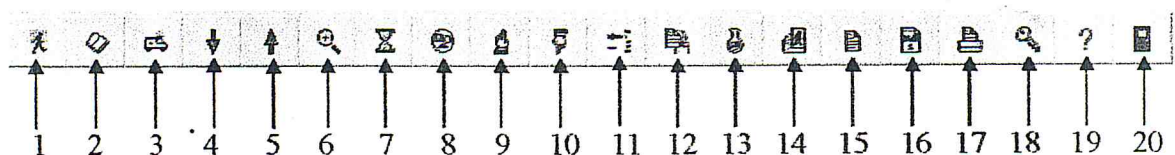
- 1 : Interne : pour afficher la liste des techniciens de centre de transfusion sanguine.
 2 : Externe : pour afficher la liste des techniciens qui transmettent le sang de centre vers les hôpitaux.



- 1 : Bon de prélèvement : pour imprimer un bon de prélèvement vide.
 2 : Fiche de sérologie : pour imprimer un fiche de sérologie vide.
 3 : Carte de groupage provisoire : pour imprimer une carte de groupage provisoire vide.
 4 : Carte de groupage : pour imprimer une carte de groupage vide.
 5 : Fiche de distribution (1,2): pour imprimer les deux faces d'une fiche de distribution de sang vide. 9-représente de recto de la fiche et 10-représente le verso de la fiche.
 6 : Relevé des maladies à déclaration : pour imprimer le relevé des maladies à déclaration obligatoire.
 7 : Carte de donneur de sang : pour imprimer une carte de donneur de sang volontaire avec les deux faces recto (1) et verso (2).
 8 : Convocation : pour imprimer des convocations pour les donneurs.



- 1 : Manuel : pour consulter l'aide manuel de logiciel.
 2 : Nous contacter : pour consulter des information de concepteurs pour les contacter.



- 1 : Raccourci a la fenêtre des nouveaux donneurs.
 2 : Raccourci a la fenêtre des nouvelles consultations.
 3 : Raccourci a la fenêtre des nouveaux patients.
 4 : Raccourci a la fenêtre des nouveaux prélèvements des flacons.
 5 : Raccourci a la fenêtre des nouvelles distributions de sang.
 6 : Raccourci a la fenêtre de recherche des donneurs.
 7 : Raccourci a la fenêtre de saisir des résultats des testes de sérologie des donneurs.
 8 : Raccourci a la fenêtre de saisir des résultats de groupage des donneurs.
 9 : Raccourci a la fenêtre de saisir des quantités sorties des paramètres.
 10 : Raccourci a la fenêtre de saisir des quantités entrées des paramètres.
 11 : Raccourci a la fenêtre de saisir des séparations des flacons de sang.
 12 : Raccourci a la fenêtre de saisir les demandes de sang.

- 13 : Raccourci a la fenêtre de saisir les prélèvements des tubes de sang.
- 14 : Raccourci a la fenêtre de saisir les nouveaux techniciens.
- 15 : Raccourci pour afficher la liste des donneurs de sang volontaires.
- 16 : Raccourci pour afficher et imprimer le relevé des malades a déclaration obligatoire.
- 17 : Raccourci pour afficher et imprimer les convocations des donneurs de sang.
- 18 : Raccourci a la fenêtre de changement de mot de passe.
- 19 : Raccourci a la fenêtre d'aide de CRTS.
- 20 : Raccourci pour fermer l'application.

Description de quelques fenêtres de CRTS :

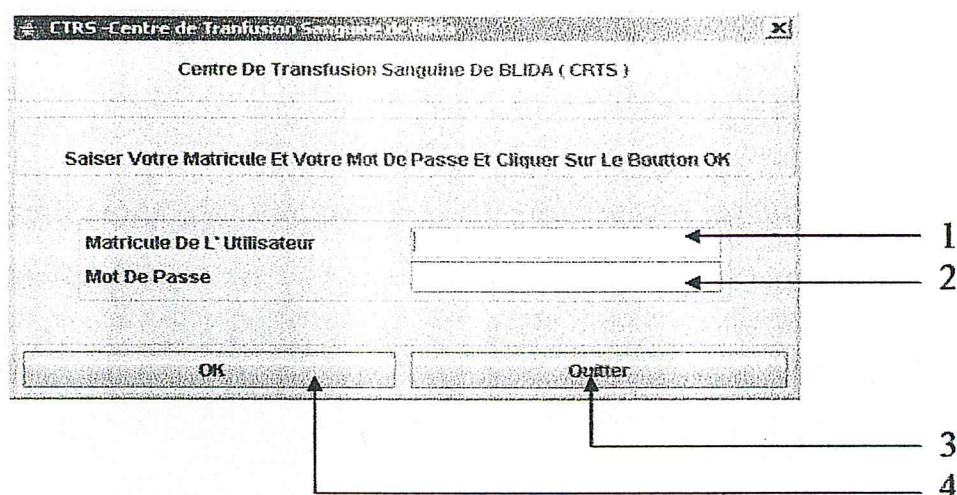


Figure53 : fenêtre pour saisir le mot de passe

- 1 : Champ de saisir pour le matricule de l'utilisateur.
- 2 : Champ de saisir pour le mot de passe de l'utilisateur.
- 3 : Bouton pour valider pour entrer dans le logiciel CRTS.
- 4 : Bouton pour fermer la fenêtre et quitter l'application CRTS.

8. Mise en œuvre :

Pour la mise en œuvre de notre système, nous citons ici quelques points primordiaux :

- ✓ Réorganisation de la structure du service selon les besoins pour la réussite du nouveau système d'information.
- ✓ Formation et familiarisation du personnel du service ainsi que les agents de saisie à l'encontre de la nouvelle méthode de traitement de l'information mise à leur disposition.
- ✓ Implémentation de la base de données.
- ✓ Installation de l'application.
- ✓ Elaboration des jeux d'essais en collaboration avec les utilisateurs afin d'obtenir des résultats satisfaisants et complets.
- ✓ Pour la maintenance de l'application, prévoir un analyste programmeur ou un programmeur expérimenté pour prendre en charge les extensions et les modifications désirées par les utilisateurs.
- ✓ Lancement de l'application.

9. Conclusion :

Dans ce chapitre nous avons présenté les différentes étapes de réalisation de notre système, ainsi que quelques interfaces de notre logiciel et leur description.

En fin on a présenté les techniques qu'on a proposées pour assurer un bon archivage des documents qui évite les pertes des documents et facilite les recherches dans documents archivés.

CONCLUSION GÉNÉRALE

En dépit des contraintes humaines et matérielles que nous avons rencontré tout le long de notre étude, nous avons pu tout de même arriver à analyser la situation existante et porter quelques nouveaux changements afin d'améliorer le système d'information manuel actuellement en vie.

Le travail que nous venons d'achever et qui nous a été confié par le responsable de centre régional de transfusion sanguine de BLIDA (CRTS), consiste à mettre en œuvre un logiciel de gestion de centre de transfusion sanguine.

Ce projet nous a permis d'apprendre une technique de modélisation objets (OMT), d'établir un premier contact avec le SGBD Oracle et quelques outils de développement de langage JAVA.

Notre nouveau système permet :

- La gestion automatique du banque du sang (flacons sortie, flacons entrées).
- Le suivi des fiches des donneurs.
- La gestion du stocke des paramètres de CRTS.
- Le suivi des commandes des patients et des hôpitaux.
- Le suivi des fiches de distribution.
- Le suivi des donneurs qui portent des maladies dangereuses.

Ce qui a motivé l'informatisations du travail de CRTS et de les rendre plus facile à utiliser tout en permettant au personnel de consulter et d'accéder efficacement aux informations autorisés.

Ce travail reste évolutif dans la mesure où CRTS s'engage dans l'informatisation de son système.

BIBLIOGRAPHIE

LIVRES :

- ❖ Programmation réseau avec Java : Eliotte Rusty Harold ; ed O'Reilly ISBN 2-84177-034-6
- ❖ Modélisation et conception orientées objet (J.Rumbaugh 1993).
- ❖ Méthode orientée objet (J. Grahem 1990).
- ❖ OMT modélisation et conception orientées objet (J.Rumbaugh et Al 1997)
- ❖ Oracle 8 (W.David Lockman 2001).

LES MEMOIRES :

- ❖ Conception et réalisation d'un système d'information pour la gestion d'une bibliothèque avec de SGBD Oracle Chikhi Hassiba et Meraga Adila (USTHB 092/2002).
- ❖ Conception et réalisation d'un système d'information pour la gestion d'une bibliothèque basé sur OMT (USTHB 034/2000).
- ❖ Conception et réalisation d'un système d'information de gynécologie obstétrique Noui Mohamed et Badji Mohamed El-yazid (INI 1989)

LES SITES :

- ❖ [http:// www. esil.univ-mrs.fr/staff/prof/henocque/](http://www.esil.univ-mrs.fr/staff/prof/henocque/)
- ❖ [http:// www. Oracle.com](http://www.Oracle.com)
- ❖ <http://www.eckelobjects.com/>
- ❖ <http://www.groups.yahoo.com/group/tij2ndenfrancais/>
- ❖ <http://www.eyrolles.com>.
- ❖ <http://www.JAVA.SUN.com>
- ❖ [http:// www.java.com](http://www.java.com)
- ❖ [http:// www.javaworld.com](http://www.javaworld.com)
- ❖ [http:// www.univ-angers.fr/docs/etudquassi/OMT](http://www.univ-angers.fr/docs/etudquassi/OMT)
- ❖ [http:// www.wikipedia.org/wiki/OMT](http://www.wikipedia.org/wiki/OMT)

ANNEXE

1. Introduction à Oracle :

Oracle est un système de gestion de base de données relationnel (SGBDR), dont voici les plus importantes caractéristiques :

- Gestion des bases de données de plusieurs centaines de gigaoctets.
- Gestion de l'accès concurrent de plusieurs utilisateurs exploitant différents applications accédant simultanément au même données.
- Disponibilité élevée des données : dans le cas des bases de données devant rester 24h/24 disponible, les tâches d'administration peuvent se faire sans interrompre l'utilisateur continue de la base.
- Conformité aux normes standards internationaux : Oracle est un système ouvert qui adhère aux standards de l'industrie dans le domaine des langage (LDD: langage de description de données, LMD: langage de manipulation de données. Supporte à 100% le SQL89 niveau 2).
- Des systèmes d'exploitations, des interfaces utilisateurs ainsi que des protocoles de communications sur réseaux.
- Offres des outils de contrôle d'accès et de sécurités pouvant gérer des logiques d'accès de données très complexes.
- Gestion des contraintes des intégrités.
- Client/serveur : Oracle permet de séparer le serveur de données et les applications clients.
- Ainsi la partie exploitation et la gestion des données partagées sont laissées à la charge du système serveur au moment où la station cliente se consacre à l'interprétation et à la représentation des résultats.
- Gestion des bases de données distribuées : Oracle permet de gérer une base de données distribuées sur différents sites du réseau comme s'il s'agissait d'une base de données unique.
- Oracle est portable sur différentes plates formes, les applications nécessitent peu ou pas de modification lors du portage d'une plate forme a une autre.
- Connectivité : Oracle permet a différents ordinateurs fonctionnant sous divers système d'exploitation de partage des données via un réseau.

2. Concepts de base de modèle relationnel :

Le modèle relationnel repose sur deux théories. La théorie mathématique des relations et celle de la normalisation des relations. Il est basé sur dix concepts simples regroupés selon trois aspects.

3. Les structures :

- **Relation** : la notion de relation découle la théorie des ensembles. Elle est définie comme étant le sous ensemble du produit cartésien de plusieurs ensembles appelés domaines.
- **Attribut/domaine** : un attribut est une colonne d'une relation caractérisée par un nom. Les attributs d'une même relation doivent être donc de noms différents. Un domaine est un ensemble de valeur peuvent être pris par un attribut.
- **Clé primaire** : une clé primaire est un ensemble formé d'une ou plusieurs colonnes qui identifie de façon unique chaque ligne de la table.
- **Clé étrangère** : une clé étrangère est un ensemble formé d'une ou plusieurs colonnes d'une table dont les valeurs doivent correspondre à une valeur dans une autre table qui est lui lié.

4. Les contraintes d'intégrités :

- ❖ **Contraintes d'intégrités de domaine :**

- **Contraintes d'initialisation :** chaque attributs doit prendre ses valeurs sur un seul domaine prédéfini.

- **Contraintes de manipulation :** toute comparaison d'attributs n'est acceptée que si ces attributs sont définis sur le même domaine.

- ❖ **Contraintes d'intégrités d'entité (clé primaire) :** impose que la ou les colonnes clés primaires soient toujours renseignées et uniques.

- ❖ **Contraintes d'intégrités référentielles (clé étrangère) :** met en évidence un lien inter table. Une suppression dans une table peut entraîner une mise à jour dans d'autres tables, afin de maintenir la cohérence des données.

5. Les avantages des bases de données relationnelles : les avantages des bases de données relationnelles sont :

- ❖ Indépendance du stockage physique des données vis-à-vis de la structure logique.
- ❖ Réduction des redondances des données.
- ❖ Optimisation de l'espace de stockage.
- ❖ Flexibilité accrue dans la conception de la base de données.
- ❖ Accès aisé aux données de la base.

6. Structures d'une base de données Oracle :

Les BD Oracle ont une structure physique et une structure logique séparées l'une de l'autre ce qui permet une plus grande flexibilité dans la gestion de la base de données. Toute manipulation des structures physiques de la base (fichier sur disque) n'affecte en rien l'accès et les références aux structures logiques de la base (table, Clé primaire,...).

7. Structure logique d'une base de données :

La structure logique d'une base de données Oracle est composée :

- ❖ Dans une ou plusieurs tablespaces.
- ❖ Les schémas des objets de la base (tables, Clé primaire, vue, cluster, procédures, stockées,...).
- ❖ Structure logique de stockage (blocs de données, extensions, segments).

8. Les tablespaces :

Une base de données Oracle est composée d'une ou plusieurs unités de stockages logiques appelées *tablespace* (ensemble de données regroupées ensemble logiquement). Cette organisation logique permet d'avoir les avantages suivants :

- ❖ Contrôler l'espace alloué à une base de données. Ainsi le cumul des capacités de stockage de toutes les *tablespace* représente la capacité globale de stockage de base de données.
- ❖ Allouer de l'espace de stockage sur plusieurs disques pour accroître les performances.
- ❖ Permettre un certain nombre de tâches administratives telle que l'archivage partiel et la reconstitution après un crash.
- ❖ Regrouper les objets appartenant à une même application, permettant ainsi de faciliter les tâches d'administration et de gestion des applications.
- ❖ Visualiser la base de données à partir d'une perspective logique, aide à comprendre comment le serveur Oracle alloue de l'espace de stockage et décharge les utilisateurs de la base de gestion physique des données.
- ❖ La base de données Oracle peut être divisée logiquement en un certain nombre de *tablespace* distincts (fortement conseillé).

- ❖ La *tablespace system* est obligatoire pour que la base de données puisse fonctionner (la base de données Oracle comporte une *tablespace system*).
- ❖ Les *tablespace* contiennent les segments de la base de données (table, Clé primaire,...).

A chaque *tablespace* sont associés un ou plusieurs fichiers physiques pour enregistrer les objets contenus dans la *tablespace*. Le cumul des tailles physiques des fichiers qui compose une *tablespace* représente la capacité totale de stockage de la *tablespace*....

9. Les schémas d'objets :

Un schéma est une collecte logique d'objets d'une base de données associée à un utilisateur. Une *tablespace* peut contenir des objets provenant de schémas différents et les objets d'un même schéma peuvent être dans différents *tablespace*.

- ❖ **Les tables** : la table est l'unité de base de stockage des données d'une BD Oracle. une table d'Oracle est identifiée par un nom et est composée de lignes (enregistrements) et colonnes (champs) ou chaque colonne a un nom, une taille et un type de données.
- ❖ **Les vues** : une vue est une présentation personnalisée des données d'une ou des plusieurs tables. Elle peut être assimilée à une requête stockée. Les vues ne contiennent aucune donnée mais exploitent les données des tables sur lesquelles elles sont basées.
- ❖ **Les séquences** : une séquence permet de générer des valeurs uniques pour des champs numériques. En programmation, les séquences simplifient considérablement la génération de valeurs pour identifier de manière unique des enregistrements d'une ou d'un ensemble de tables.
- ❖ **Les unités de programme** : le terme unité de programme est utilisé pour référencer les fonctions et procédures stockées ainsi que les packages. Les fonctions et les procédures stockées sont un ensemble d'instructions SQL et PL/SQL regroupés pour exécuter une tâche spécifique fréquemment demandée par les utilisateurs. Les packages sont un regroupement de fonctions et de procédures stockées ainsi que d'autres packages. Ils permettent entre autre, de faciliter la gestion des traitements stockés (fonctions, procédures et packages) et d'augmenter les performances de la base.
- ❖ **Les synonymes** : un synonyme est un alias associé à une table, vue, séquence ou une unité de programme, un synonyme n'est pas un objet en lui-même mais un alias qui permet de référencer directement un objet.
- ❖ **Les Clés primaires** : les Clés primaires sont créées en vue d'accroître les performances d'Oracle lors de traitement des requêtes sur les tables d'une base. Les Clés primaires peuvent être créées sur une ou plusieurs colonnes d'une table. Une des Clés primaires créées, Oracle se charge de manière automatique de maintenir et d'utiliser ses Clés primaires.

Une Clé primaire est physiquement et logiquement indépendante de la table sur laquelle elle est créée. Si une Clé primaire est supprimée toutes les applications continuent de s'exécuter normalement sauf que l'accès à certaines données peut être plus lent.

- ❖ **Les clusters** : un cluster est une méthode particulière de stockage physique des données sur disque. Elle consiste à regrouper physiquement sur le disque plusieurs tables qui ont les colonnes communes et qui sont souvent utilisées ensemble. En rassemblant physiquement les enregistrements communs, cette technique permet un gain de temps considérable lors de l'accès aux données des tables qui sont souvent utilisées en jointure.
- ❖ **Les clusters hachés** : les clusters hachés sont identiques aux clusters normaux du point de vue de stockage physique des tables, sauf que les enregistrements sont

regroupés en utilisant une fonction de hachage sur la clé cluster. Ainsi les enregistrements qui ont la même valeur en appliquant la fonction de hachage sur la clé cluster sont regroupés ensemble sur disque.

❖ **Lien de la base de données** : un lien de base de données (data base link) est un objet qui décrit un chemin d'accès entre une base de données et une autre dans un environnement de base de données distribuée.

Introduction a JAVA :

- 1990: une équipe d'informaticiens de SUN, dirigée par James Gosling, travaille sur la conception du langage Oak.
- 1993: le Web est apparu et SUN a compris qu'un langage comme Oak pouvait servir à développer de petites applications (les applets) exécutables au sein de documents au format HTML. Oak fut donc rebaptisé Java.
- 1995: première version publique de Java.

Java est un langage programmation simple, orienté objet, multi-plateforme, robuste, sûr et multi thread. Rien n'étant parfait, Java a tout de même un défaut : il est assez lent.

Le langage Java est composé principalement de deux *spécifications* :

Celle de la JVM (*Java Virtual Machine*) et celle de l'API (*Application Programming Interface*). Après avoir proposé ces spécifications, Sun Microsystems a diffusé plusieurs *implémentations*, principalement pour l'environnement *Solaris* (la version *Sun* d'Unix) et *Windows*. La première implémentation distribuée l'a été sous le nom de JDK 1.0. (*Java Development Kit*).

Il est fourni avec un ensemble de "packages" : les classes de base du JDK (*Java Developer's Toolkit*). Sans ces classes de base, Java n'est rien. Elles fournissent les fonctionnalités de base (entrées/sorties, réseau, etc...). Le langage lui-même se trouve dans le package (ensemble de classes) `java.lang`.

Du fait de son fonctionnement au travers d'une machine virtuelle, Java est un langage "qui voyage".

Nombreuses vérifications à la compilation et à l'exécution. Le compilateur `javac` génère un ensemble d'instructions (appelées *bytecode*) qui est exécutable par un processeur virtuel appelé la machine virtuelle Java (JVM, *Java Virtual Machine*).

Lors de la phase de compilation, Java s'assure qu'il n'y a aucun problème, que toutes les erreurs pouvant empêcher l'exécution du programme ont bien été traitées.

Cette vérification permet d'éviter qu'un problème grave ne corrompe la machine virtuelle ou le système qui l'exécute. Puisque les programmes Java "voyagent" d'une machine à l'autre.

- ✓ vérification de pointeur nul,
- ✓ vérification des bornes des tableaux,
- ✓ traitement très puissant des exceptions, vérification du *byte code* (pas de virus, très difficile de "cracker" des programmes compilés, etc...).

Un *Thread* est une partie de code, un "flot d'instructions" s'exécutant en concurrence avec d'autres *Threads* dans un même processus. Les applets ne sont pas des applications indépendantes, elles ne possèdent pas forcément de méthode *main()*.

Les bases à connaître :

- L'entité de base de tout code Java est la classe.
- Tout fichier source doit contenir une seule classe publique.
- Tout fichier source doit porter le nom de la classe publique qu'il contient, suivi du suffixe `.java`.
- Une classe peut contenir uniquement des données (attributs) et/ou des méthodes.
- Une méthode peut contenir uniquement des données et/ou du code.
- Trois variables d'environnement sont très importantes :
 - ✓ `CLASSPATH` : chemin de recherche des classes de base.

- ✓ **JAVA_HOME** : répertoire de base de l'installation du JDK.
 - ✓ **PATH** : doit contenir le répertoire contenant le compilateur java (javac), la machine virtuelle (java), le debugger (jdb), etc... En général \$JAVA_HOME/bin
- Ces variables sont souvent positionnées automatiquement lors de l'installation d'une IDE (symantec café, etc...), qui inclue le JDK.
- Java est immunisé contre les virus et les programmes bogués ou mal intentionnés. La protection est répartie sur quatre couches :
 - ❖ Le compilateur javac s'assure que les règles d'usage du langage, strictes, sont respectées; il est en particulier interdit de manipuler des pointeurs vers la mémoire du système et de transformer des nombres en adresses mémoire.
 - ❖ Le navigateur vérifie les instructions du bytecode pour vérifier que les règles de sécurité sont respectées ; en particulier, le code ne doit pas passer outre aux permissions d'accès.
 - ❖ Un chargeur de classes (class loader) s'assure qu'aucune classe importée ne viole l'espace qui lui est alloué.
 - ❖ Des spécifications pour l'environnement empêchent l'exécution des applets qui entreprendraient des actions destructives : c'est le browser qui vérifie si ces spécifications sont respectées et qui, en dernier ressort, interdit d'effacer des zones du disque dur ou autorise d'écrire sur le disque dur.
 - Java propose un ensemble de classes pour la programmation réseau. Ces classes sont regroupées dans le paquetage *java.net*. On peut programmer très facilement des architectures client/serveur en mode TCP ou UDP à l'aide des classes proposées.
 - Java lui-même n'est pas distribué. En revanche, il est fourni avec un ensemble de classes qui intègrent une gestion plus ou moins transparente du réseau.
 - Java s'occupe tout seul de la gestion mémoire.
 - L'architecture de JAVA est entièrement objet, cette architecture permet une réutilisation des objets créés par d'autres programmeurs (Pas besoin de réinventer la roue). Sa collection de classes standard est extrêmement riche (fonctions réseau, bases de données...).
 - JAVA est lent. Un remède à la lenteur: Le compilateur Just In Time (JIT).
 - Les principaux outils, composants du JDK :
 - ❖ **javac** : le compilateur
 - ❖ **java** : la machine java
 - ❖ **appletviewer** : visualiseur d'applet
 - ❖ **javadoc** : générateur de documentation
 - **compiler** : javac
 - Syntaxe : `javac pgm.java <RC>`
 - Génère autant de fich.class que de classes définies dans pgm.java
 - **la machine java** : java exécution d'une application java :
 - Syntaxe : `java NomAppli <RC>`
 - Lance l'application avec pour point d'entrée la méthode main().

GLOSSAIRE

Acteur : (modèle fonctionnel) est un objet explicite dans le modèle objet.

Analyse : Etape du cycle de développement dans laquelle le problème du monde réel est examiné afin d'en comprendre les spécifications, sans entreprendre d'implémentation.

Agrégation : la relation de la partie au tout, des composants à l'assemblage.

Attribut : est une donnée maintenue par les instances d'une classe.

Association : relation entre instances de deux ou plusieurs classes décrivant un groupe de liens avec une structure et une sémantique commune.

Banque de données (réservoir de données) : est un objet passif stockant les données pour un accès ultérieur.

Classe : est un enrichissement de la notion usuelle de *type*. Une classe permet de définir un type d'objet, éventuellement compliqué, associant des données et des procédures qui accèdent à ces données. Les données se présentent sous forme de champs désignés par identificateurs et dotés d'un type. Ces champs sont généralement des variables qui représentent l'état de l'objet.

Clef : est une ensemble d'attributs qui identifie uniquement un objet ou un lien.

Diagramme d'état : est un graphe dont les nœuds sont des états et les arcs des transitions marquées par leur évènement associé.

Diagramme de classe : décrit des ensembles d'objets et leurs caractéristiques et les liens inter classes.

Diagramme de flot de données : (DFD) est un graphe décrit les relations fonctionnelles qui existent entre les différentes valeurs calculées par un système.

Diagramme d'objet : fournit une notation graphique pour les concepts de classe et d'objets.

Discriminateur : un attribut de type énuméré qui indique quelle propriété est abstraite par la relation de généralisation.

Etat : valeurs des attributs et des liens d'un objet à un instant donné.

Evènement : quelque chose survenant instantanément à un moment donné.

Feuille : (en modèle objet) permet de découper le modèle objet complet en une série de pages.

Généralisation : relation entre une classe et un ou plusieurs versions raffinées de cette classe.

Héritage : l'héritage permet de définir des types d'objets plus précis à partir de types d'objets plus généraux.

Module : (en modèle objet) est unité logique qui groupe des classes, associations et généralisations.

Modèle dynamique : description des aspects de contrôle d'un système, comprenant les aspects temporels, le séquençement des opérations et l'interaction des objets.

Modèle fonctionnel : description des aspects du système qui transforment les valeurs en utilisant des fonctions, des représentations, des contraintes et des dépendances fonctionnelles.

Modèle objet : description de la structure des objets d'un système, comprenant leur identité, leurs relations avec les autres objets, leurs attributs et leurs relations.

Multiplicité : désigne le nombre d'instances d'une classe qui peuvent être liées à une instance d'une classe associée.

Lien : est une connexion physique entre instance d'objets.

Objet : possède deux aspects totalement indépendants :

- Un objet est une *chose dotée d'une identité* (on peut le référencer). Ceci s'oppose à la notion de *valeur* (par exemple 12 *n'est pas un objet*).
- Un objet peut, accessoirement, être composé de données et de procédures liées à ces données : les données de l'objet constituent le contexte de travail pour les procédures associées.

OMT : Object Modeling Technique de modélisation par objets.

Orienté objet : stratégie de développement de logiciel employant des objets comme base de l'analyse, de la conception et de l'implémentation.

Process : un process est une opération qui transforme des données.

Qualifieur : un attribut particulier qui permet de réduire la multiplicité d'une association.

Rôle : est une extrémité d'association.

Scénario : est une séquence réelle d'évènements qui se produit pendant une exécution du système.

Spécialisation : relation réciproque de généralisation.

Sous-classe : version raffinée d'une autre classe.

Super-classe : version plus abstraite d'une autre classe.

Table : organisation des données disposant d'un nombre spécifique de colonnes et d'un nombre arbitraire de rangées, souvent appelé relation.

Tablespace : est une unité de stockage logique utilisé par Oracle.

JAVA : est un langage de programmation orienté objet et d'une syntaxe proche de C++.

JDK : (Java Development Kit), qui contient *javac*, le compilateur qui transforme votre programme source en *bytecode*, *java*, l'interpréteur de *bytecode*, pour exécuter les applications, l'*AppletViewer*, pour exécuter les applets, *javadoc*, un programme permettant de créer automatiquement la documentation de vos programmes au format HTML, et d'autres utilitaires.

JRE : (Java Runtime Environment), qui contient un interpréteur de *bytecode* et tout ce qui est nécessaire pour diffuser vos applications aux utilisateurs.

يعتبر المركز الجهوي لحقن الدم مؤسسة ذات طابع خدماتي، تسعى لتوفير حاجيات المرضى من الدم. وهذا الأخير يعتبر مادة حيوية ضرورية لحياة الإنسان، و لا يمكن الحصول عليها إلا من تبرعات المواطنين، و هي كثيرا ما تكون مفقودة في المستشفيات لكثرة الطلب عليها من قبل المرضى و لقلة المتبرعين. و على هذا الأساس كان من واجب المركز أن يقوم بحملات التوعية و تحسيس المواطنين بأهمية التبرع بالدم و ضرورتها لفائدة المرضى الذين هم بأمس الحاجة إليها. من جهة أخرى كان من واجب المركز أن يسهل عملية التبرع و التقليل من مدة انتظار المتبرعين في طوابير قصد عمل خيري و تطوعي كهذا.

و من الأسباب التي تعطل عملية التبرع الخلل الكبير في النظام المعلوماتي اليدوي الذي يستعمله المركز في تسيير معلوماته.

من هذا المنطلق حاولنا من خلال دراستنا الأولية للمركز أن نمسح الغبار على النقاط السوداء في النظام المعمول به حاليا، و استخلاص الأسباب المؤدية إليها. و بعد تلك الدراسة الأولية، قمنا بتشكيل نظام معلوماتي جديد يتماشى و المتطلبات العصرية للمركز اعتمادا على طريقة " OMT " .

بعد انتهاءنا من المرحلة التشكيلية قمنا بإتجاز برنامج نهائي و قابل للاستعمال بعد توفير كل وسائل الإعلام الآلي المطلوبة. و هذا البرنامج يسهل عملية البحث على معلومات المتبرعين و يحافظ عليها خلال تنقلها بين مصالح المركز، و يحدد مسؤوليات الأفراد في كل مركز.

كلمات المفاتيح: مركز حقن الدم، OMT، JAVA، Oracle.

☆ مريض في انتظارك، حياته متعلقة في قطرة من دمك، فلا تبخل عليه بها ☆

Summarized

The regional center of blood transfusion is a service that aspires to cover the needs of the patients concerning blood. This last being vital for man and its acquirement depends on the grants made by the citizens. It is made of it a rare enough matter in our hospitals and it is: to the increasing demand on this product, and to the rarity of the donors. For it the centers organized companions of sensitizations to put forward the importance of the grants for the patients, whose life is dependent of it. Of another quoted the center put everything in work for easiness these grants and so to reduce the time of waiting of the donors.

Among the reasons that influence on the grants, there's the heaviness of the system of information manual that uses the center. Taking it in considerations us has, after a preliminary investigation to the center, to try to correct the pain working of this obsolete system.

We also conceived a new system of information; modern and reliable while being based on the method of conception "OMT".

After the stage of conception we have finalized operational software and this after the acquirement of the computer material required.

This software facilitates the research of the data relations to the donors and permits the transfer and the use of these data between the different services of the center. In addition, it puts every user owing his responsibilities.

Key Words: The center of blood transfusion, OMT, JAVA, Oracle.

☆ A patient waits for you; his life depends on a taste of your blood. ☆

RESUME

Le centre régional de transfusion sanguine est un service qui aspire à couvrir les besoins des malades en matière du sang. Cette dernière étant vitale pour l'homme et son acquisition dépend des dons faits par les citoyens. C'est en fait une matière assez rare dans nos hôpitaux, et cela est dû : à la demande croissante sur ce produit, et à la rareté des donneurs. Pour cela le centre a organisé des campagnes de sensibilisations pour faire valoir l'importance des dons pour les malades, dont la vie en est dépendante. D'un autre côté le centre a tout mis en œuvre pour faciliter ces dons et ainsi réduire le temps d'attente des donneurs.

Parmi les causes qui influent sur les dons, il y'a la lourdeur du système d'informations manuel qu'utilise le centre. Prenant cela en considérations nous avons, après une enquête préliminaire au centre, essayer de corriger le mal fonctionnement du à ce système obsolète.

Nous avons aussi conçu un nouveau système d'informations, moderne et fiable en se basant sur la méthode de conception « OMT ».

Après l'étape de conception nous avons finalisé un logiciel opérationnel et cela après l'acquisition du matériel informatique requis.

Ce logiciel facilite la recherche des données relations aux donneurs et permet le transfert et l'utilisation de ces données entre les différents services du centre. En plus, il met chaque utilisateur devant ses responsabilités.

Mots clés : Centre de transfusion sanguine, OMT, JAVA, Oracle.



Un malade vous attend, sa vie dépend d'une goutte de votre sang.

